

Microsoft Entra authentication documentation

Learn how to manage and deploy Microsoft Entra self-service password reset, multifactor authentication, custom banned password list, and smart lockout.

About authentication

OVERVIEW

[What is authentication?](#)

[What are authentication methods?](#)

[Security information registration](#)

[What is passwordless?](#)

Deploy self-service password reset

CONCEPT

[How self-service password reset works](#)

[How password writeback works](#)

TUTORIAL

[Enable self-service password reset](#)

[Enable password writeback to on-premises](#)

DEPLOY

[Deployment guide for self-service password reset](#)

[Enable password reset from the Windows login screen](#)

Enable Microsoft Entra multifactor authentication

CONCEPT

How Microsoft Entra multifactor authentication works

TUTORIAL

[Enable Microsoft Entra multifactor authentication](#)

[Enable risk-based Microsoft Entra multifactor authentication](#)

DEPLOY

[Deployment guide for Microsoft Entra multifactor authentication](#)

[Use NPS extension to integrate on-premises applications](#)

Deploy Microsoft Entra password protection

CONCEPT

[Eliminate weak passwords in the cloud](#)

[Eliminate weak passwords on-premises](#)

GET STARTED

[Configure the banned password list](#)

HOW-TO GUIDE

[Deploy Microsoft Entra password protection](#)

What is Microsoft Entra authentication?

Article • 03/04/2025

One of the main features of an identity platform is to verify, or *authenticate*, credentials when a user signs in to a device, application, or service. In Microsoft Entra ID, authentication involves more than just the verification of a username and password. To improve security and reduce the need for help desk assistance, Microsoft Entra authentication includes the following components:

- Self-service password reset
- Microsoft Entra multifactor authentication
- Hybrid integration to write password changes back to on-premises environment
- Hybrid integration to enforce password protection policies for an on-premises environment
- Passwordless authentication

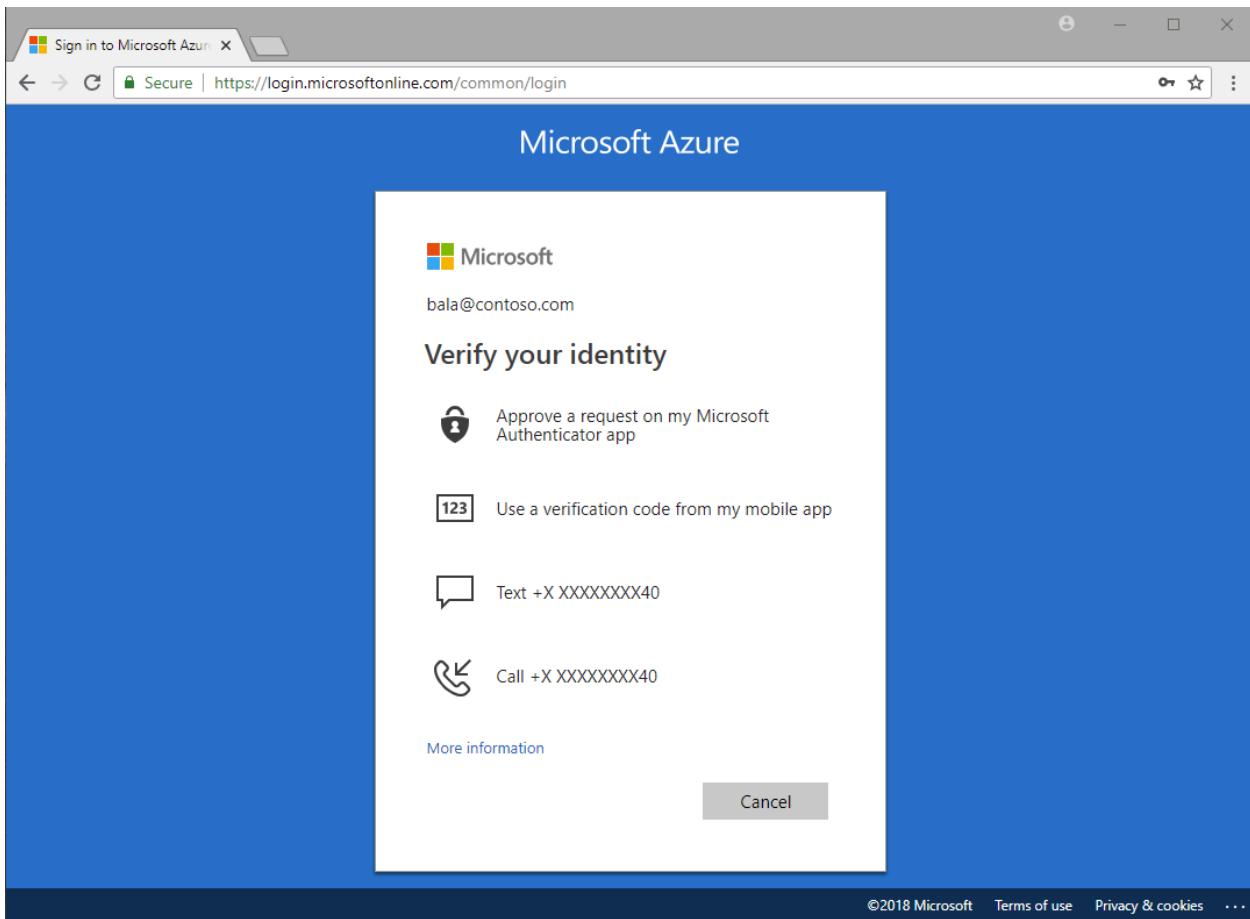
Take a look at our short video to learn more about these authentication components.

<https://learn-video.azurefd.net/vod/player?id=5ee3cad5-3360-48da-b520-1a0d96710a38&locale=en-us&embedUrl=%2Fentra%2Fidentity%2Fauthentication%2Foverview-authentication> ↗

Improve the end-user experience

Microsoft Entra ID helps to protect a user's identity and simplify their sign-in experience. Features like self-service password reset let users update or change their passwords using a web browser from any device. This feature is especially useful when the user has forgotten their password or their account is locked. Without waiting for a helpdesk or administrator to provide support, a user can unblock themselves and continue to work.

Microsoft Entra multifactor authentication lets users choose an additional form of authentication during sign-in, such as a phone call or mobile app notification. This ability reduces the requirement for a single, fixed form of secondary authentication like a hardware token. If the user doesn't currently have one form of additional authentication, they can choose a different method and continue to work.



Passwordless authentication removes the need for the user to create and remember a secure password at all. Capabilities like Windows Hello for Business or FIDO2 security keys let users sign in to a device or application without a password. This ability can reduce the complexity of managing passwords across different environments.

Self-service password reset

Self-service password reset gives users the ability to change or reset their password, with no administrator or help desk involvement. If a user's account is locked or they forget their password, they can follow prompts to unblock themselves and get back to work. This ability reduces help desk calls and loss of productivity when a user can't sign in to their device or an application.

Self-service password reset works in the following scenarios:

- **Password change** - when a user knows their password but wants to change it to something new.
- **Password reset** - when a user can't sign in, such as when they forgot password, and want to reset their password.
- **Account unlock** - when a user can't sign in because their account is locked out and want to unlock their account.

When a user updates or resets their password using self-service password reset, that password can also be written back to an on-premises Active Directory environment. Password writeback makes sure that a user can immediately use their updated credentials with on-premises devices and applications.

Microsoft Entra multifactor authentication

Multifactor authentication is a process where a user is prompted during the sign-in process for an additional form of identification, such as to enter a code on their cellphone or to provide a fingerprint scan.

If you only use a password to authenticate a user, it leaves an insecure vector for attack. If the password is weak or has been exposed elsewhere, is it really the user signing in with the username and password, or is it an attacker? When you require a second form of authentication, security is increased as this additional factor isn't something that's easy for an attacker to obtain or duplicate.



Microsoft Entra multifactor authentication works by requiring two or more of the following authentication methods:

- Something you know, typically a password.
- Something you have, such as a trusted device that isn't easily duplicated, like a phone or hardware key.
- Something you are - biometrics like a fingerprint or face scan.

Users can register themselves for both self-service password reset and Microsoft Entra multifactor authentication in one step to simplify the on-boarding experience.

Administrators can define what forms of secondary authentication can be used.

Microsoft Entra multifactor authentication can also be required when users perform a self-service password reset to further secure that process.

Password protection

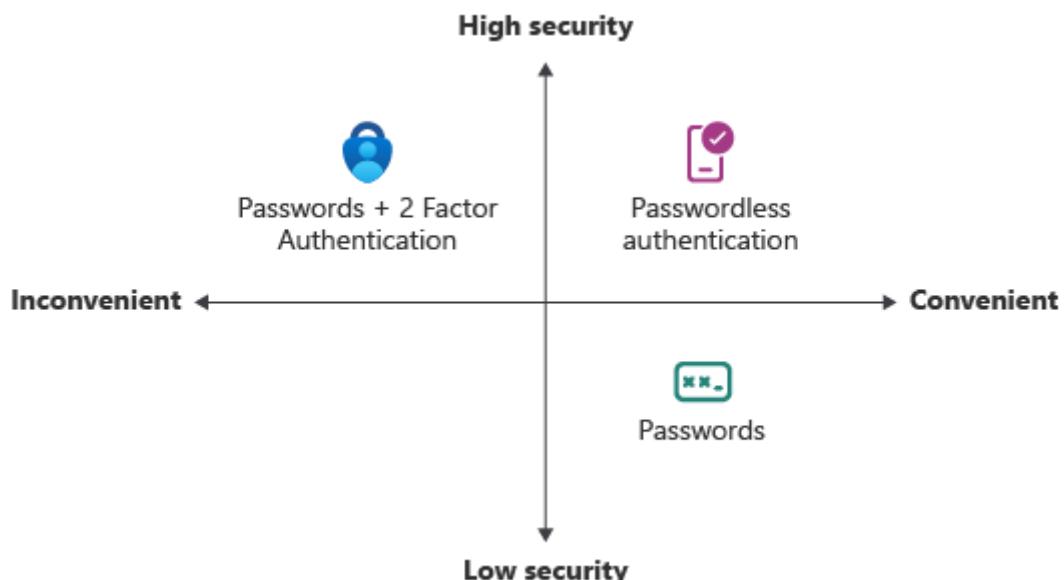
By default, Microsoft Entra ID blocks weak passwords such as *Password1*. A global banned password list is automatically updated and enforced that includes known weak passwords. If a Microsoft Entra user tries to set their password to one of these weak passwords, they receive a notification to choose a more secure password.

To increase security, you can define custom password protection policies. These policies can use filters to block any variation of a password containing a name such as *Contoso* or a location like *London*, for example.

For hybrid security, you can integrate Microsoft Entra password protection with an on-premises Active Directory environment. A component installed in the on-premises environment receives the global banned password list and custom password protection policies from Microsoft Entra ID, and domain controllers use them to process password change events. This hybrid approach makes sure that no matter how or where a user changes their credentials, you enforce the use of strong passwords.

Passwordless authentication

The end-goal for many environments is to remove the use of passwords as part of sign-in events. Features like Azure password protection or Microsoft Entra multifactor authentication help improve security, but a username and password remains a weak form of authentication that can be exposed or brute-force attacked.



When you sign in with a passwordless method, credentials are provided by using methods like biometrics with Windows Hello for Business, or a FIDO2 security key. These authentication methods can't be easily duplicated by an attacker.

Microsoft Entra ID provides ways to natively authenticate using passwordless methods to simplify the sign-in experience for users and reduce the risk of attacks.

Next steps

To get started, see the [tutorial for self-service password reset \(SSPR\)](#) and [Microsoft Entra multifactor authentication](#).

To learn more about self-service password reset concepts, see [How Microsoft Entra self-service password reset works](#).

To learn more about multifactor authentication concepts, see [How Microsoft Entra multifactor authentication works](#).

Feedback

Was this page helpful?



[Provide product feedback](#) ↗

Tutorial: Enable users to unlock their account or reset passwords using Microsoft Entra self-service password reset

Article • 03/04/2025

Microsoft Entra self-service password reset (SSPR) gives users the ability to change or reset their password, with no administrator or help desk involvement. If Microsoft Entra ID locks a user's account or they forget their password, they can follow prompts to unblock themselves and get back to work. This ability reduces help desk calls and loss of productivity when a user can't sign in to their device or an application. We recommend this video on [How to enable and configure SSPR in Microsoft Entra ID](#). We also have a video for IT administrators on [resolving the six most common end-user error messages with SSPR](#).

Important

This tutorial shows an administrator how to enable self-service password reset. If you're an end user already registered for self-service password reset and need to get back into your account, go to the [Microsoft Online password reset](#) page.

If your IT team hasn't enabled the ability to reset your own password, reach out to your helpdesk for additional assistance.

In this tutorial you learn how to:

- ✓ Enable self-service password reset for a group of Microsoft Entra users
- ✓ Set up authentication methods and registration options
- ✓ Test the SSPR process as a user

Important

In March 2023, we announced the deprecation of managing authentication methods in the legacy multifactor authentication and self-service password reset (SSPR) policies. Beginning September 30, 2025, authentication methods can't be managed in these legacy MFA and SSPR policies. We recommend customers use

the manual migration control to migrate to the Authentication methods policy by the deprecation date.

Video tutorial

You can also follow along in a related video: [How to enable and configure SSPR in Microsoft Entra ID ↗](#).

Prerequisites

To finish this tutorial, you need the following resources and privileges:

- A working Microsoft Entra tenant with at least a Microsoft Entra ID P1 license is required for password reset. For more information about license requirements for password change and password reset in Microsoft Entra ID, see [Licensing requirements for Microsoft Entra self-service password reset](#).
- An account with at least the Authentication Policy Administrator role.
- A non-administrator user with a password you know, like *testuser*. You'll test the end-user SSPR experience using this account in this tutorial.
 - If you need to create a user, see [Quickstart: Add new users to Microsoft Entra ID](#).
- A group that the non-administrator user is a member of, like *SSPR-Test-Group*. You'll enable SSPR for this group in this tutorial.
 - If you need to create a group, see [Create a basic group and add members using Microsoft Entra ID](#).

Enable self-service password reset

Microsoft Entra ID lets you enable SSPR for *None*, *Selected*, or *All* users. This granular ability lets you choose a subset of users to test the SSPR registration process and workflow. When you're comfortable with the process and the time is right to communicate the requirements with a broader set of users, you can select a group of users to enable for SSPR. Or, you can enable SSPR for everyone in the Microsoft Entra tenant.

ⓘ Note

Currently, you can only enable one Microsoft Entra group for SSPR using the Microsoft Entra admin center. As part of a wider deployment of SSPR, Microsoft

Entra ID supports nested groups.

In this tutorial, set up SSPR for a set of users in a test group. Use the *SSPR-Test-Group* and provide your own Microsoft Entra group as needed:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Password reset** from the menu on the left side.
3. From the **Properties** page, under the option *Self service password reset enabled*, choose **Selected**.
4. If your group isn't visible, choose **No groups selected**, browse for and select your Microsoft Entra group, like *SSPR-Test-Group*, and then choose **Select**.

The screenshot shows the 'Password reset | Properties' page in the Microsoft Entra admin center. The left sidebar has sections for 'Manage' (Properties, Authentication methods, Registration, Notifications, Customization, On-premises integration, Administrator Policy) and 'Activity' (Audit logs, Usage & insights). The main area shows 'Self service password reset enabled' set to 'Selected'. A note says 'These settings only apply to end users in your organization. Admins are always enabled for self-service password reset and are required to use two authentication methods to reset their password.' A large red box highlights the 'No groups selected' button in the 'Select group' section. A save button is at the top right.

5. To enable SSPR for the select users, select **Save**.

Select authentication methods and registration options

When users need to unlock their account or reset their password, they're prompted for another confirmation method. This extra authentication factor makes sure that Microsoft Entra finished only approved SSPR events. You can choose which authentication methods to allow, based on the registration information the user provides.

1. From the menu on the left side of the **Authentication methods** page, set the **Number of methods required to reset** to 2.

To improve security, you can increase the number of authentication methods required for SSPR.

2. Choose the **Methods available to users** that your organization wants to allow. For this tutorial, check the boxes to enable the following methods:

- *Mobile app notification*
- *Mobile app code*
- *Email*
- *Mobile phone*

You can enable other authentication methods, like *Office phone* or *Security questions*, as needed to fit your business requirements.

3. To apply the authentication methods, select **Save**.

Before users can unlock their account or reset a password, they must register their contact information. Microsoft Entra ID uses this contact information for the different authentication methods set up in the previous steps.

An administrator can manually provide this contact information, or users can go to a registration portal to provide the information themselves. In this tutorial, set up Microsoft Entra ID to prompt the users for registration the next time they sign in.

1. From the menu on the left side of the **Registration** page, select **Yes** for **Require users to register when signing in**.
2. Set **Number of days before users are asked to reconfirm their authentication information** to **180**.

It's important to keep the contact information up to date. If outdated contact information exists when an SSPR event starts, the user might not be able to unlock their account or reset their password.

3. To apply the registration settings, select **Save**.

Note

The interruption to request to register contact information during signing in only occurs if the conditions configured on the settings are met. This only applies to

users and admin accounts that are enabled to reset passwords using Microsoft Entra self-service password reset.

Set up notifications and customizations

To keep users informed about account activity, you can set up Microsoft Entra ID to send email notifications when an SSPR event happens. These notifications can cover both regular user accounts and admin accounts. For admin accounts, this notification provides another layer of awareness when a privileged administrator account password is reset using SSPR. Microsoft Entra ID can notify all Administrators when someone uses SSPR on an admin account.

1. From the menu on the left side of the **Notifications** page, set up the following options:

- Set **Notify users on password resets?** option to **Yes**.
- Set **Notify all admins when other admins reset their password?** to **Yes**.

2. To apply the notification preferences, select **Save**.

If users need more help with the SSPR process, you can customize the "Contact your administrator" link. The user can select this link in the SSPR registration process and when they unlock their account or resets their password. To make sure your users get the support needed, we recommend you provide a custom helpdesk email or URL.

1. From the menu on the left side of the **Customization** page, set **Customize helpdesk link** to **Yes**.
2. In the **Custom helpdesk email or URL** field, provide an email address or web page URL where your users can get more help from your organization, like <https://support.contoso.com/>
3. To apply the custom link, select **Save**.

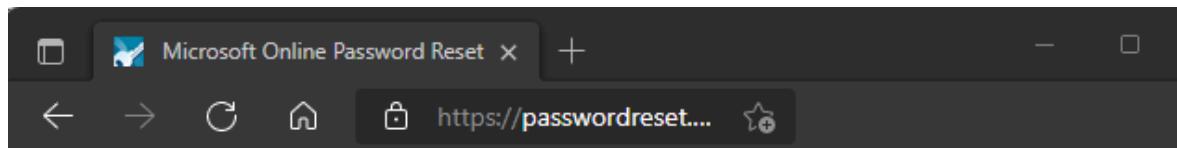
Test self-service password reset

With SSPR enabled and set up, test the SSPR process with a user that's part of the group you selected in the previous section, like *Test-SSPR-Group*. The following example uses the *testuser* account. Provide your own user account. It's part of the group you enabled for SSPR in the first section of this tutorial.

Note

When you test self-service password reset, use a non-administrator account. By default, Microsoft Entra ID enables self-service password reset for admins. They're required to use two authentication methods to reset their password. For more information, see [Administrator reset policy differences](#).

1. To see the manual registration process, open a new browser window in InPrivate or incognito mode, and browse to <https://aka.ms/ssprsetup>. Microsoft Entra ID directs users to this registration portal when they sign in next time.
2. Sign in with a non-administrator test user, like *testuser*, and register your authentication methods contact information.
3. Once finished, select the button marked **Looks good** and close the browser window.
4. Open a new browser window in InPrivate or incognito mode, and browse to <https://aka.ms/sspr>.
5. Enter your non-administrator test users' account information, like *testuser*, the characters from the CAPTCHA, and then select **Next**.



6. Follow the verification steps to reset your password. When finished, you receive an email notification that your password was reset.

Clean up resources

In a later tutorial in this series, you set up password writeback. This feature writes password changes from Microsoft Entra SSPR back to an on-premises AD environment. If you want to continue with this tutorial series to set up password writeback, don't disable SSPR now.

If you no longer want to use the SSPR functionality you set up as part of this tutorial, set the SSPR status to **None** using the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Password reset**.

3. From the **Properties** page, under the option *Self service password reset enabled*, select **None**.
4. To apply the SSPR change, select **Save**.

FAQs

This section explains common questions from administrators and end-users who try SSPR:

- Why aren't on-premises password policies displayed during SSPR?

At this time, Microsoft Entra Connect and cloud sync don't support sharing password policy details with the cloud. SSPR only displays the cloud password policy details, and can't show on-premises policies.

- Why do federated users wait up to 2 minutes after they see **Your password has been reset** before they can use passwords that are synchronized from on-premises?

For federated users whose passwords are synchronized, the source of authority for the passwords is on-premises. As a result, SSPR updates only the on-premises passwords. Password hash synchronization back to Microsoft Entra ID is scheduled for every 2 minutes.

- When a newly created user who is pre-populated with SSPR data such as phone and email visits the SSPR registration page, **Don't lose access to your account!** appears as the title of the page. Why don't other users who have SSPR data pre-populated see the message?

A user who sees **Don't lose access to your account!** is a member of SSPR/combined registration groups that are configured for the tenant. Users who don't see **Don't lose access to your account!** weren't part of the SSPR/combined registration groups.

- When some users go through SSPR process and reset their password, why don't they see the password strength indicator?

Users who don't see weak/strong password strength have synchronized password writeback enabled. Since SSPR can't determine the password policy of the customer's on-premises environment, it can't validate password strength or weakness.

Next steps

In this tutorial, you enabled Microsoft Entra self-service password reset for a selected group of users. You learned how to:

- ✓ Enable self-service password reset for a group of Microsoft Entra users
- ✓ Set up authentication methods and registration options
- ✓ Test the SSPR process as a user

[Enable Microsoft Entra multifactor authentication](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication

Article • 03/04/2025

Multifactor authentication is a process in which a user is prompted for additional forms of identification during a sign-in event. For example, the prompt could be to enter a code on their cellphone or to provide a fingerprint scan. When you require a second form of identification, security is increased because this additional factor isn't easy for an attacker to obtain or duplicate.

Microsoft Entra multifactor authentication and Conditional Access policies give you the flexibility to require MFA from users for specific sign-in events.

Important

This tutorial shows an administrator how to enable Microsoft Entra multifactor authentication. To step through the multifactor authentication as a user, see [Sign in to your work or school account using your two-step verification method](#).

If your IT team hasn't enabled the ability to use Microsoft Entra multifactor authentication, or if you have problems during sign-in, reach out to your Help desk for additional assistance.

In this tutorial you learn how to:

- ✓ Create a Conditional Access policy to enable Microsoft Entra multifactor authentication for a group of users.
- ✓ Configure the policy conditions that prompt for MFA.
- ✓ Test configuring and using multifactor authentication as a user.

Prerequisites

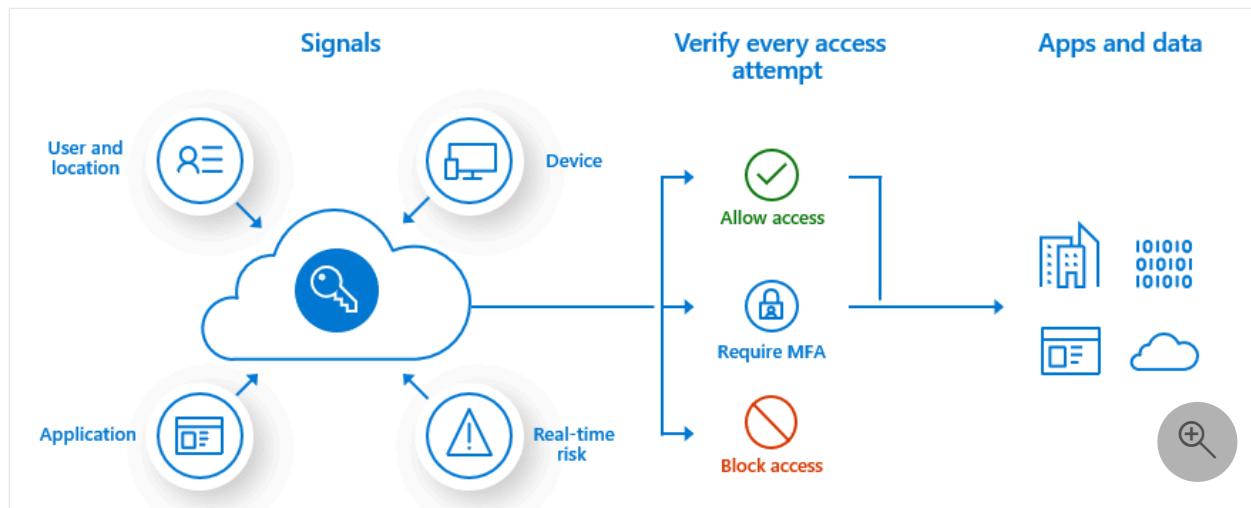
To complete this tutorial, you need the following resources and privileges:

- A working Microsoft Entra tenant with Microsoft Entra ID P1 or trial licenses enabled.
 - If you need to, [create one for free](#).

- An account with at least the [Conditional Access Administrator](#) role. Some MFA settings can also be managed by an [Authentication Policy Administrator](#).
- A non-administrator account with a password that you know. For this tutorial, we created such an account, named *testuser*. In this tutorial, you test the end-user experience of configuring and using Microsoft Entra multifactor authentication.
 - If you need information about creating a user account, see [Add or delete users using Microsoft Entra ID](#).
- A group that the non-administrator user is a member of. For this tutorial, we created such a group, named *MFA-Test-Group*. In this tutorial, you enable Microsoft Entra multifactor authentication for this group.
 - If you need more information about creating a group, see [Create a basic group and add members using Microsoft Entra ID](#).

Create a Conditional Access policy

The recommended way to enable and use Microsoft Entra multifactor authentication is with Conditional Access policies. Conditional Access lets you create and define policies that react to sign-in events and that request additional actions before a user is granted access to an application or service.



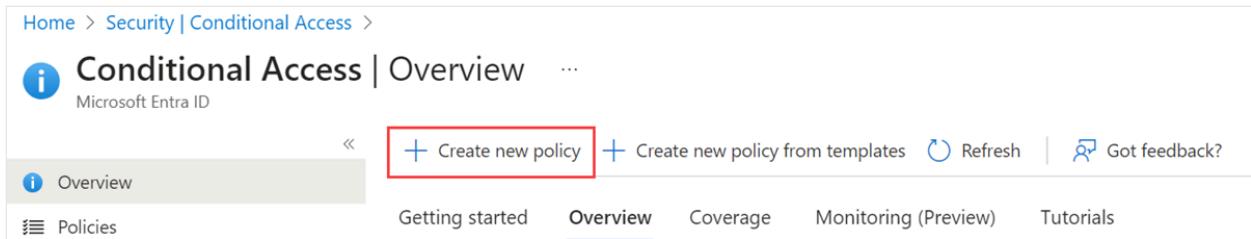
Conditional Access policies can be applied to specific users, groups, and apps. The goal is to protect your organization while also providing the right levels of access to the users who need it.

In this tutorial, we create a basic Conditional Access policy to prompt for MFA when a user signs in. In a later tutorial in this series, we configure Microsoft Entra multifactor authentication by using a risk-based Conditional Access policy.

First, create a Conditional Access policy and assign your test group of users as follows:

1. Sign in to the Microsoft Entra admin center [↗](#) as at least a **Conditional Access Administrator**.

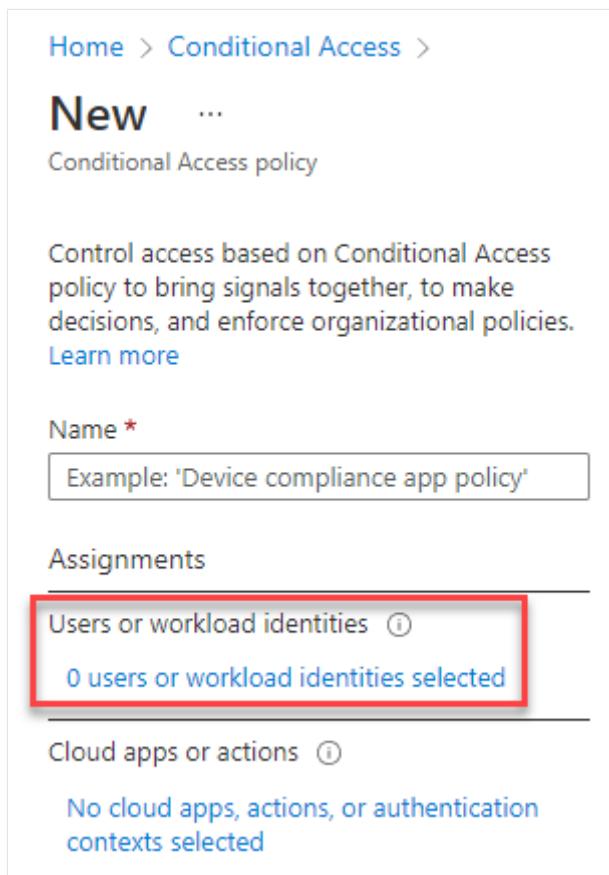
2. Browse to **Protection > Conditional Access > Overview**, select **+ Create new policy**.



The screenshot shows the Microsoft Entra admin center interface. At the top, there's a navigation bar with 'Home > Security | Conditional Access >'. Below it is the title 'Conditional Access | Overview' with a Microsoft Entra ID icon. A red box highlights the '+ Create new policy' button. The top navigation bar also includes 'Create new policy from templates', 'Refresh', and 'Got feedback?' buttons. The main menu bar below the title has tabs: 'Overview' (which is selected and highlighted in grey), 'Policies', 'Getting started', 'Overview' (underlined in blue), 'Coverage', 'Monitoring (Preview)', and 'Tutorials'.

1. Enter a name for the policy, such as *MFA Pilot*.

2. Under **Assignments**, select the current value under **Users or workload identities**.



The screenshot shows the 'New Conditional Access policy' creation page. At the top, there's a breadcrumb navigation 'Home > Conditional Access >'. The title is 'New' with a 'Conditional Access policy' subtitle. Below the title, there's a descriptive text: 'Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies.' followed by a 'Learn more' link. The 'Name *' field is present with a placeholder 'Example: 'Device compliance app policy''. The 'Assignments' section is expanded, showing the 'Users or workload identities' tab, which is highlighted with a red box. It displays '0 users or workload identities selected'. The 'Cloud apps or actions' tab is also visible but not selected. Below the assignments section, there's a note: 'No cloud apps, actions, or authentication contexts selected'.

3. Under **What does this policy apply to?**, verify that **Users and groups** is selected.

4. Under **Include**, choose **Select users and groups**, and then select **Users and groups**.

New

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies.

[Learn more](#)

Name *

Example: 'Device compliance app policy'

Assignments

Users or workload identities (i)

Specific users included

✖ "Select users and groups" must be configured

Cloud apps or actions (i)

No cloud apps, actions, or authentication contexts selected

Conditions (i)

0 conditions selected

Access controls

Control access based on who the policy will apply to, such as users and groups, workload identities, directory roles, or external guests.

[Learn more](#)

What does this policy apply to?

Users and groups

Include Exclude None All users Select users and groups All guest and external users (i) Directory roles (i) Users and groups

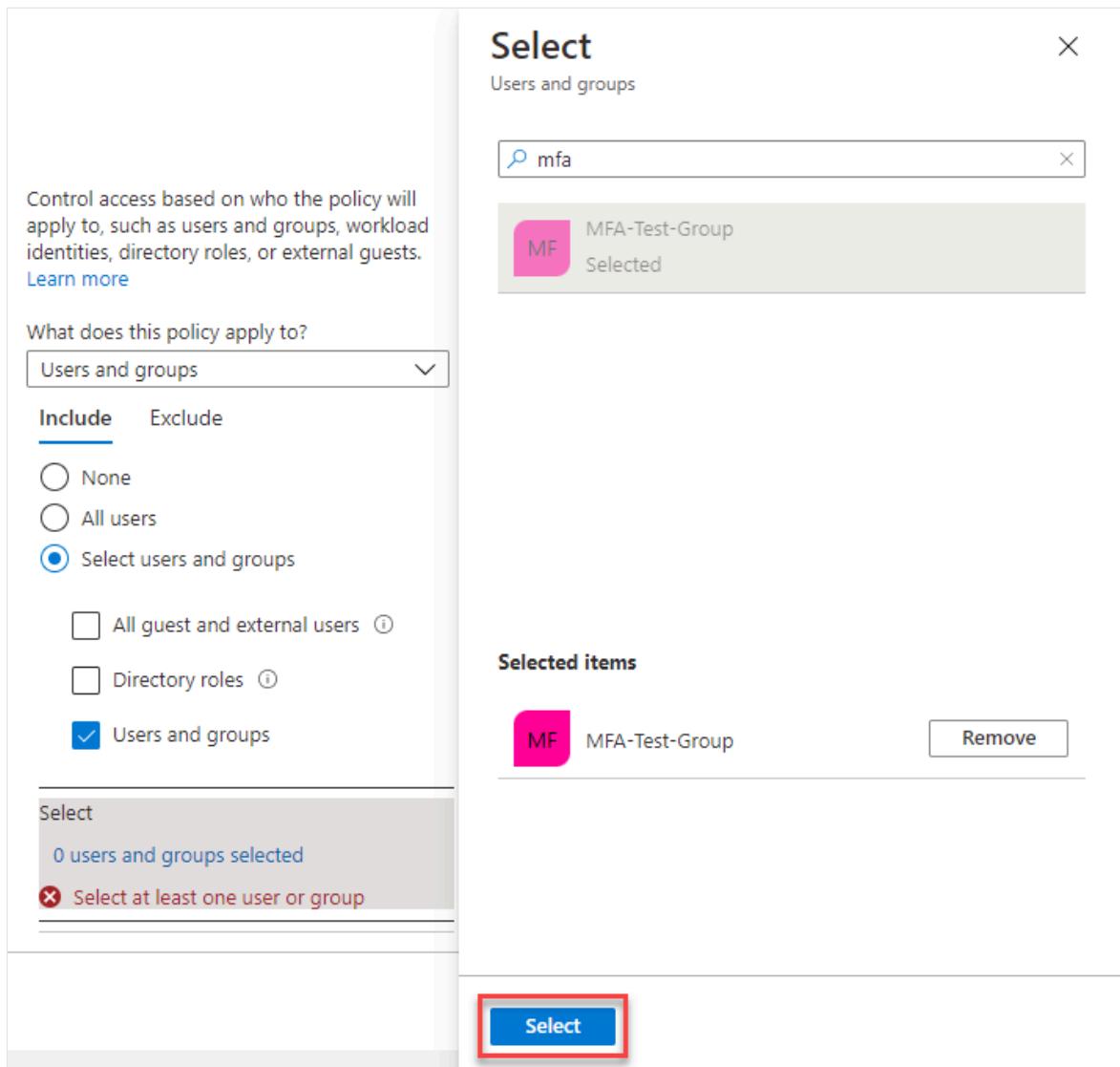
Select

0 users and groups selected

✖ Select at least one user or group

Since no one is assigned yet, the list of users and groups (shown in the next step) opens automatically.

5. Browse for and select your Microsoft Entra group, such as *MFA-Test-Group*, then choose **Select**.



We've selected the group to apply the policy to. In the next section, we configure the conditions under which to apply the policy.

Configure the conditions for multifactor authentication

Now that the Conditional Access policy is created and a test group of users is assigned, define the cloud apps or actions that trigger the policy. These cloud apps or actions are the scenarios that you decide require additional processing, such as prompting for multifactor authentication. For example, you could decide that access to a financial application or use of management tools require an additional prompt for authentication.

Configure which apps require multifactor authentication

For this tutorial, configure the Conditional Access policy to require multifactor authentication when a user signs in.

1. Select the current value under **Cloud apps or actions**, and then under **Select what this policy applies to**, verify that **Cloud apps** is selected.

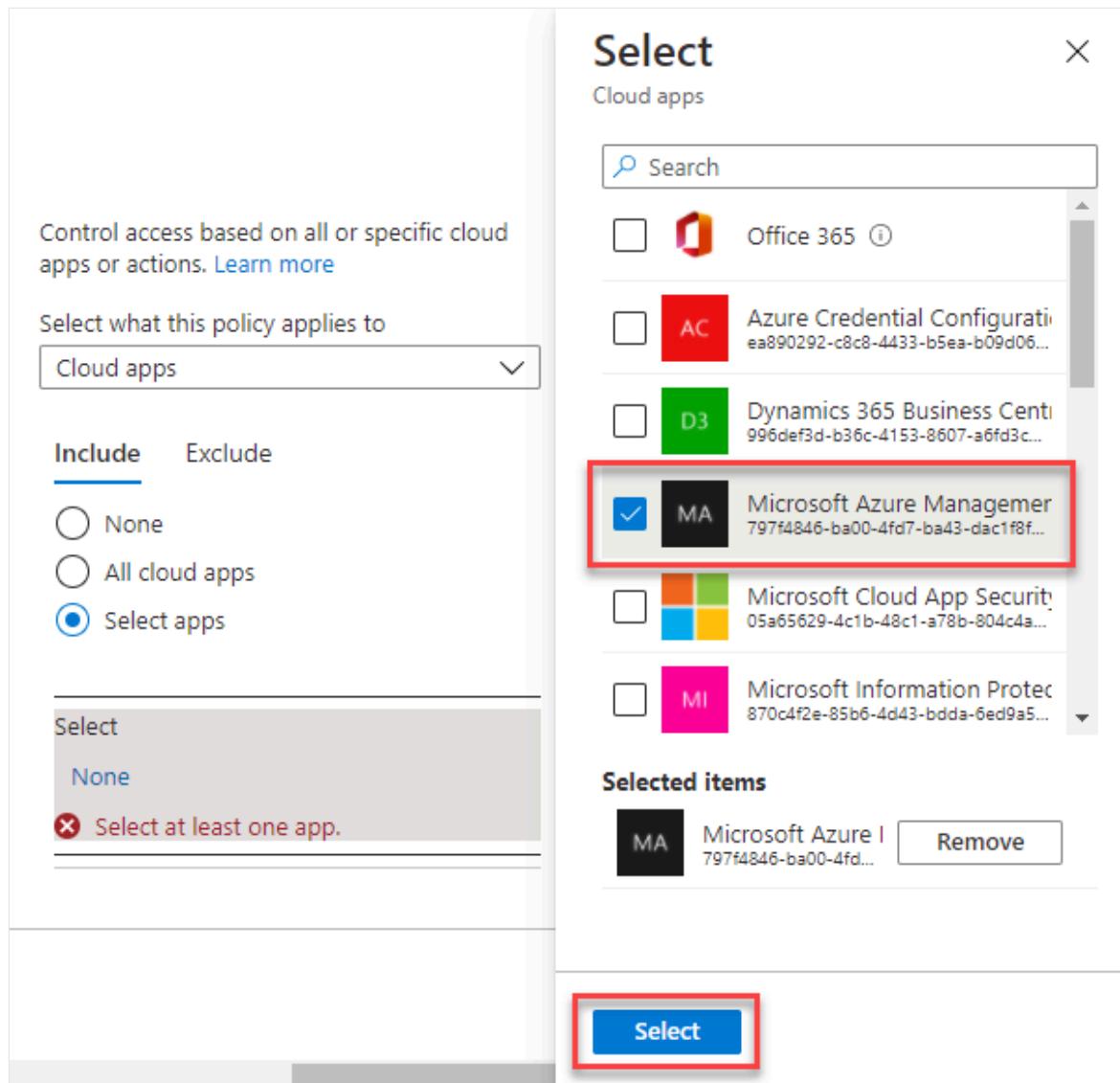
2. Under **Include**, choose **Select resources**.

Since no apps are yet selected, the list of apps (shown in the next step) opens automatically.

 **Tip**

You can choose to apply the Conditional Access policy to **All resources** (formerly 'All cloud apps') or **Select resources**. To provide flexibility, you can also exclude certain apps from the policy.

3. Browse the list of available sign-in events that can be used. For this tutorial, select **Windows Azure Service Management API** so that the policy applies to sign-in events. Then choose **Select**.



The screenshot shows the 'Select' dialog box for a Conditional Access policy. On the left, there's a summary section with a link to learn more about controlling access based on all or specific cloud apps or actions. Below it, a dropdown menu shows 'Cloud apps' is selected. Under the 'Include' tab, the 'Select apps' option is chosen. A note at the bottom says 'Select at least one app.' On the right, the 'Cloud apps' list is shown with several items: Office 365, Azure Credential Configuration, Dynamics 365 Business Central, Microsoft Azure Management (which is selected and highlighted with a red border), Microsoft Cloud App Security, and Microsoft Information Protection. At the bottom of the list, there's a 'Selected items' section with a single item: Microsoft Azure Management. A large blue 'Select' button is at the bottom right of the dialog.

Configure multifactor authentication for access

Next, we configure access controls. Access controls let you define the requirements for a user to be granted access. They might be required to use an approved client app or a device that's hybrid-joined to Microsoft Entra ID.

In this tutorial, configure the access controls to require multifactor authentication during a sign-in event.

1. Under **Access controls**, select the current value under **Grant**, and then select **Grant access**.

The screenshot shows the Microsoft Conditional Access Overview page with a 'New' policy named 'MFA Pilot'. The 'Access controls' section is expanded, showing the 'Grant' tab selected. The 'Grant' tab has a red box around it, and the 'Grant access' radio button is selected, also with a red box. Below it, other options like 'Require multifactor authentication' and 'Require password change' are listed with checkboxes. At the bottom right of the dialog, there is a 'Select' button.

Home > Security | Conditional Access > Conditional Access | Overview >

New

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies. [Learn more](#)

Name * MFA Pilot ✓

Assignments

Users ⓘ Specific users included

Target resources ⓘ 1 app included

Conditions ⓘ 0 conditions selected

Access controls

Grant ⓘ 0 controls selected

Session ⓘ 0 controls selected

Enable policy Report-only On Off

Create Select

2. Select **Require multifactor authentication**, and then choose **Select**.

Grant

X

Control access enforcement to block or grant access. [Learn more](#)

- Block access
- Grant access

- Require multifactor authentication** ⓘ

ⓘ Consider testing the new "Require authentication strength". [Learn more](#)

- Require authentication strength** ⓘ

⚠ "Require authentication strength" cannot be used with "Require multifactor authentication". [Learn more](#)

- Require device to be marked as compliant** ⓘ

- Require Microsoft Entra hybrid joined device** ⓘ

- Require approved client app** ⓘ
[See list of approved client apps](#)

- Require app protection policy** ⓘ
[See list of policy protected client apps](#)

- Require password change** ⓘ

For multiple controls

- Require all the selected controls**
- Require one of the selected controls**

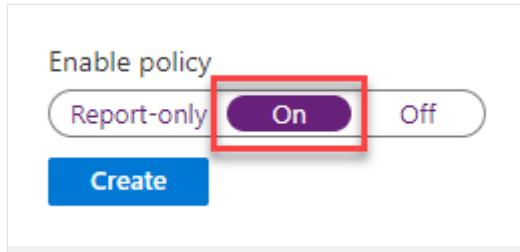
Select

Activate the policy

Conditional Access policies can be set to **Report-only** if you want to see how the configuration would affect users, or **Off** if you don't want to use policy right now.

Because a test group of users is targeted for this tutorial, let's enable the policy, and then test Microsoft Entra multifactor authentication.

1. Under **Enable policy**, select **On**.



2. To apply the Conditional Access policy, select **Create**.

Test Microsoft Entra multifactor authentication

Let's see your Conditional Access policy and Microsoft Entra multifactor authentication in action.

First, sign in to a resource that doesn't require MFA:

1. Open a new browser window in InPrivate or incognito mode and browse to <https://account.activedirectory.windowsazure.com>.

Using a private mode for your browser prevents any existing credentials from affecting this sign-in event.

2. Sign in with your non-administrator test user, such as *testuser*. Be sure to include @ and the domain name for the user account.

If this is the first instance of signing in with this account, you're prompted to change the password. However, there's no prompt for you to configure or use multifactor authentication.

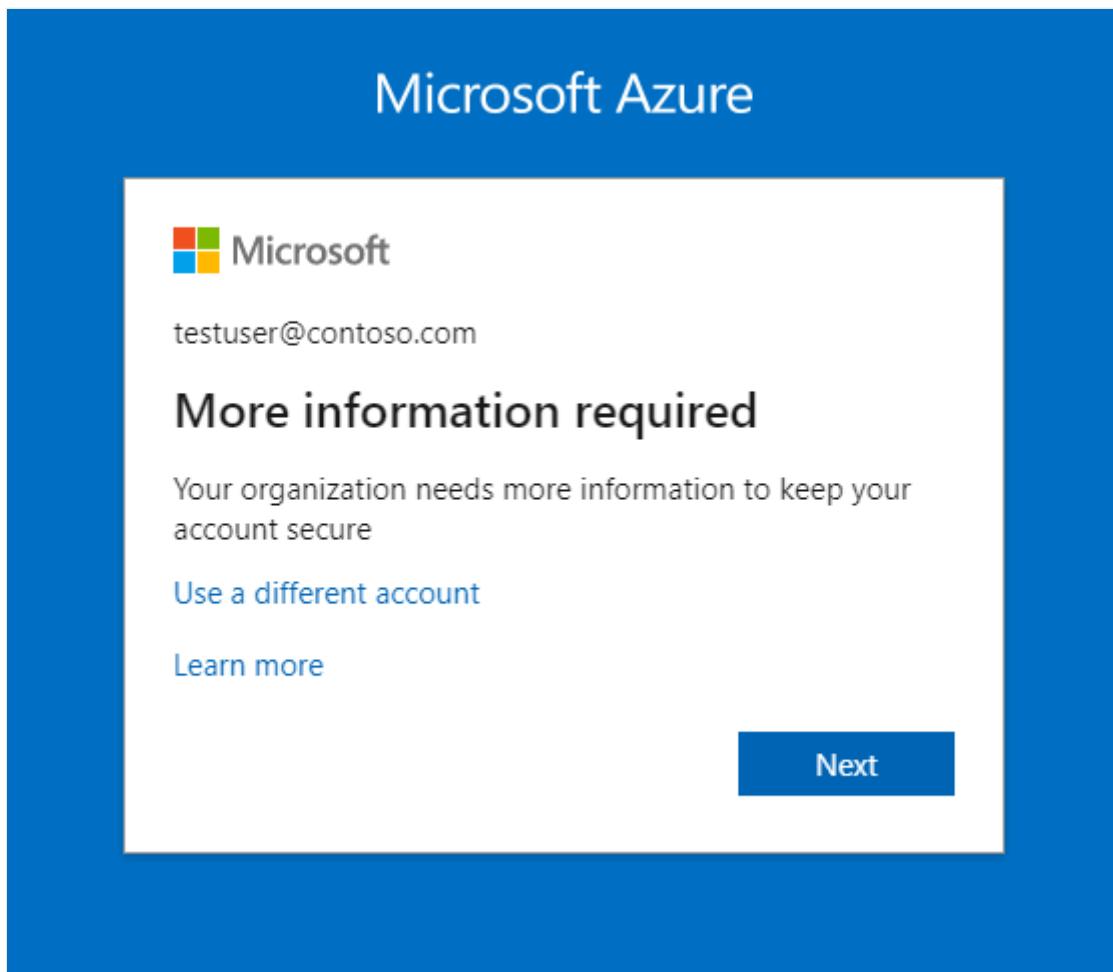
3. Close the browser window.

You configured the Conditional Access policy to require additional authentication for sign in. Because of that configuration, you're prompted to use Microsoft Entra multifactor authentication or to configure a method if you haven't yet done so. Test this new requirement by signing in to the Microsoft Entra admin center:

1. Open a new browser window in InPrivate or incognito mode and sign in to the [Microsoft Entra admin center](#).

2. Sign in with your non-administrator test user, such as *testuser*. Be sure to include @ and the domain name for the user account.

You're required to register for and use Microsoft Entra multifactor authentication.



3. Select **Next** to begin the process.

You can choose to configure an authentication phone, an office phone, or a mobile app for authentication. *Authentication phone* supports text messages and phone calls, *office phone* supports calls to numbers that have an extension, and *mobile app* supports using a mobile app to receive notifications for authentication or to generate authentication codes.

Additional security verification

Secure your account by adding phone verification to your password. [View video to know how to secure your account](#)

Step 1: How should we contact you?

Mobile app

How do you want to use the mobile app?—

- Receive notifications for verification
- Use verification code

To use these verification methods, you must set up the Microsoft Authenticator app.

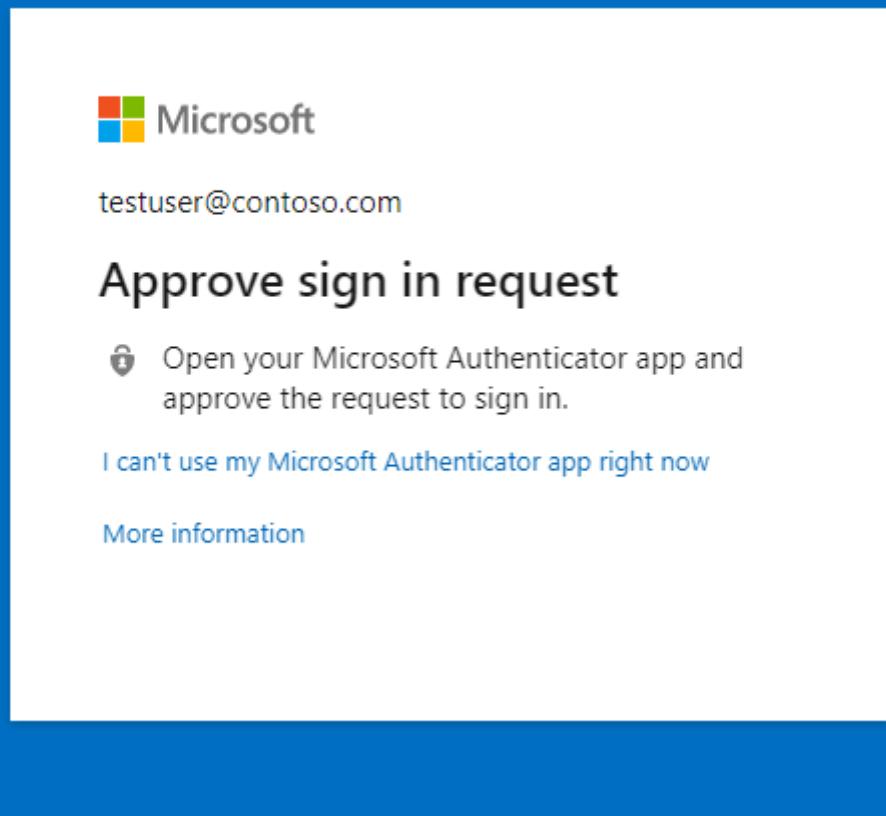
Set up

Please configure the mobile app.

Next

4. Complete the instructions on the screen to configure the method of multifactor authentication that you've selected.
5. Close the browser window, and sign in to the [Microsoft Entra admin center](#) again to test the authentication method that you configured. For example, if you configured a mobile app for authentication, you should see a prompt like the following.

Microsoft Azure



6. Close the browser window.

Clean up resources

If you no longer want to use the Conditional Access policy that you configured as part of this tutorial, delete the policy by using the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Conditional Access Administrator](#).
2. Browse to **Policies > Conditional Access**, and then select the policy that you created, such as **MFA Pilot**.
3. select **Delete**, and then confirm that you want to delete the policy.

A screenshot of the Microsoft Entra admin center interface. The navigation path is Home > Contoso > Security > Conditional Access. Below this, a policy named "MFA Pilot" is listed as a "Conditional Access policy". A red box highlights the "Delete" button, which is located at the bottom left of the policy card.

Next steps

In this tutorial, you enabled Microsoft Entra multifactor authentication by using Conditional Access policies for a selected group of users. You learned how to:

- ✓ Create a Conditional Access policy to enable Microsoft Entra multifactor authentication for a group of Microsoft Entra users.
- ✓ Configure the policy conditions that prompt for multifactor authentication.
- ✓ Test configuring and using multifactor authentication as a user.

[Enable password writeback for self-service password reset \(SSPR\)](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Tutorial: Enable cloud sync self-service password reset writeback to an on-premises environment

Article • 03/04/2025

Microsoft Entra Connect cloud sync can synchronize Microsoft Entra password changes in real time between users in disconnected on-premises Active Directory Domain Services (AD DS) domains. Microsoft Entra Connect cloud sync can run side-by-side with [Microsoft Entra Connect](#) at the domain level to simplify password writeback for additional scenarios, such as users who are in disconnected domains because of a company split or merge. You can configure each service in different domains to target different sets of users depending on their needs. Microsoft Entra Connect cloud sync uses the lightweight Microsoft Entra cloud provisioning agent to simplify the setup for self-service password reset (SSPR) writeback and provide a secure way to send password changes in the cloud back to an on-premises directory.

Prerequisites

- A Microsoft Entra tenant with at least a Microsoft Entra ID P1 or trial license enabled. If needed, [create one for free](#).
- A [Hybrid Identity Administrator](#) account
- Microsoft Entra ID configured for self-service password reset. If needed, complete this tutorial to enable Microsoft Entra SSPR.
- An on-premises AD DS environment configured with [Microsoft Entra Connect cloud sync version 1.1.977.0 or later](#). Learn how to [identify the agent's current version](#). If needed, configure Microsoft Entra Connect cloud sync using [this tutorial](#).

Deployment steps

1. [Configure Microsoft Entra Connect cloud sync service account permissions](#)
2. [Enable password writeback in Microsoft Entra Connect cloud sync](#)
3. [Enable password writeback for SSPR](#)

Configure Microsoft Entra Connect cloud sync service account permissions

Permissions for cloud sync are configured by default. If permissions need to be reset, see [Troubleshooting](#) for more details about the specific permissions required for password writeback and how to set them by using PowerShell.

Enable password writeback in SSPR

You can enable Microsoft Entra Connect cloud sync provisioning directly in the Microsoft Entra admin center or through PowerShell.

Enable password writeback in the Microsoft Entra admin center

With password writeback enabled in Microsoft Entra Connect cloud sync, now verify, and configure Microsoft Entra self-service password reset (SSPR) for password writeback. When you enable SSPR to use password writeback, users who change or reset their password have that updated password synchronized back to the on-premises AD DS environment as well.

To verify and enable password writeback in SSPR, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Hybrid Identity Administrator](#).
2. Browse to **Protection > Password reset**, then choose **On-premises integration**.
3. Check the option for **Enable password write back for synced users**.
4. (optional) If Microsoft Entra Connect provisioning agents are detected, you can additionally check the option for **Write back passwords with Microsoft Entra Connect cloud sync**.
5. Check the option for **Allow users to unlock accounts without resetting their password** to **Yes**.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and various icons for account management. The main title is "Password reset | On-premises integration". Below the title, it says "Woodgrove - Azure Active Directory". The left sidebar has sections for "Diagnose and solve problems", "Manage", "Properties", "Authentication methods", "Registration", "Notifications", "Customization", "On-premises integration" (which is highlighted in grey), and "Administrator Policy". Under "Activity", there are links for "Audit logs" and "Usage & insights". Under "Troubleshooting + Support", there is a link for "New support request". The main content area displays two status cards: "Azure AD Connect sync agent" (Status: Set up complete) and "Azure AD Connect provisioning agent (cloud sync)" (Status: Errors detected). It also shows several configuration checkboxes: "Enable password write back for synced users", "Write back passwords with Azure AD Connect cloud sync", and "Allow users to unlock accounts without resetting their password". A note at the bottom states: "If you uninstall Azure AD Connect provisioning agents from your on-premises servers, please make sure to disable this setting..." with a "Learn more" link. At the bottom right are "Save" and "Discard" buttons.

6. When ready, select **Save**.

PowerShell

With PowerShell you can enable Microsoft Entra Connect cloud sync by using the `Set-AADCloudSyncPasswordWritebackConfiguration` cmdlet on the servers with the provisioning agents.

```
PowerShell

Import-Module 'C:\Program Files\Microsoft Azure AD Connect Provisioning Agent\Microsoft.CloudSync.PowerShell.dll'
Set-AADCloudSyncPasswordWritebackConfiguration -Enable $true -Credential $(Get-Credential)
```

Clean up resources

If you no longer want to use the SSPR writeback functionality you configured as part of this tutorial, complete the following steps:

1. Sign in to the Microsoft Entra admin center [↗](#) as at least a [Hybrid Identity Administrator](#).

2. Browse to **Protection > Password reset**, then choose **On-premises integration**.
3. Uncheck the option for **Enable password write back for synced users**.
4. Uncheck the option for **Write back passwords with Microsoft Entra Connect cloud sync**.
5. Uncheck the option for **Allow users to unlock accounts without resetting their password**.
6. When ready, select **Save**.

If you no longer want to use the Microsoft Entra Connect cloud sync for SSPR writeback functionality but want to continue using Microsoft Entra Connect Sync agent for writebacks complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Hybrid Identity Administrator**.
2. Browse to **Protection > Password reset**, then choose **On-premises integration**.
3. Uncheck the option for **Write back passwords with Microsoft Entra Connect cloud sync**.
4. When ready, select **Save**.

You can also use PowerShell to disable Microsoft Entra Connect cloud sync for SSPR writeback functionality, from your Microsoft Entra Connect cloud sync server, run `Set-AADCloudSyncPasswordWritebackConfiguration` using Hybrid Identity Administrator credentials to disable password writeback with Microsoft Entra Connect cloud sync.

PowerShell

```
Import-Module 'C:\\Program Files\\Microsoft Azure AD Connect Provisioning Agent\\Microsoft.CloudSync.Powershell.dll'
Set-AADCloudSyncPasswordWritebackConfiguration -Enable $false -Credential $(Get-Credential)
```

Supported operations

Passwords are written back in the following situations for end-users and administrators.

[] Expand table

Account	Supported operations
End users	Any end-user self-service voluntary change password operation. Any end-user self-service force change password operation, for example, password expiration. Any end-user self-service password reset that originates from password reset.

Account	Supported operations
Administrators	<p>Any administrator self-service voluntary change password operation.</p> <p>Any administrator self-service force change password operation, for example, password expiration.</p> <p>Any administrator self-service password reset that originates from password reset.</p> <p>Any administrator-initiated end-user password reset from the Microsoft Entra admin center.</p> <p>Any administrator-initiated end-user password reset from the Microsoft Graph API.</p>

Unsupported operations

Passwords aren't written back in the following situations.

[Expand table](#)

Account	Unsupported operations
End users	<p>Any end user resetting their own password by using PowerShell cmdlets or the Microsoft Graph API.</p>
Administrators	<p>Any administrator-initiated end-user password reset by using PowerShell cmdlets.</p> <p>Any administrator-initiated end-user password reset from the Microsoft 365 admin center.</p> <p>Any administrator can't use password reset tool to reset their own password, or any other Administrator in Microsoft Entra ID for password writeback.</p>

Validation scenarios

Try the following operations to validate scenarios using password writeback. All validation scenarios require cloud sync is installed and the user is in scope for password writeback.

[Expand table](#)

Scenario	Details
Reset password from the sign-in page	<p>Have two users from disconnected domains and forests perform SSPR. You could also have Microsoft Entra Connect and cloud sync deployed side-by-side and have one user in the scope of cloud sync configuration and another in scope of Microsoft Entra Connect and have those users reset their password.</p>

Scenario	Details
Force expired password change	Have two users from disconnected domains and forests change expired passwords. You could also have Microsoft Entra Connect and cloud sync deployed side-by-side and have one user in the scope of cloud sync configuration and another in scope of Microsoft Entra Connect.
Regular password change	Have two users from disconnected domains and forests perform routine password change. You could also have Microsoft Entra Connect and cloud sync side by side and have one user in the scope of cloud sync config and another in scope of Microsoft Entra Connect.
Admin reset user password	Have two users disconnected domains and forests reset their password from the Microsoft Entra admin center or Frontline worker portal. You could also have Microsoft Entra Connect and cloud sync side by side and have one user in the scope of cloud sync config and another in scope of Microsoft Entra Connect
Self-service account unlock	Have two users from disconnected domains and forests unlock accounts in the SSPR portal resetting the password. You could also have Microsoft Entra Connect and cloud sync side by side and have one user in the scope of cloud sync config and another in scope of Microsoft Entra Connect.

Troubleshooting

- The Microsoft Entra Connect cloud sync group Managed Service Account should have the following permissions set to writeback the passwords by default:
 - Reset password
 - Write permissions on lockoutTime
 - Write permissions on pwdLastSet
 - Extended rights for "Unexpire Password" on the root object of each domain in that forest, if not already set.

If these permissions aren't set, you can set the PasswordWriteBack permission on the service account by using the Set-AADCloudSyncPermissions cmdlet and on-premises enterprise administrator credentials:

PowerShell

```
Import-Module 'C:\\Program Files\\Microsoft Azure AD Connect Provisioning Agent\\Microsoft.CloudSync.Powershell.dll'
Set-AADCloudSyncPermissions -PermissionType PasswordWriteBack -EACredential $(Get-Credential)
```

After you updated the permissions, it can take up to an hour or more for these permissions to replicate to all the objects in your directory.

- If passwords for some user accounts aren't written back to the on-premises directory, make sure that inheritance isn't disabled for the account in the on-premises AD DS environment. Write permissions for passwords must be applied to descendant objects for the feature to work correctly.
- Password policies in the on-premises AD DS environment can prevent password resets from being correctly processed. If you're testing this feature and want to reset password for users more than once per day, the group policy for Minimum password age must be set to 0. This setting can be found under Computer Configuration > Policies > Windows Settings > Security Settings > Account Policies > Password Policy within gpmc.msc.
- If you update the group policy, wait for the updated policy to replicate, or use the gpupdate /force command.
- For passwords to be changed immediately, Minimum password age must be set to 0. However, if users adhere to the on-premises policies, and the Minimum password age is set to a value greater than zero, password writeback won't work after the on-premises policies are evaluated.

For more information about how to validate or set up the appropriate permissions, see [Configure account permissions for Microsoft Entra Connect](#).

Next steps

- For more information about cloud sync and a comparison between Microsoft Entra Connect and cloud sync, see [What is Microsoft Entra Connect cloud sync?](#)
- For a tutorial about setting up password writeback by using Microsoft Entra Connect, see [Tutorial: Enable Microsoft Entra self-service password reset writeback to an on-premises environment](#).

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Tutorial: Enable Microsoft Entra self-service password reset writeback to an on-premises environment

Article • 03/04/2025

With Microsoft Entra self-service password reset (SSPR), users can update their password or unlock their account using a web browser. We recommend this video on [How to enable and configure SSPR in Microsoft Entra ID](#). In a hybrid environment where Microsoft Entra ID is connected to an on-premises Active Directory Domain Services (AD DS) environment, this scenario can cause passwords to be different between the two directories.

Password writeback can be used to synchronize password changes in Microsoft Entra back to your on-premises AD DS environment. Microsoft Entra Connect provides a secure mechanism to send these password changes back to an existing on-premises directory from Microsoft Entra ID.

ⓘ Important

This tutorial shows an administrator how to enable self-service password reset back to an on-premises environment. If you're an end user already registered for self-service password reset and need to get back into your account, go to <https://aka.ms/sspr>.

If your IT team hasn't enabled the ability to reset your own password, reach out to your helpdesk for additional assistance.

In this tutorial, you learn how to:

- ✓ Configure the required permissions for password writeback
- ✓ Enable the password writeback option in Microsoft Entra Connect
- ✓ Enable password writeback in Microsoft Entra SSPR

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- A working Microsoft Entra tenant with at least a Microsoft Entra ID P1 or trial license enabled.

- If needed, [create one for free ↗](#).
- For more information, see [Licensing requirements for Microsoft Entra SSPR](#).
- An account with [Hybrid Identity Administrator](#).
- Microsoft Entra ID configured for self-service password reset.
 - If needed, [complete the previous tutorial to enable Microsoft Entra SSPR](#).
- An existing on-premises AD DS environment configured with a current version of Microsoft Entra Connect.
 - If needed, configure Microsoft Entra Connect using the [Express](#) or [Custom](#) settings.
 - To use password writeback, domain controllers can run any supported version of Windows Server.

Configure account permissions for Microsoft Entra Connect

Microsoft Entra Connect lets you synchronize users, groups, and credential between an on-premises AD DS environment and Microsoft Entra ID. You typically install Microsoft Entra Connect on a Windows Server 2016 or later computer that's joined to the on-premises AD DS domain.

To correctly work with SSPR writeback, the account specified in Microsoft Entra Connect must have the appropriate permissions and options set. If you're not sure which account is currently in use, open Microsoft Entra Connect and select the [View current configuration](#) option. The account that you need to add permissions to is listed under **Synchronized Directories**. The following permissions and options must be set on the account:

- **Reset password**
- **Change password**
- **Write permissions on** `lockoutTime`
- **Write permissions on** `pwdLastSet`
- **Extended rights** for "Unexpire Password" on the root object of *each domain* in that forest, if not already set.

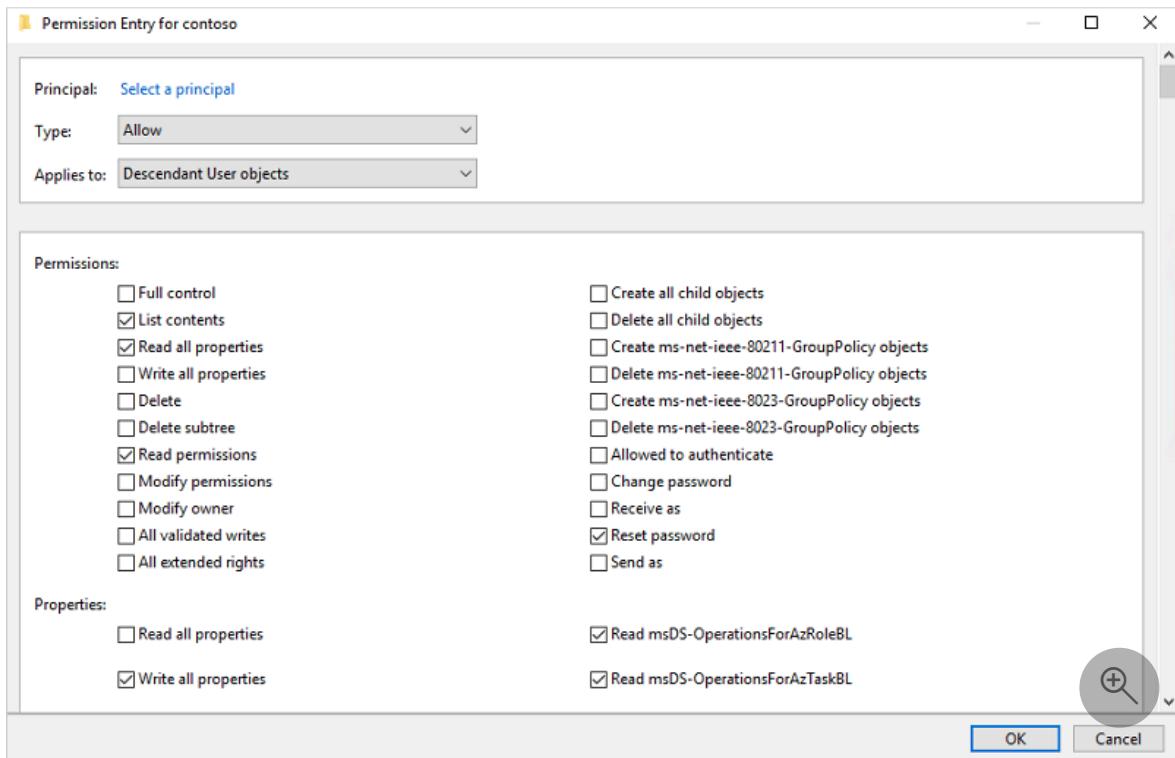
If you don't assign these permissions, writeback may appear to be configured correctly, but users encounter errors when they manage their on-premises passwords from the cloud. When setting "Unexpire Password" permissions in Active Directory, it must be applied to **This object and all descendant objects**, **This object only**, or **All descendant objects**, or the "Unexpire Password" permission can't be displayed.

Tip

If passwords for some user accounts aren't written back to the on-premises directory, make sure that inheritance isn't disabled for the account in the on-prem AD DS environment. Write permissions for passwords must be applied to descendant objects for the feature to work correctly.

To set up the appropriate permissions for password writeback to occur, complete the following steps:

1. In your on-premises AD DS environment, open **Active Directory Users and Computers** with an account that has the appropriate *domain administrator* permissions.
2. From the **View** menu, make sure that **Advanced features** are turned on.
3. In the left panel, right-select the object that represents the root of the domain and select **Properties > Security > Advanced**.
4. From the **Permissions** tab, select **Add**.
5. For **Principal**, select the account that permissions should be applied to (the account used by Microsoft Entra Connect).
6. In the **Applies to** drop-down list, select **Descendant User objects**.
7. Under **Permissions**, select the box for the following option:
 - **Reset password**
8. Under **Properties**, select the boxes for the following options. Scroll through the list to find these options, which may already be set by default:
 - **Write lockoutTime**
 - **Write pwdLastSet**



1. When ready, select **Apply / OK** to apply the changes.
2. From the **Permissions** tab, select **Add**.
3. For **Principal**, select the account that permissions should be applied to (the account used by Microsoft Entra Connect).
4. In the **Applies to** drop-down list, select **This object and all descendant objects**
5. Under *Permissions*, select the box for the following option:
 - **Unexpire Password**
6. When ready, select **Apply / OK** to apply the changes and exit any open dialog boxes.

When you update permissions, it might take up to an hour or more for these permissions to replicate to all the objects in your directory.

Password policies in the on-premises AD DS environment may prevent password resets from being correctly processed. For password writeback to work most efficiently, the group policy for *Minimum password age* must be set to 0. This setting can be found under **Computer Configuration > Policies > Windows Settings > Security Settings > Account Policies** within `gpmc.msc`.

If you update the group policy, wait for the updated policy to replicate, or use the `gpupdate /force` command.

 **Note**

If you need to allow users to change or reset passwords more than one time per day, *Minimum password age* must be set to 0. Password writeback will work after on-premises password policies are successfully evaluated.

Enable password writeback in Microsoft Entra Connect

One of the configuration options in Microsoft Entra Connect is for password writeback. When this option is enabled, password change events cause Microsoft Entra Connect to synchronize the updated credentials back to the on-premises AD DS environment.

To enable SSPR writeback, first enable the writeback option in Microsoft Entra Connect. From your Microsoft Entra Connect server, complete the following steps:

1. Sign in to your Microsoft Entra Connect server and start the **Microsoft Entra Connect** configuration wizard.
2. On the **Welcome** page, select **Configure**.
3. On the **Additional tasks** page, select **Customize synchronization options**, and then select **Next**.
4. On the **Connect to Microsoft Entra ID** page, enter a Hybrid Administrator credential for your Azure tenant, and then select **Next**.
5. On the **Connect directories and Domain/OU filtering** pages, select **Next**.
6. On the **Optional features** page, select the box next to **Password writeback** and select **Next**.
7. On the **Directory extensions** page, select **Next**.
8. On the **Ready to configure** page, select **Configure** and wait for the process to finish.
9. When you see the configuration finish, select **Exit**.

ⓘ Note

Updating `PasswordWritebackEnabled` from [OnPremDirectorySynchronization service features](#) is not supported as this feature flag is not in use.

Enable password writeback for SSPR

With password writeback enabled in Microsoft Entra Connect, now configure Microsoft Entra SSPR for writeback. SSPR can be configured to writeback through Microsoft Entra Connect Sync agents and Microsoft Entra Connect provisioning agents (cloud sync).

When you enable SSPR to use password writeback, users who change or reset their password have that updated password synchronized back to the on-premises AD DS environment as well.

To enable password writeback in SSPR, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as a **Global Administrator**.
2. Browse to **Protection > Password reset**, then choose **On-premises integration**.
3. Check the option for **Write back passwords to your on-premises directory**.
4. (optional) If Microsoft Entra Connect provisioning agents are detected, you can additionally check the option for **Write back passwords with Microsoft Entra Connect cloud sync**.
5. Check the option for **Allow users to unlock accounts without resetting their password** to Yes.
6. When ready, select **Save**.

Clean up resources

If you no longer want to use the SSPR writeback functionality you have configured as part of this tutorial, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as a **Global Administrator**.
2. Browse to **Protection > Password reset**, then choose **On-premises integration**.
3. Uncheck the option for **Write back passwords to your on-premises directory**.
4. Uncheck the option for **Write back passwords with Microsoft Entra Connect cloud sync**.
5. Uncheck the option for **Allow users to unlock accounts without resetting their password**.
6. When ready, select **Save**.

If you no longer want to use the Microsoft Entra Connect cloud sync for SSPR writeback functionality but want to continue using Microsoft Entra Connect Sync agent for writebacks complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as a **Global Administrator**.
2. Browse to **Protection > Password reset**, then choose **On-premises integration**.
3. Uncheck the option for **Write back passwords with Microsoft Entra Connect cloud sync**.
4. When ready, select **Save**.

If you no longer want to use any password functionality, complete the following steps from your Microsoft Entra Connect server:

1. Sign in to your Microsoft Entra Connect server and start the **Microsoft Entra Connect** configuration wizard.
2. On the **Welcome** page, select **Configure**.
3. On the **Additional tasks** page, select **Customize synchronization options**, and then select **Next**.
4. On the **Connect to Microsoft Entra ID** page, enter a Hybrid Administrator credential, and then select **Next**.
5. On the **Connect directories and Domain/OU filtering** pages, select **Next**.
6. On the **Optional features** page, deselect the box next to **Password writeback** and select **Next**.
7. On the **Ready to configure** page, select **Configure** and wait for the process to finish.
8. When you see the configuration finish, select **Exit**.

 **Important**

Enabling password writeback for the first time may trigger password change events 656 and 657, even if a password change has not occurred. This is because all password hashes are re-synchronized after a password hash synchronization cycle has run.

Next steps

In this tutorial, you enabled Microsoft Entra SSPR writeback to an on-premises AD DS environment. You learned how to:

- ✓ Configure the required permissions for password writeback
- ✓ Enable the password writeback option in Microsoft Entra Connect
- ✓ Enable password writeback in Microsoft Entra SSPR

[Evaluate risk at sign in](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Tutorial: Configure custom banned passwords for Microsoft Entra password protection

Article • 03/04/2025

Users often create passwords that use common local words such as a school, sports team, or famous person. These passwords are easy to guess, and weak against dictionary-based attacks. To enforce strong passwords in your organization, the Microsoft Entra custom banned password list lets you add specific strings to evaluate and block. A password change request fails if there's a match in the custom banned password list.

In this tutorial you learn how to:

- ✓ Enable custom banned passwords
- ✓ Add entries to the custom banned password list
- ✓ Test password changes with a banned password

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- A working Microsoft Entra tenant with at least a Microsoft Entra ID P1 or trial license enabled.
 - If needed, [create one for free ↗](#).
- An account with at least the [Authentication Policy Administrator](#) role.
- A non-administrator user with a password you know, such as *testuser*. You test a password change event using this account in this tutorial.
 - If you need to create a user, see [Quickstart: Add new users to Microsoft Entra ID](#).
 - To test the password change operation using a banned password, the Microsoft Entra tenant must be [configured for self-service password reset](#).

What are banned password lists?

Microsoft Entra ID includes a global banned password list. The contents of the global banned password list isn't based on any external data source. Instead, the global banned password list is based on the ongoing results of Microsoft Entra security telemetry and analysis. When a user or administrator tries to change or reset their

credentials, the desired password is checked against the list of banned passwords. The password change request fails if there's a match in the global banned password list. You can't edit this default global banned password list.

To give you flexibility in what passwords are allowed, you can also define a custom banned password list. The custom banned password list works alongside the global banned password list to enforce strong passwords in your organization. Organizational-specific terms can be added to the custom banned password list, such as the following examples:

- Brand names
- Product names
- Locations, such as company headquarters
- Company-specific internal terms
- Abbreviations that have specific company meaning
- Months and weekdays with your company's local languages

When a user attempts to reset a password to something that's on the global or custom banned password list, they see one of the following error messages:

- *Unfortunately, your password contains a word, phrase, or pattern that makes your password easily guessable. Please try again with a different password.*
- *Unfortunately, you can't use that password because it contains words or characters that have been blocked by your administrator. Please try again with a different password.*

The custom banned password list is limited to a maximum of 1000 terms. It's not designed for blocking large lists of passwords. To maximize the benefits of the custom banned password list, review the [custom banned password list concepts](#) and [password evaluation algorithm overview](#).

Configure custom banned passwords

Let's enable the custom banned password list and add some entries. You can add additional entries to the custom banned password list at any time.

To enable the custom banned password list and add entries to it, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods**, then **Password protection**.

3. Set the option for **Enforce custom list** to **Yes**.
4. Add strings to the **Custom banned password list**, one string per line. The following considerations and limitations apply to the custom banned password list:

- The custom banned password list can contain up to 1000 terms.
- The custom banned password list is case-insensitive.
- The custom banned password list considers common character substitution, such as "o" and "0", or "a" and "@".
- The minimum string length is four characters, and the maximum is 16 characters.

Specify your own custom passwords to ban, as shown in the following example

Authentication methods | Password protection

Contoso - Microsoft Entra ID Security

Manage

- Policies
- Password protection**
- Registration campaign
- Authentication strengths
- Settings

Custom smart lockout

Lockout threshold: 10

Lockout duration in seconds: 60

Custom banned passwords

Enforce custom list: Yes

Custom banned password list:

- contoso
- fabrikam
- tailwind
- michigan
- wolverine
- harbaugh
- howard

Password protection for Windows Server Active Directory

Enable password protection on Windows Server Active Directory: No

Mode: Enforced

5. Leave the option for **Enable password protection on Windows Server Active Directory** to **No**.
6. To enable the custom banned passwords and your entries, select **Save**.

It may take several hours for updates to the custom banned password list to be applied.

For a hybrid environment, you can also [deploy Microsoft Entra password protection to an on-premises environment](#). The same global and custom banned password lists are used for both cloud and on-premises password change requests.

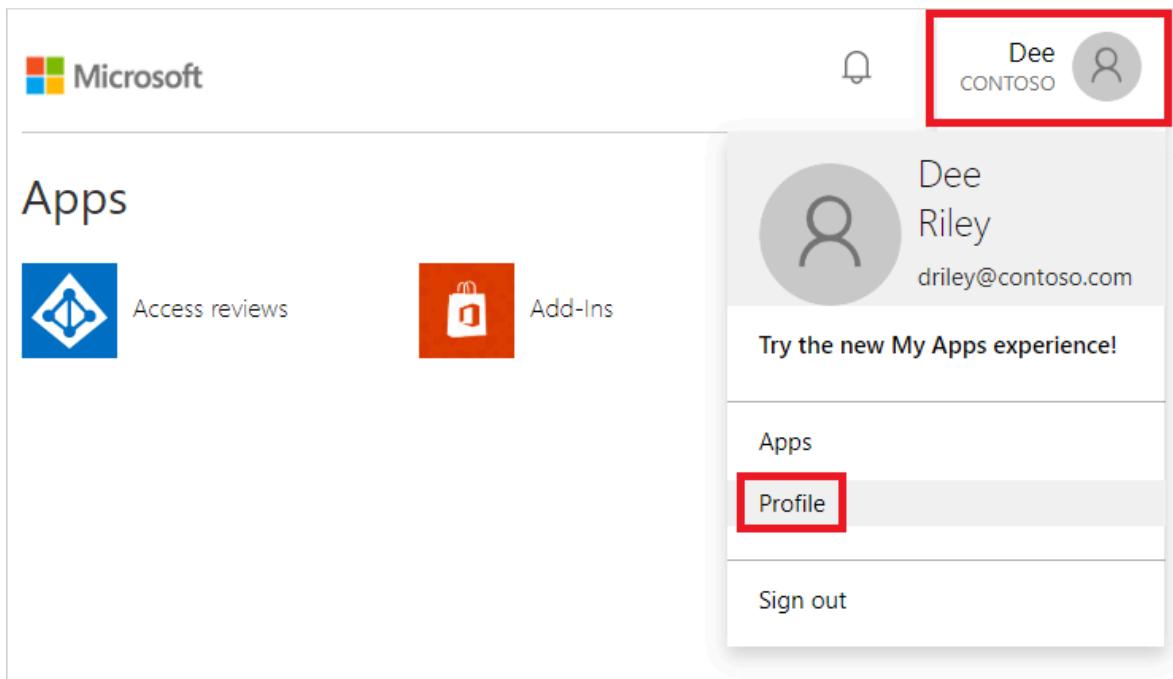
Test custom banned password list

To see the custom banned password list in action, try to change the password to a variation of one that you added in the previous section. When Microsoft Entra ID tries to process the password change, the password is matched against an entry in the custom banned password list. An error is then displayed to the user.

ⓘ Note

Before a user can reset their password in the web-based portal, the Microsoft Entra tenant must be [configured for self-service password reset](#). If needed, the user can then [register for SSPR at https://aka.ms/ssprsetup](#).

1. Go to the **My Apps** page at <https://myapps.microsoft.com>.
2. In the top-right corner, select your name, then choose **Profile** from the drop-down menu.



3. On the **Profile** page, select **Change password**.
4. On the **Change password** page, enter the existing (old) password. Enter and confirm a new password that's on the custom banned password list you defined in the previous section, then select **Submit**.
5. An error message is returned that tells you the password has been blocked by the administrator, as shown in the following example:

change password

Strong password required. Enter 8-256 characters. Do not include common words or names. Combine uppercase letters, lowercase letters, numbers, and symbols.

User ID

driley@contoso.com

Old password

Create new password

Password strength

Unfortunately, you can't use that password because it contains words or characters that have been blocked by your administrator. Please try again with a different password.

Confirm new password

submit

cancel

Clean up resources

If you no longer want to use the custom banned password list you have configured as part of this tutorial, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods**, then **Password protection**.
3. Set the option for **Enforce custom list** to **No**.
4. To update the custom banned password configuration, select **Save**.

Next steps

In this tutorial, you enabled and configured custom password protection lists for Microsoft Entra ID. You learned how to:

- ✓ Enable custom banned passwords
- ✓ Add entries to the custom banned password list
- ✓ Test password changes with a banned password

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Tutorial: Use risk detections for user sign-ins to trigger Microsoft Entra multifactor authentication or password changes

Article • 03/04/2025

To protect your users, you can configure risk-based Microsoft Entra Conditional Access policies that automatically respond to risky behaviors. These policies can automatically block a sign-in attempt or require extra action, such as require a secure password change or prompting for Microsoft Entra multifactor authentication. These policies work with existing Microsoft Entra Conditional Access policies as an extra layer of protection for your organization. Users might never trigger a risky behavior in one of these policies, but your organization is protected if an attempt to compromise your security is made.

Important

This tutorial shows an administrator how to enable risk-based multifactor authentication (MFA).

If your IT team hasn't enabled the ability to use Microsoft Entra multifactor authentication or you have problems during sign-in, reach out to your helpdesk for additional assistance.

In this tutorial, you learn how to:

- ✓ Understand the available policies
- ✓ Enable Microsoft Entra multifactor authentication registration
- ✓ Enable risk-based password changes
- ✓ Enable risk-based multifactor authentication
- ✓ Test risk-based policies for user sign-in attempts

Prerequisites

To complete this tutorial, you need the following resources and privileges:

- A working Microsoft Entra tenant with at least a Microsoft Entra ID P2 or trial license enabled.
 - If needed, [create one for free ↗](#).

- An account with Security Administrator privileges.
- Microsoft Entra ID configured for self-service password reset and Microsoft Entra multifactor authentication
 - If needed, [complete the tutorial to enable Microsoft Entra SSPR](#).
 - If needed, [complete the tutorial to enable Microsoft Entra multifactor authentication](#).

Overview of Microsoft Entra ID Protection

Each day, Microsoft collects and analyses trillions of anonymized signals as part of user sign-in attempts. These signals help build patterns of good user sign-in behavior, and identify potential risky sign-in attempts. Microsoft Entra ID Protection can review user sign-in attempts and take extra action if there's suspicious behavior:

Some of the following actions might trigger Microsoft Entra ID Protection risk detection:

- Users with leaked credentials.
- Sign-ins from anonymous IP addresses.
- Impossible travel to atypical locations.
- Sign-ins from infected devices.
- Sign-ins from IP addresses with suspicious activity.
- Sign-ins from unfamiliar locations.

This article guides you through enabling three policies to protect users and automate the response to suspicious activity.

- Multifactor authentication registration policy
 - Makes sure users are registered for Microsoft Entra multifactor authentication. If a sign-in risk policy prompts for MFA, the user must already be registered for Microsoft Entra multifactor authentication.
- User risk policy
 - Identifies and automates response to user accounts that might have compromised credentials. Can prompt the user to create a new password.
- Sign in risk policy
 - Identifies and automates response to suspicious sign-in attempts. Can prompt the user to provide extra forms of verification using Microsoft Entra multifactor authentication.

When you enable a risk-based policy, you can also choose the threshold for risk level - *low*, *medium*, or *high*. This flexibility lets you decide how aggressive you want to be in enforcing any controls for suspicious sign-in events. Microsoft recommends the following policy configurations.

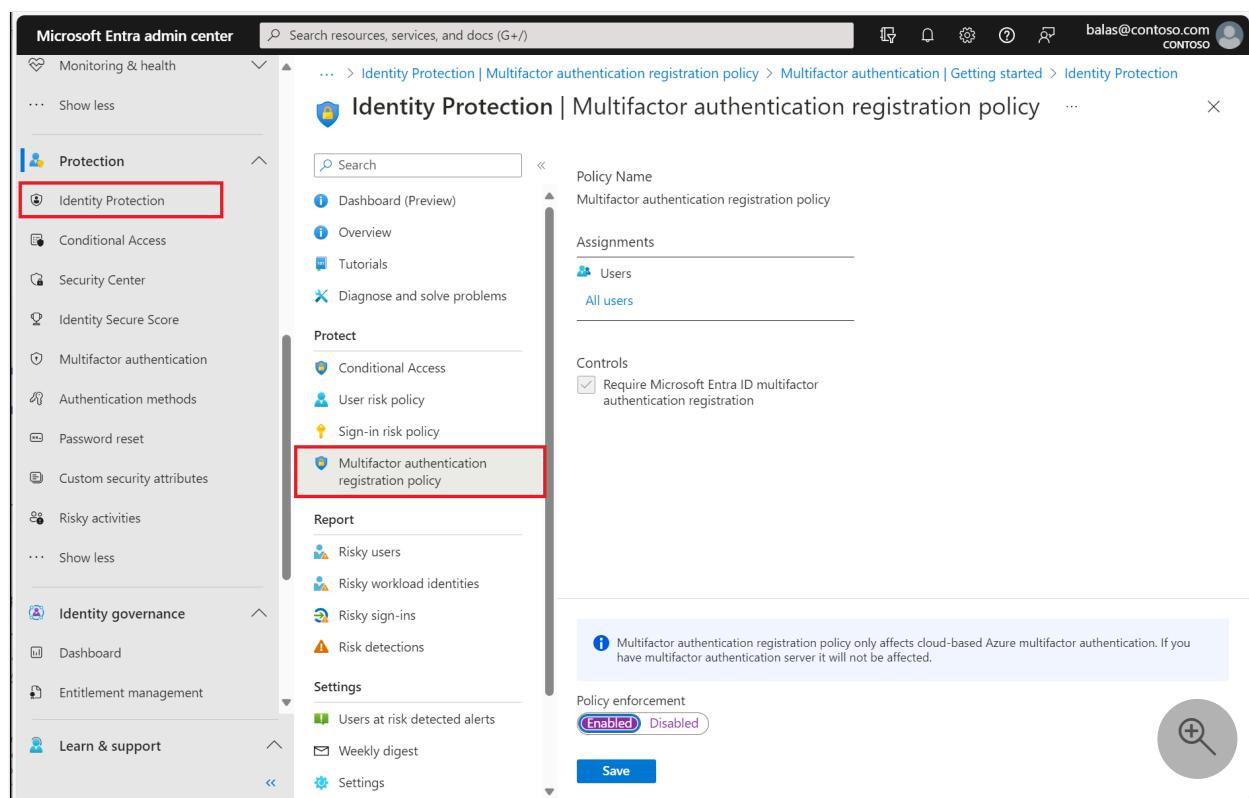
For more information about Microsoft Entra ID Protection, see [What is Microsoft Entra ID Protection?](#)

Enable multifactor authentication registration policy

Microsoft Entra ID Protection includes a default policy that can help get users registered for Microsoft Entra multifactor authentication. If you use other policies to protect sign-in events, you would need users to have already registered for MFA. When you enable this policy, it doesn't require users to perform MFA at each sign-in event. The policy only checks the registration status for a user and asks them to preregister if needed.

It's recommended to enable this registration policy for users that use multifactor authentication. To enable this policy, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Security Administrator](#).
2. Browse to **Protection > Identity Protection > Multifactor authentication registration policy**.
3. By default, the policy applies to *All users*. If desired, select **Assignments**, then choose the users or groups to apply the policy on.
4. Under **Controls**, select **Access**. Make sure the option for *Require Microsoft Entra multifactor authentication registration* is checked, then choose **Select**.
5. Set **Enforce Policy** to *On*, then select **Save**.



Enable user risk policy for password change

Microsoft works with researchers, law enforcement, various security teams at Microsoft, and other trusted sources to find username and password pairs. When one of these pairs matches an account in your environment, a risk-based password change can be requested. This policy and action requires the user update their password before they can sign in to make sure any previously exposed credentials no longer work.

To enable this policy, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Conditional Access Administrator**.
2. Browse to **Protection > Conditional Access**.
3. Select **New policy**.
4. Give your policy a name. We recommend that organizations create a meaningful standard for the names of their policies.
5. Under **Assignments**, select **Users or workload identities**.
 - a. Under **Include**, select **All users**.
 - b. Under **Exclude**, select **Users and groups** and choose your organization's emergency access or break-glass accounts.
 - c. Select **Done**.
6. Under **Cloud apps or actions > Include**, select **All resources (formerly 'All cloud apps')**.
7. Under **Conditions > User risk**, set **Configure** to **Yes**.
 - a. Under **Configure user risk levels needed for policy to be enforced**, select **High**.
This guidance is based on Microsoft recommendations and might be different for each organization
 - b. Select **Done**.
8. Under **Access controls > Grant**, select **Grant access**.
 - a. Select **Require authentication strength**, then select the built-in **Multifactor authentication** authentication strength from the list.
 - b. Select **Require password change**.
 - c. Select **Select**.
9. Under **Session**.
 - a. Select **Sign-in frequency**.
 - b. Ensure **Every time** is selected.
 - c. Select **Select**.
10. Confirm your settings and set **Enable policy** to **Report-only**.
11. Select **Create** to create to enable your policy.

After administrators confirm the settings using [report-only mode](#), they can move the **Enable policy** toggle from **Report-only** to **On**.

Passwordless scenarios

For organizations that adopt [passwordless authentication methods](#) make the following changes:

Update your passwordless user risk policy

1. Under Users:

- a. **Include**, select **Users and groups** and target your passwordless users.

2. Under Access controls > Block access for passwordless users.

Tip

You might need to have two policies for a period of time while deploying passwordless methods.

- One that allows self-remediation for those not using passwordless methods.
- Another that blocks passwordless users at high risk.

Remediate and unblock passwordless user risk

1. Require administrator [investigation and remediation](#) of any risk.
2. Unblock the user.

Enable sign-in risk policy for MFA

Most users have a normal behavior that can be tracked. When they fall outside of this norm, it could be risky to allow them to successfully sign in. Instead, you might want to block that user, or ask them to perform a multifactor authentication. If the user successfully completes the MFA challenge, you can consider it a valid sign-in attempt and grant access to the application or service.

To enable this policy, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Conditional Access Administrator](#).
2. Browse to **Protection > Conditional Access**.
3. Select **New policy**.
4. Give your policy a name. We recommend that organizations create a meaningful standard for the names of their policies.

5. Under **Assignments**, select **Users or workload identities**.
 - a. Under **Include**, select **All users**.
 - b. Under **Exclude**, select **Users and groups** and choose your organization's emergency access or break-glass accounts.
 - c. Select **Done**.
6. Under **Cloud apps or actions > Include**, select **All resources** (formerly 'All cloud apps').
7. Under **Conditions > Sign-in risk**, set **Configure** to **Yes**.
 - a. Under **Select the sign-in risk level this policy will apply to**, select **High** and **Medium**. **This guidance is based on Microsoft recommendations and might be different for each organization**
 - b. Select **Done**.
8. Under **Access controls > Grant**, select **Grant access**.
 - a. Select **Require authentication strength**, then select the built-in **Multifactor authentication** authentication strength from the list.
 - b. Select **Select**.
9. Under **Session**.
 - a. Select **Sign-in frequency**.
 - b. Ensure **Every time** is selected.
 - c. Select **Select**.
10. Confirm your settings and set **Enable policy** to **Report-only**.
11. Select **Create** to create to enable your policy.

After administrators confirm the settings using [report-only mode](#), they can move the **Enable policy** toggle from **Report-only** to **On**.

Passwordless scenarios

For organizations that adopt [passwordless authentication methods](#) make the following changes:

Update your passwordless sign-in risk policy

1. Under **Users**:
 - a. **Include**, select **Users and groups** and target your passwordless users.
 - b. Under **Exclude**, select **Users and groups** and choose your organization's emergency access or break-glass accounts.
 - c. Select **Done**.
2. Under **Cloud apps or actions > Include**, select **All resources** (formerly 'All cloud apps').
3. Under **Conditions > Sign-in risk**, set **Configure** to **Yes**.

- a. Under **Select the sign-in risk level this policy will apply to**, select **High** and **Medium**. For more information on risk levels, see [Choosing acceptable risk levels](#).
 - b. Select **Done**.
4. Under **Access controls > Grant**, select **Grant access**.
 - a. Select **Require authentication strength**, then select the built-in **Passwordless MFA** or **Phishing-resistant MFA** based on which method the targeted users have.
 - b. Select **Select**.
 5. Under **Session**:
 - a. Select **Sign-in frequency**.
 - b. Ensure **Every time** is selected.
 - c. Select **Select**.

Test risky sign events

Most user sign-in events don't trigger the risk-based policies configured in the previous steps. A user might never see a prompt for MFA or to reset their password. If their credentials remain secure and their behavior consistent, their sign-in events would be successful.

To test the Microsoft Entra ID Protection policies created in the previous steps, you need a way to simulate risky behavior or potential attacks. The steps to do these tests vary based on the Microsoft Entra ID Protection policy you want to validate. For more information on scenarios and steps, see [Simulate risk detections in Microsoft Entra ID Protection](#).

Clean up resources

If you complete your testing and no longer want to have the risk-based policies enabled, return to each policy you want to disable and set **Enable policy** to *Off* or delete them.

Next steps

In this tutorial, you enabled risk-based user policies for Microsoft Entra ID Protection. You learned how to:

- ✓ Understand the available policies for Microsoft Entra ID Protection
- ✓ Enable Microsoft Entra multifactor authentication registration

- ✓ Enable risk-based password changes
- ✓ Enable risk-based multifactor authentication
- ✓ Test risk-based policies for user sign-in attempts

[Learn more about Microsoft Entra ID Protection](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Tutorial: Enable security notifications for audit log events

Article • 03/04/2025

In this tutorial, you learn how to create an [Azure Logic App](#) that monitors Microsoft Entra audit logs. A logic app can send a security email notification to users based on different audit log events.

This tutorial focuses on security notifications that get emailed when there's a change to a user's authentication methods. You can also use logic apps to create workflows that send security notifications for other audit log events. These security notifications help update users and notify them of any risky activity. Users can quickly take the correct steps to report it.

You recently changed your authentication methods

We have been notified of the following action: Reset password (by admin) on 2023-07-27 P17:58:42.

If you initiated this, no action is required.

If you haven't, please report it now.

Instructions

1. Review your account activity in [Microsoft Security Info](#).
2. If you do not recognize this action, report it immediately:
 - Go to [ReportItNow](#) and select your security event.
 - Provide any additional information in the form and submit.

Information and Support

- Technical Assistance - Contact [Helpdesk](#) support services

Do NOT reply to this email. This is an unmonitored mailbox.

For more information, contact the [Security Department](#)

[Report device](#)

Contoso, Ltd., 4567 Main St Buffalo, NY 98052

Facilitated by



Prerequisites

To use this feature, you need:

- An Azure subscription. If you don't have an Azure subscription, you can [sign up for a free trial ↗](#).

- A Microsoft Entra tenant.
- A user who's at least a [Security Administrator](#) for the Microsoft Entra tenant.
- An Event Hubs namespace and an event hub in your Azure subscription. Learn how to [create an event hub](#).
- Enable logs to be streamed to the event hub. Learn how to [stream logs to an event hub](#). Only select the logs that you want the security notification to be sent for. For this tutorial, we'll stream Audit Logs.
- An email account from a service that works with Azure Logic Apps, such as Office 365 Outlook or Outlook.com. For other supported email providers, review [Connectors for Azure Logic Apps](#).

Create a logic app

1. Sign in to the Azure portal.
2. In the home page, under **Azure services**, select **Logic Apps**.
3. Select **Add**.
4. In **Create Logic App**, configure your logic app:
 - a. Select the **Subscription** in which you want to create the logic app.
 - b. Select the **Resource Group** you created for the event hub.
 - c. Enter the **Logic App name**, and the system immediately checks to see if the name is available.
 - d. Select a **Region** for the logic app.
 - e. For **Plan type**, select the **Consumption** tier. Choose a region and plan type that aligns with your organization's size and needs. To learn about differences between tiers, see the [Standard and Consumption logic app workflow](#).
 - f. Don't change any other settings.

Note

Only some regions support Zone redundancy. Depending on your location, your Zone redundancy section might be automatically enabled or disabled. For more information, see [Protect logic apps from region failures with zone redundancy and availability zones](#).

The screenshot shows the 'Create Logic App' wizard in the Microsoft Azure portal. The top navigation bar includes 'Microsoft Azure', a search bar, and a 'Search resources, services, and docs (G+ /)' button. The breadcrumb trail shows 'Home > Logic apps > Create Logic App'. The current step is 'Basics'. Other tabs include 'Hosting', 'Networking', 'Monitoring', 'Tags', and 'Review + create'. A descriptive text explains what a logic app is: 'Create a logic app, which lets you group workflows as a logical unit for easier management, deployment and sharing of resources. Workflows let you connect your business-critical apps and services with Azure Logic Apps, automating your workflows without writing a single line of code.'

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Resource Group * [Create new](#)

Instance Details

Logic App name * .azurewebsites.net

Publish * Workflow Docker Container

Region *

Not finding your App Service Plan? Try a different region or select your App Service Environment.

Plan

The plan type you choose dictates how your app scales, what features are enabled, and how it is priced. [Learn more](#)

Plan type * Standard: Best for enterprise-level, serverless applications, with event-based scaling and networking isolation. Consumption: Best for entry-level. Pay only as much as your workflow runs.

Windows Plan (East US) * [Create new](#)

Pricing plan **Workflow Standard WS1** (210 total ACU, 3.5 GB memory, 1 vCPU)

[Review + create](#) [< Previous](#) [Next : Hosting >](#)

g. Select **Review + create**. Then, review your logic app settings and select **Create**.

h. Wait for the deployment to be complete.

Select the blank template

1. After Azure successfully deploys your logic app resource, select **Go to resource** or find and select your logic app resource by typing the name in the Azure search box.

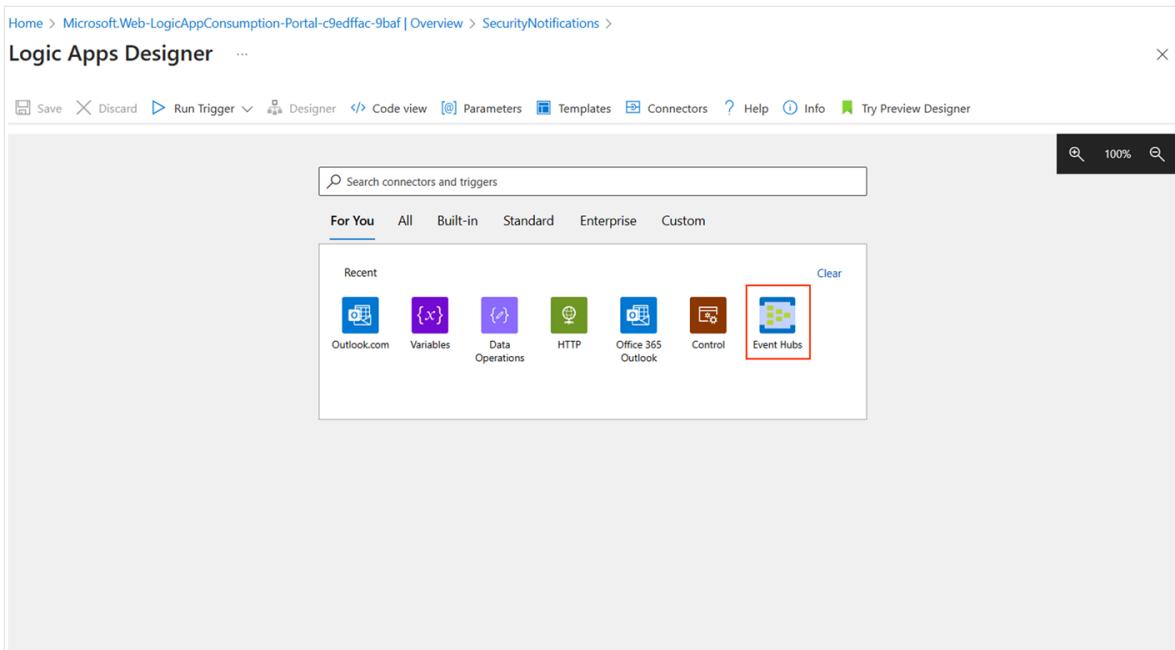
The screenshot shows the Microsoft Azure Logic Apps portal. At the top, it displays the deployment name: Microsoft.Web-LogicApp-Portal-f367c216-896d | Overview. Below the header, there's a search bar and several action buttons: Delete, Cancel, Redeploy, Download, and Refresh. On the left, a navigation sidebar includes links for Overview, Inputs, Outputs, and Template. The main content area features a green checkmark icon followed by the text "Your deployment is complete". It provides deployment details: Deployment name: Microsoft.Web-LogicApp-Portal-f367c216-896d, Subscription: Visual Studio Enterprise Subscription, Resource group: DefaultResourceGroup-EUS, Start time: 7/19/2023, 9:56:06 AM, Correlation ID: 0868b6f8-3ad1-464d-acf3-85b07e9de8c9. There are also sections for "Deployment details" and "Next steps" with a "Go to resource" button.

2. Scroll down past the video under **Templates**, select **Blank Logic App**. After you select the template, the designer shows an empty workflow.

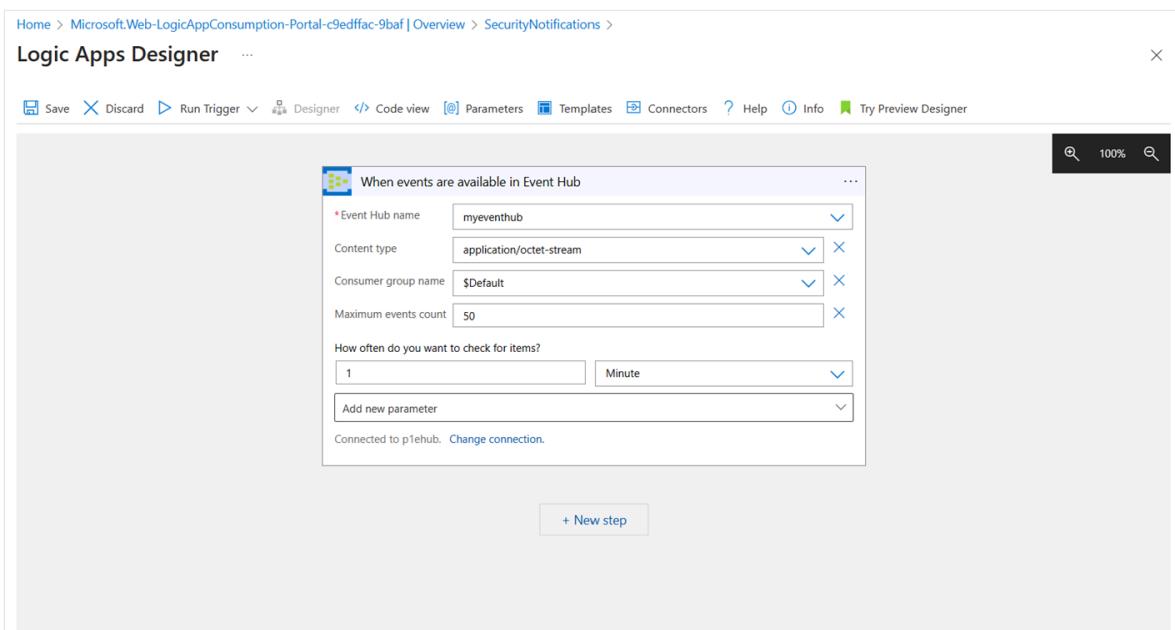
The screenshot shows the Logic Apps Designer interface. At the top, there's a "Watch on YouTube" button. Below it, a section titled "Start with a common trigger" displays various trigger options: "When a message is received in a Service Bus queue", "When a HTTP request is received", "When a new tweet is posted", "When an Event Grid resource event occurs", "Recurrence", "When a new email is received in Outlook.com", "When a new file is created on OneDrive", and "When a file is added to FTP server". Below this, a "Templates" section allows users to choose a template to create their Logic App. It includes filters for "Category: All" and "Sort by: Popularity". The templates listed are: "Blank Logic App" (highlighted with a red border), "Azure Monitor - Metrics Alert Handler", "Auto tier Azure blobs based on the last modified time.", and "Delete old Azure blobs". At the bottom, there are four small preview cards: "HTTP Request-", "Peek-lock receive a", "Correlated in-order", and "Receive an X12 EDI".

Logic Apps Designer

1. In the connectors and triggers section, select **Event Hubs** or search for it in the search bar.



2. Select **When events are available in Event Hubs** trigger. If you're using the Event Hubs trigger for the first time, you'll be prompted to create a connection to your event hub. For more information and steps, see [Create an event hub connection](#).
3. In **Event Hub name**, select the event hub you created in [Prerequisites](#). Select the event hub where you want your logic app to send security notifications.
4. Under **How often do you want to check for items?**, select how often you want the event hub to be checked. In this tutorial, we check for events every one (1) minute.

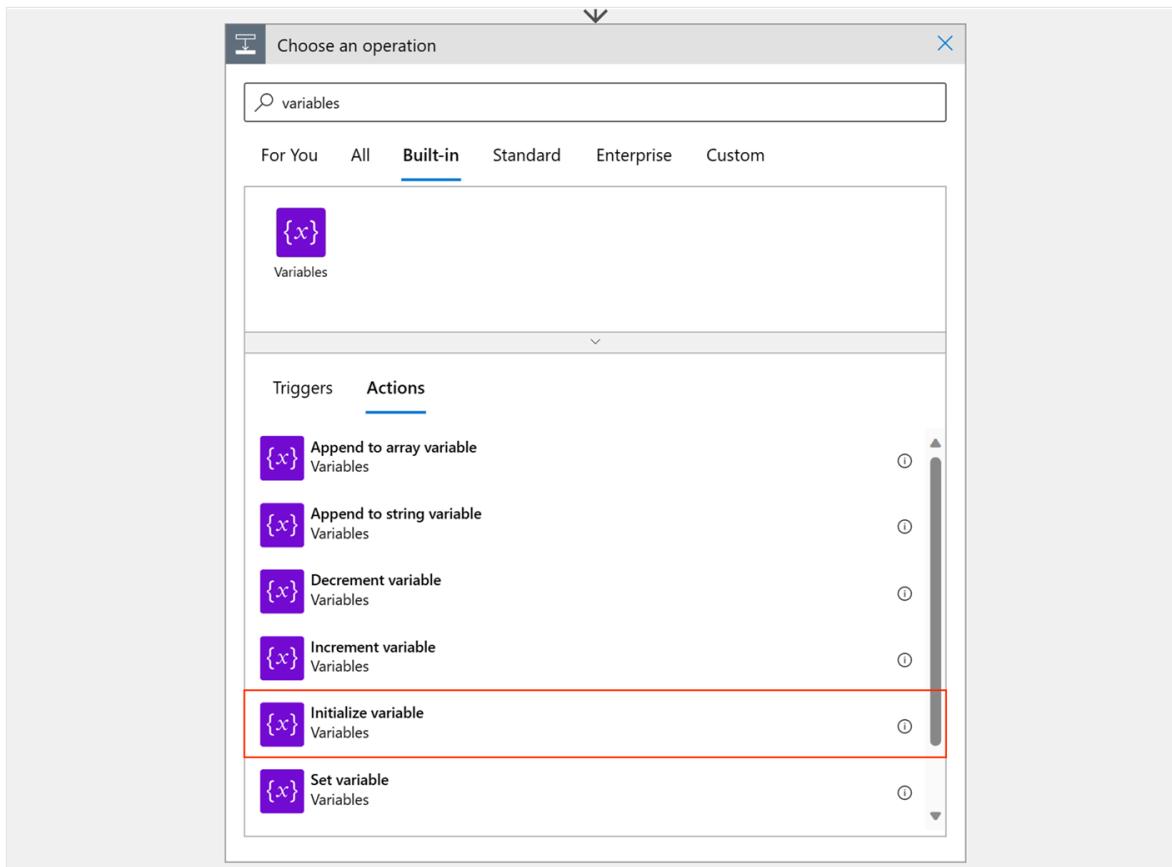


Initialize Variables

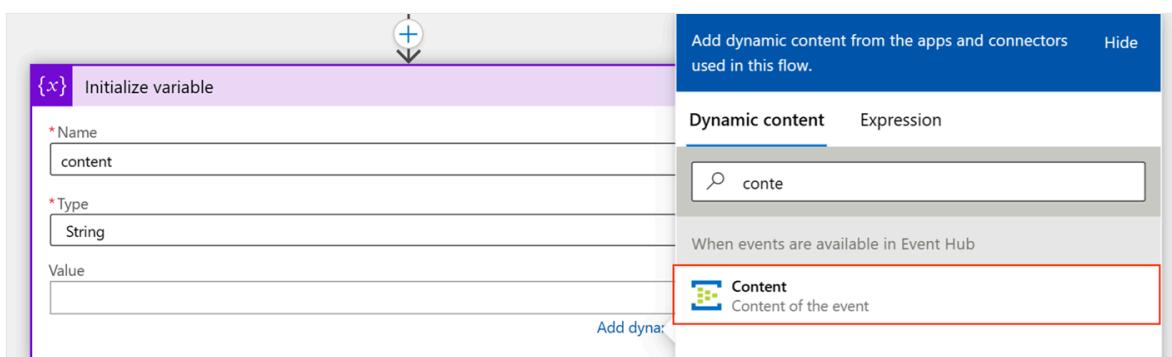
Here, we'll initialize three variables. One is the content of the event that was triggered and streamed to the event hub. The two others are empty variables for our email body,

and the date and time of the activity, which we'll later fill with information from the event.

1. On the designer, under **When events are available in Event Hubs trigger**, select **New step**.
2. Under **Choose an operation**, select **Built-in**. In the search box, enter *variables*, and select **Initialize variable**.



3. For **Name**, type *content*.
4. For **Type**, select **String**.
5. Place the cursor in the **Value** property, and **Dynamic Content** appears.
6. In **Dynamic Content**, search for and select **Content**.

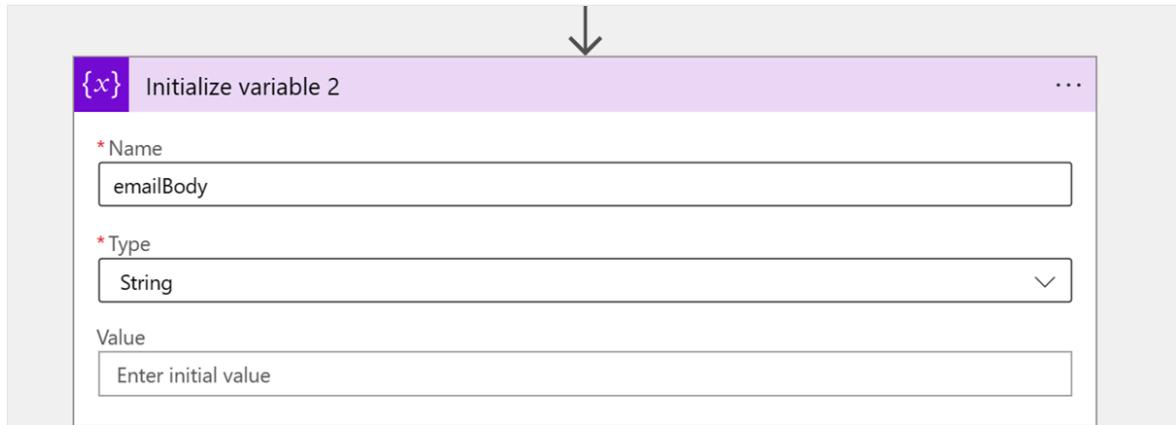


7. Select **New step**.

8. Under **Choose an operation**, select **Built-in**. In the search box, enter *variables*, and select **Initialize variable**.

9. Give the variable a name, such as *emailBody*.

10. For **Type**, select **String**, and leave **Value** blank.

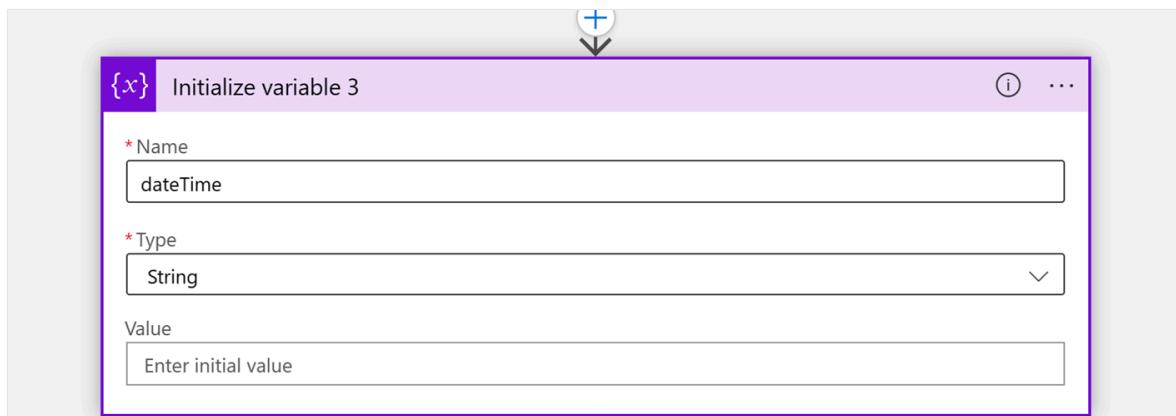


11. Select **New step**.

12. Under **Choose an operation**, select **Built-in**. In the search box, enter *variables*, and select **Initialize variable**.

13. Give the variable a name, such as *dateTime*.

14. For **Type**, select **String**, and leave **Value** blank.



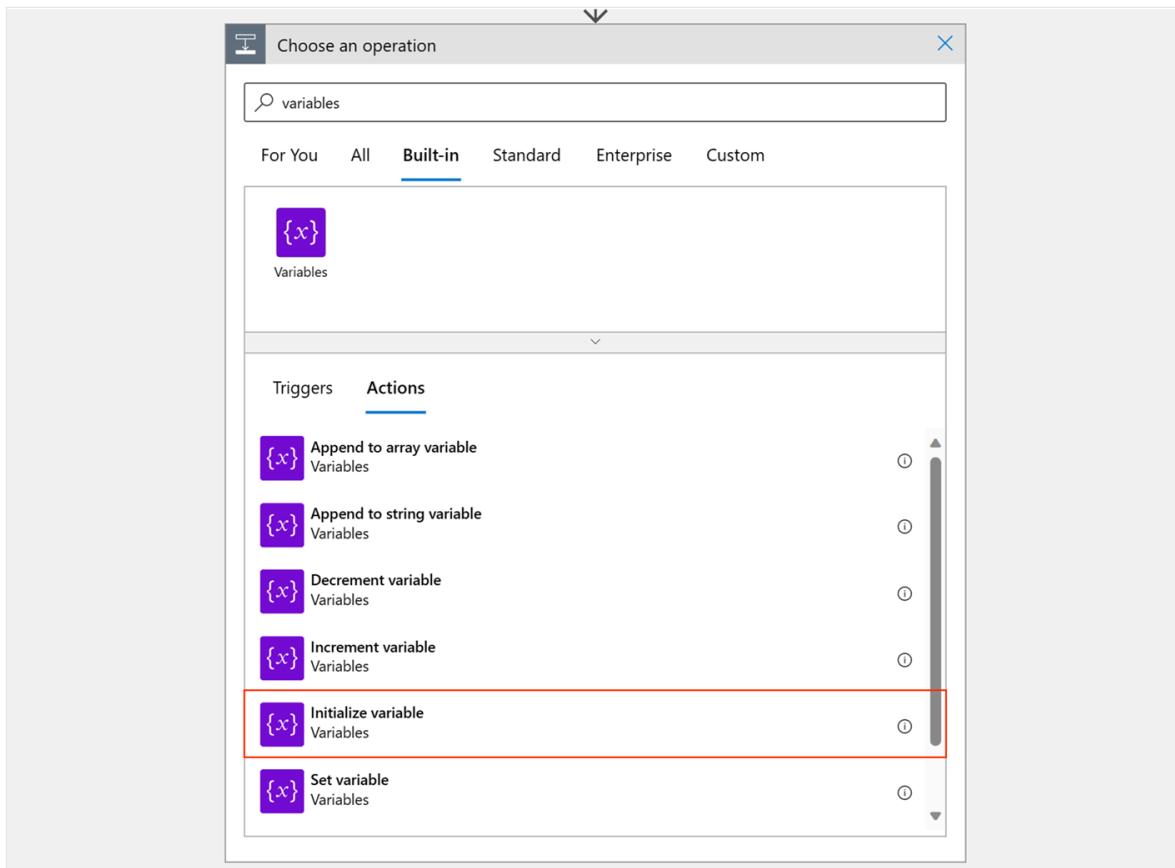
Parse JSON

Now we'll format the raw JSON that we received from the events that were streamed to the event hub by parsing the JSON so we can access specific data within that content.

1. Under **Initialize variable 3**, select **New step**.

2. In the Search connectors and actions search bar, type *Parse JSON*.

3. Switch to the Actions tab and select Parse JSON.



4. In Content, select Add dynamic content.

5. In Dynamic Content, select content under Variables.

6. In the Schema section, copy and paste the following JSON template:

```
JSON

{
  "type": "object",
  "properties": {
    "records": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "time": {
            "type": "string"
          },
          "resourceId": {
            "type": "string"
          },
          "operationName": {
            "type": "string"
          },
          "operationVersion": {
            "type": "string"
          }
        }
      }
    }
}
```

```
"category": {
    "type": "string"
},
"tenantId": {
    "type": "string"
},
"resultSignature": {
    "type": "string"
},
"durationMs": {
    "type": "integer"
},
"correlationId": {
    "type": "string"
},
"Level": {
    "type": "integer"
},
"properties": {
    "type": "object",
    "properties": {
        "id": {
            "type": "string"
        },
        "category": {
            "type": "string"
        },
        "correlationId": {
            "type": "string"
        },
        "result": {
            "type": "string"
        },
        "resultReason": {
            "type": "string"
        },
        "activityDisplayName": {
            "type": "string"
        },
        "activityDateTime": {
            "type": "string"
        },
        "loggedByService": {
            "type": "string"
        },
        "operationType": {
            "type": "string"
        },
        "userAgent": {},
        "initiatedBy": {
            "type": "object",
            "properties": {
                "user": {
                    "type": "object",
                    "properties": {
                        "properties": {
                            "type": "string"
                        }
                    }
                }
            }
        }
    }
}
```

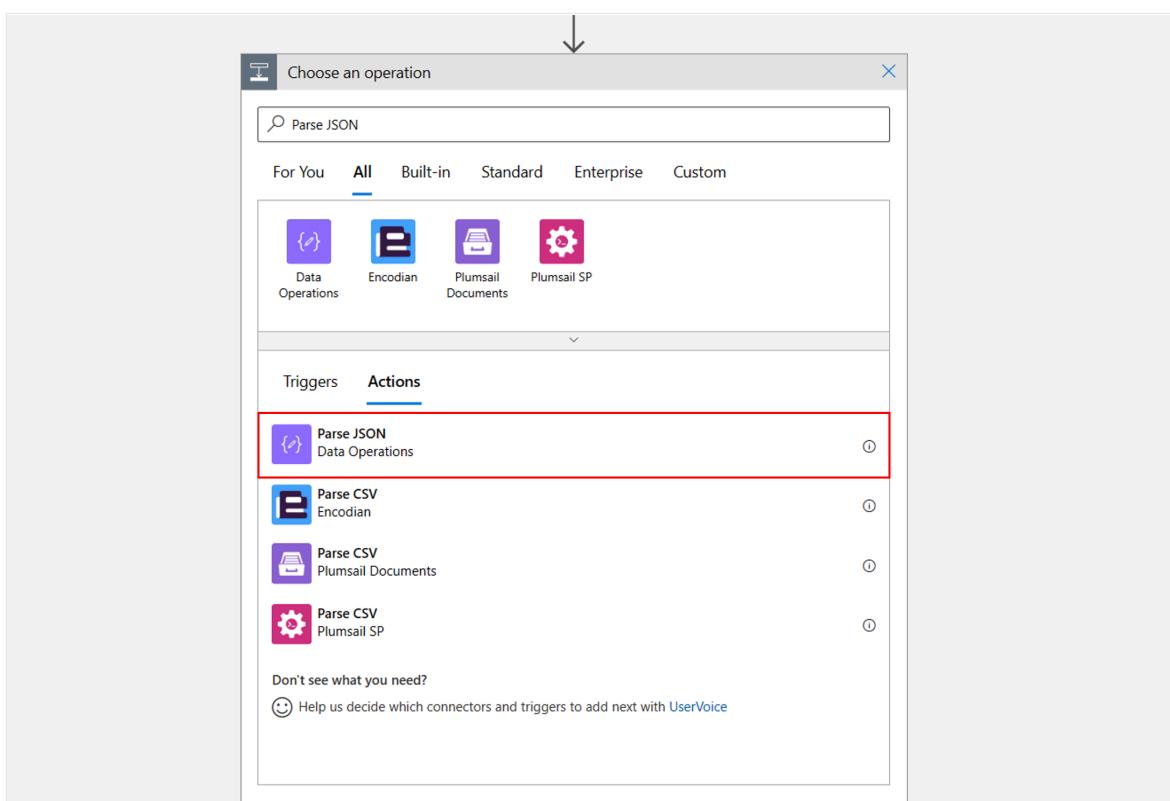
```
        "id": {
            "type": "string"
        },
        "displayName": {},
        "userPrincipalName": {
            "type": "string"
        },
        "ipAddress": {
            "type": "string"
        },
        "roles": {
            "type": "array"
        }
    }
}
},
"targetResources": {
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "id": {
                "type": "string"
            },
            "displayName": {},
            "type": {
                "type": "string"
            },
            "userPrincipalName": {
                "type": "string"
            },
            "modifiedProperties": {
                "type": "array"
            },
            "administrativeUnits": {
                "type": "array"
            }
        },
        "required": [
            "id",
            "displayName",
            "type",
            "userPrincipalName",
            "modifiedProperties",
            "administrativeUnits"
        ]
    }
},
"additionalDetails": {
    "type": "array"
}
}
},
}];
```

```

    "required": [
        "time",
        "resourceId",
        "operationName",
        "operationVersion",
        "category",
        "tenantId",
        "resultSignature",
        "durationMs",
        "correlationId",
        "Level",
        "properties"
    ]
}
}
}
}

```

7. The Parse JSON action should now look like this screenshot:

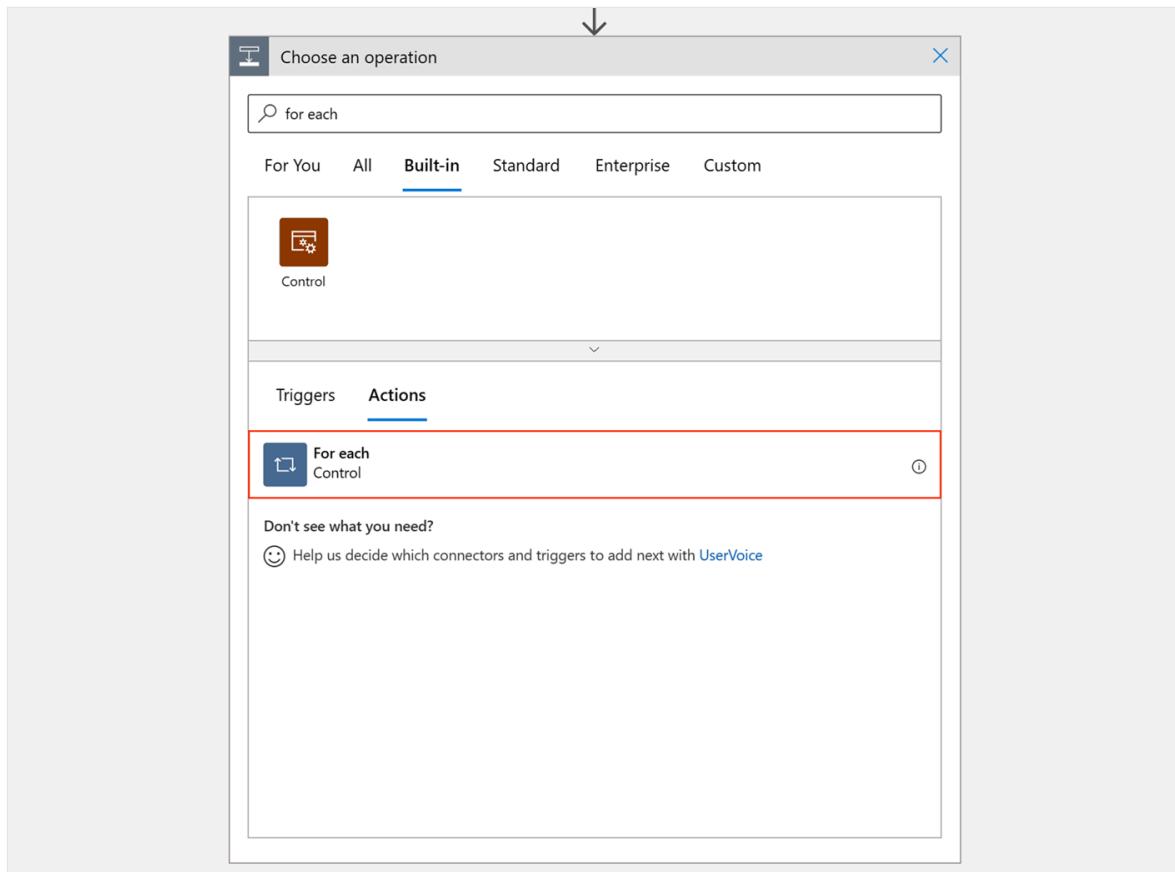


Security notification email body

Next, we'll compose and style the security email that alerts users about the actions taken on their account. Here, we want to inform users of the activity that took place, and prompt them to report it if it wasn't their action.

- Under Parse JSON, select New step.

2. Under **Choose an operation**, select **Built-in**. In the search box, enter *for each*, and from the list of **Actions**, select **For each**.



3. Under **Select an output from previous steps**, select **Add dynamic content**.

4. In **Dynamic content**, select **records**.

Add dynamic content from the apps and connectors used in this flow. [Hide](#)

Dynamic content Expression



Search dynamic content

Parse JSON

[See more](#)



roles



targetResources



additionalDetails

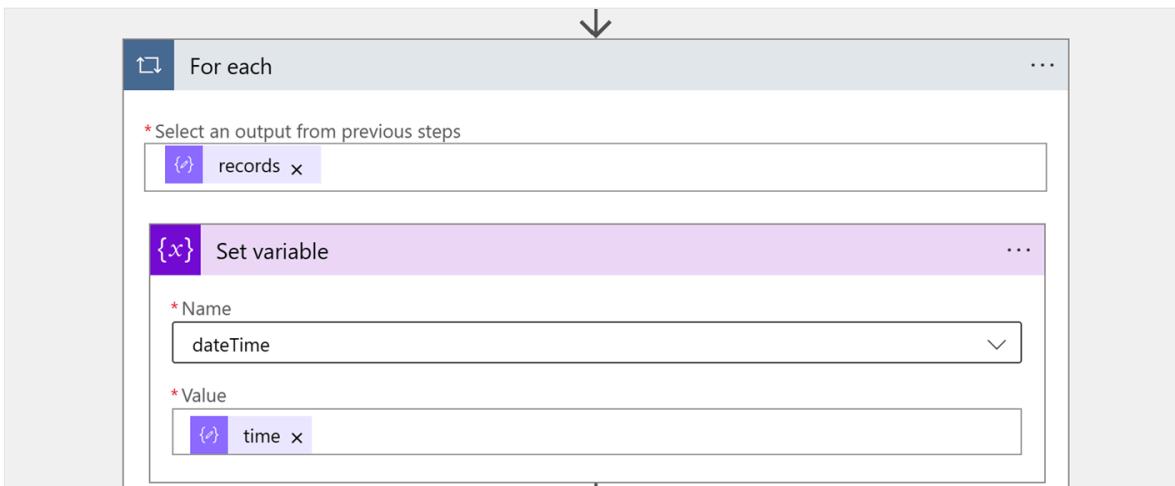


records

5. Inside the **For each** action, select **Add an action**.

The screenshot shows the 'For each' action configuration in Power Automate. The 'records' variable is selected in the dropdown for 'Select an output from previous steps'. A red box highlights the 'Add an action' button at the bottom right.

6. Under **Choose an operation**, select **Built-in**. In the search box, enter **variables**, and select **Set variable**.
7. Under **Name**, select the *dateTime* variable you created.
8. Inside **Value**, select **Add dynamic content**.
9. In **Dynamic content**, search for and select **time** under Parse JSON.



10. Under **Set variable**, select **Built-in**. In the search box, enter **variables**, and select **Set variable**.
11. Under **Name**, select the *emailBody* variable you created.
12. Under **Value**, input the text you want to display in the body of the security notification email. The body can be formatted with html. You can start with this template and customize it. For example, replace the href placeholders with links that are relevant to your organization.

HTML

```
<div>
<h2>
    You recently changed your authentication methods
</h2>
<p>
    We have been notified of the following action: (operation) on
    (date & time). <br><br>
    If you initiated this, no action is required. <br><br>
    If you haven't, please report it now. <br><br>
    <b>Instructions</b>
    <ol>
        <li>Review your account activity in <a href="https://mysignins.microsoft.com/security-info" class="link">Microsoft Security Info</a>.</li>
        <li>If you do not recognize this action, report it immediately:</li>
    <ul>
        <li>Go to <a href="#" class="link">ReportItNow</a> and
            select your security event.</li>
        <li>Provide any additional information in the form and
            submit.</li>
    </ul>
    </ol>
    <b>Information and Support</b>
    <ul>
        <li>Technical Assistance - Contact <a href="#">
```

```

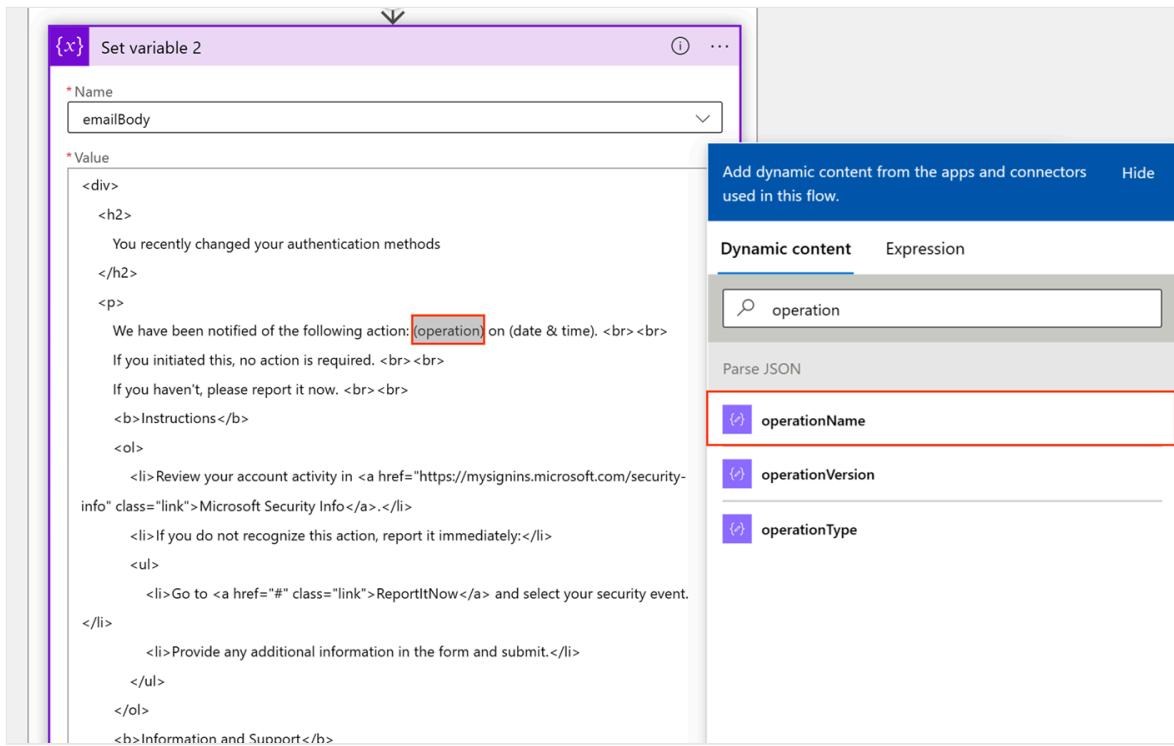
        <class="link">Helpdesk</a> support services</li>
    </ul>
    <b>Do NOT reply to this email. This is an unmonitored mailbox.</b>
<br>
    For more information, contact the <a href="#">
<class="link">Security Department</a>
<br><br>
    <a href="#"><button type="button">Report device</button></a><br>
<br>
    <div class="footer">
        Contoso, Ltd., 4567 Main St Buffalo, NY 98052<br>
        <br>Facilitated by <br>
        
    </div>
    <style>
        .link {
            text-decoration:none;
            color: #0078D4
        }
        button {
            background-color: #0078D4;
            color: white;
            padding: 10px;
            border-radius: 5px;
            text-decoration: none;
        }
        button:hover {
            cursor: pointer;
        }
        .footer {
            width: 100%;
            height: 10%;
            padding-top: 10px;
            padding-left: 10px;
            padding-right: 10px;
            background-color: rgb(237, 237, 237);
        }
    </style>
</p>
</div>

```

Adding dynamic content to the email body

1. If you're using the above template, copy and paste it into the Value field of the Set Variable action.
2. Inside the value field where you pasted the template, go back to the first few lines of text and highlight "(content)". See the image below.

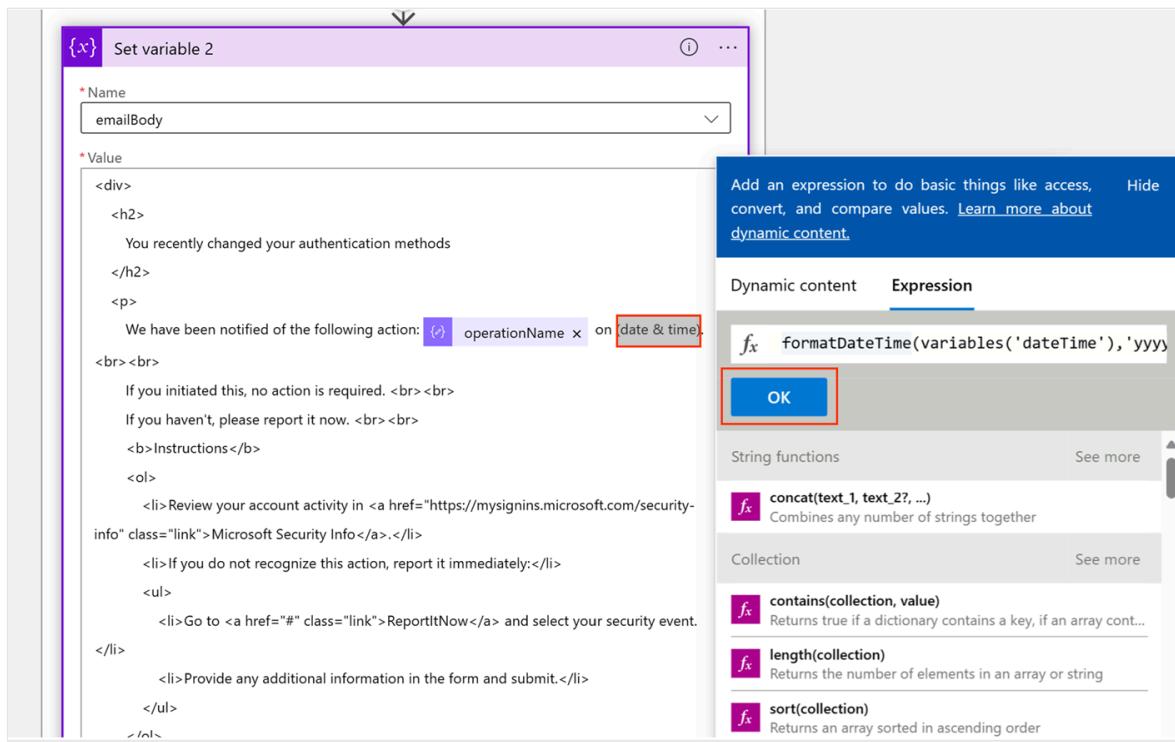
- Once that text has been highlighted, you'll see **Dynamic content** pop up on the right of the action box. In the search bar of Dynamic content, search and select `operationName`.



- Again, inside the value field where you pasted the template, go back to the first few lines of text and highlight "(date & time)". See the image below.
- Once that text has been highlighted, you should see the Dynamic content section pop up on the right of the action box. Go to the Expression tab and input the following code in the input box:

```
code
formatDateTime(variables('dateTime'), 'yyyy-MM-ddTHH:mm:ss')
```

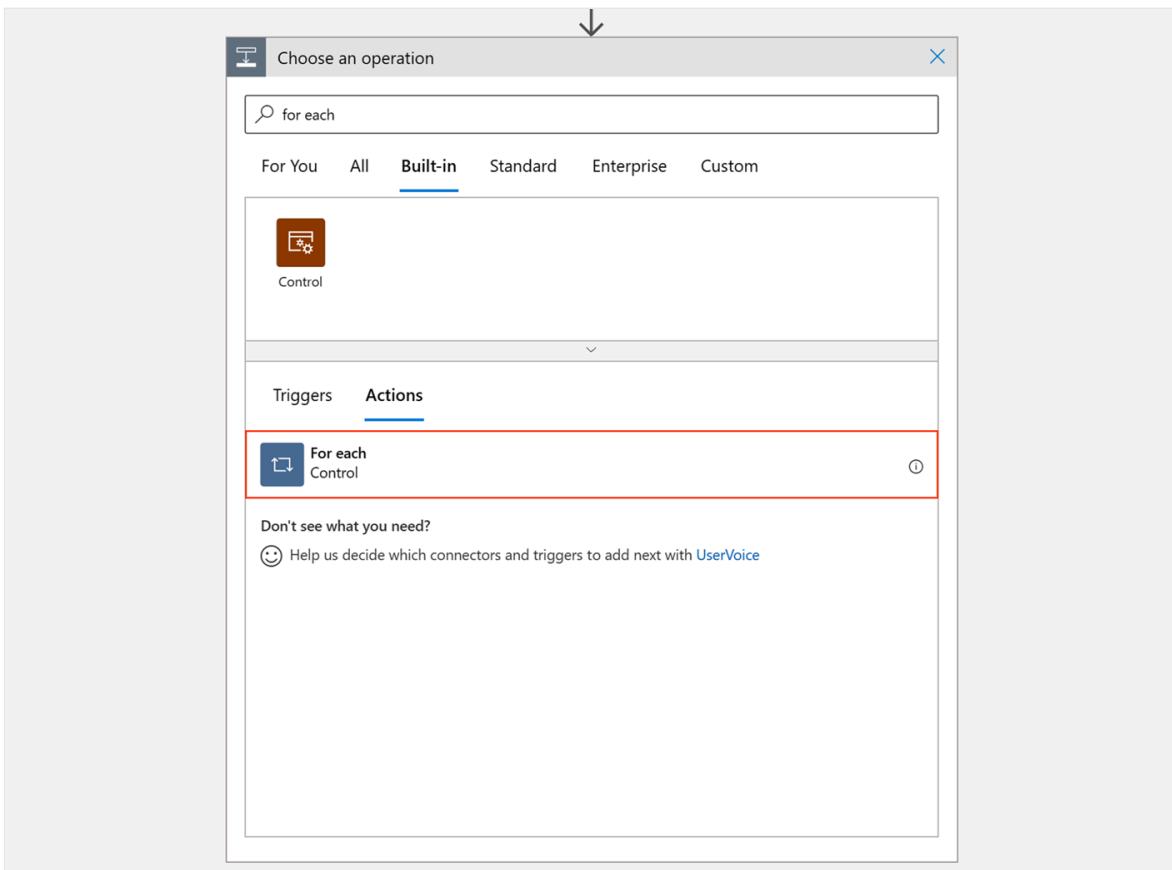
- After pasting the preceding code in the input box, select **OK**.



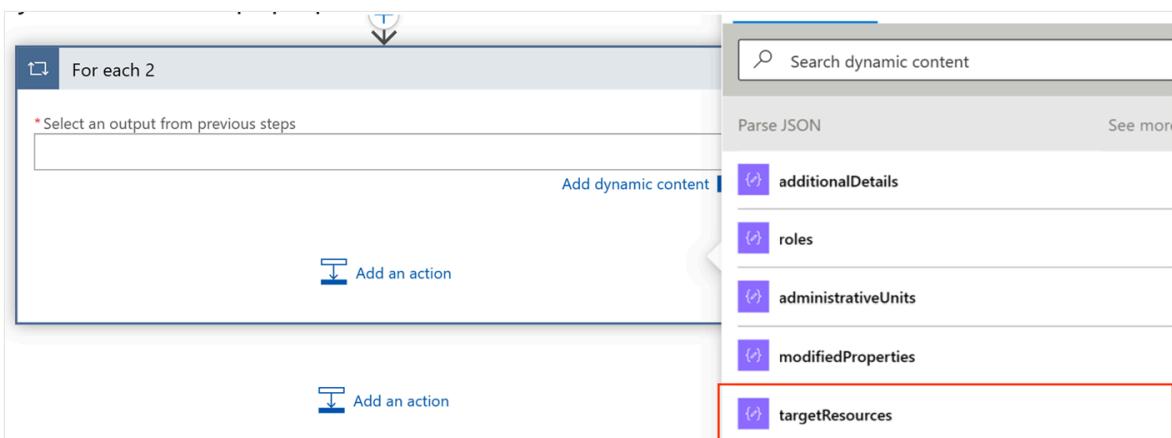
For more information about using dynamic content to customize the email further, see [Workflow dynamic content](#).

Sending the security email

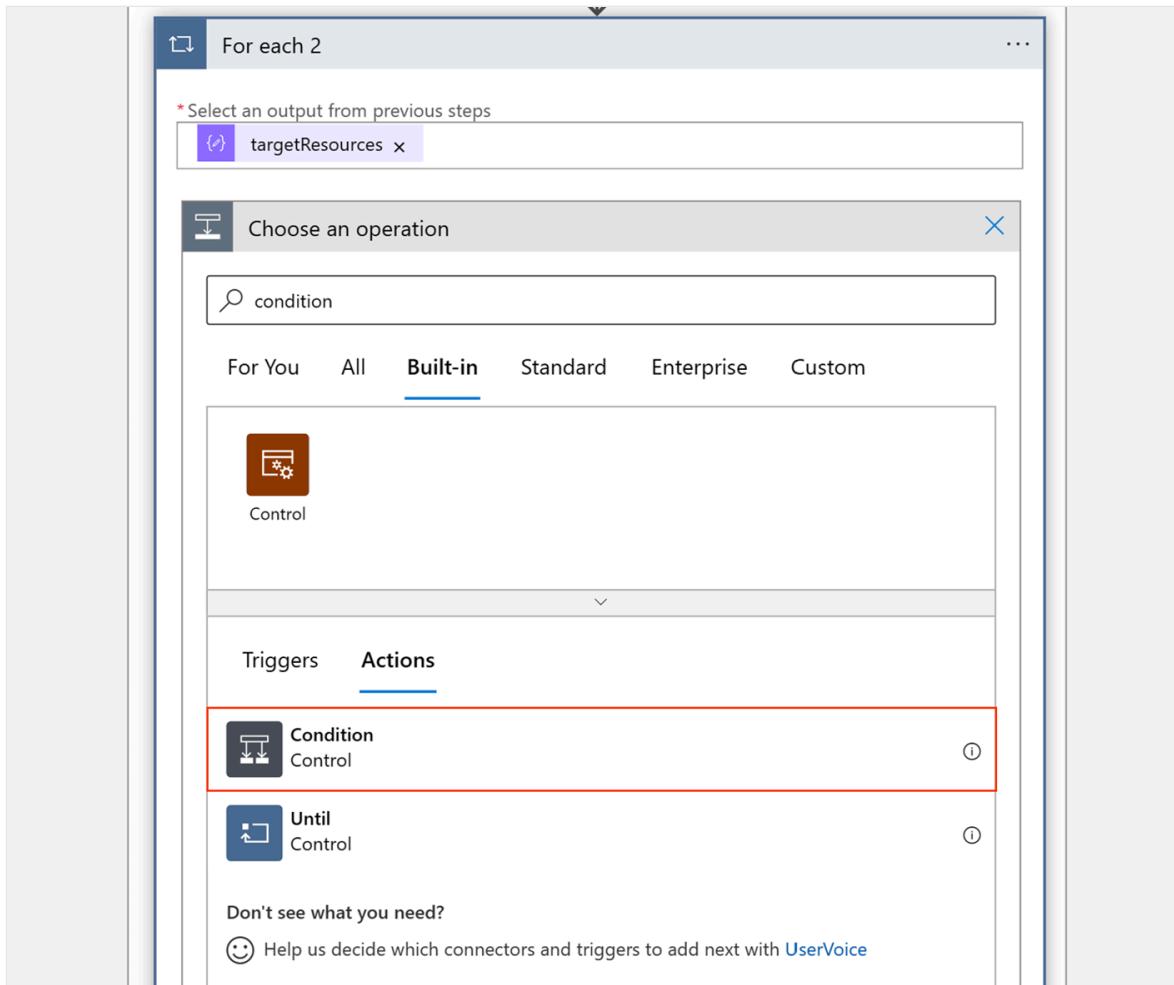
1. Below the **Set variable** action, select **Add an action**.
2. Under **Choose an operation**, select **Built-in**. In the search box, enter **for each** and from the actions list, select the action named **For each**.



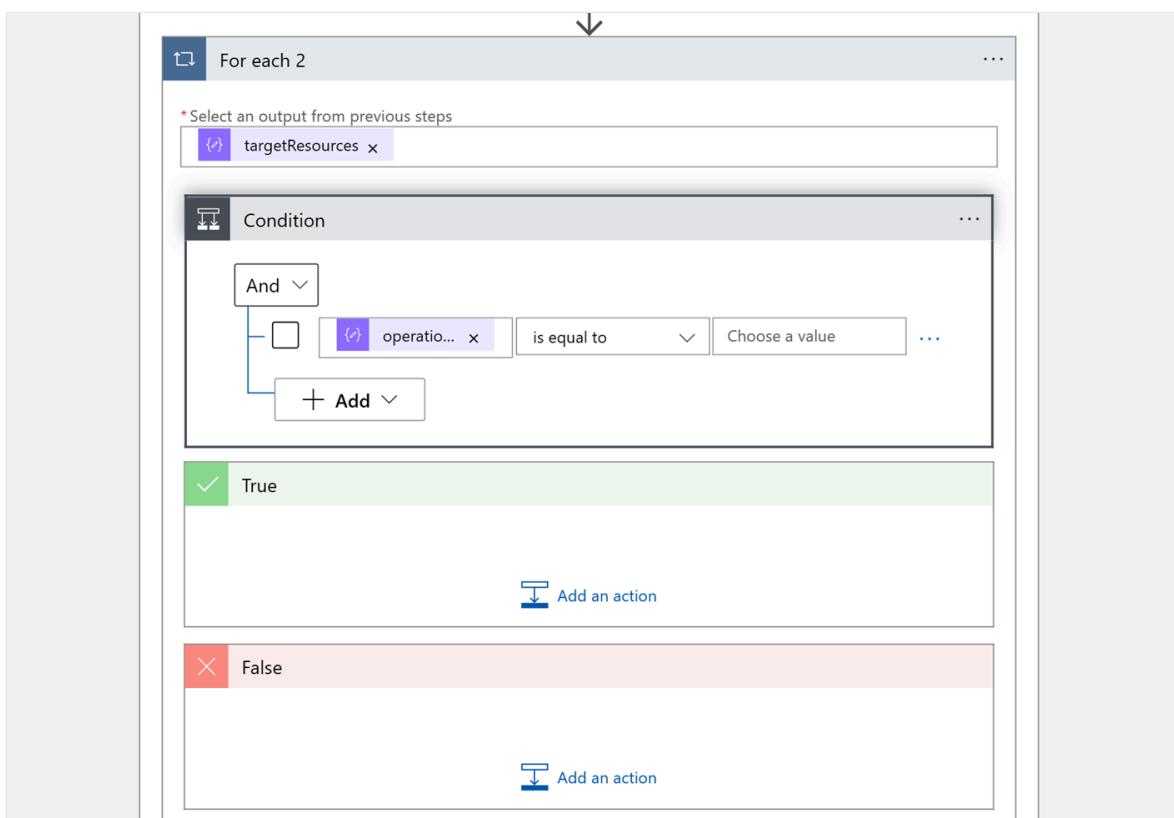
3. Inside **Select an output from previous steps**, select **targetResources** from the **Dynamic content**.



4. Inside the **For each 2** action block and under **targetResources**, select **Add an action**.
5. Under **Choose an operation**, select **Built-in**. In the search box, enter *condition* and from the actions list, select the action named **Condition**.



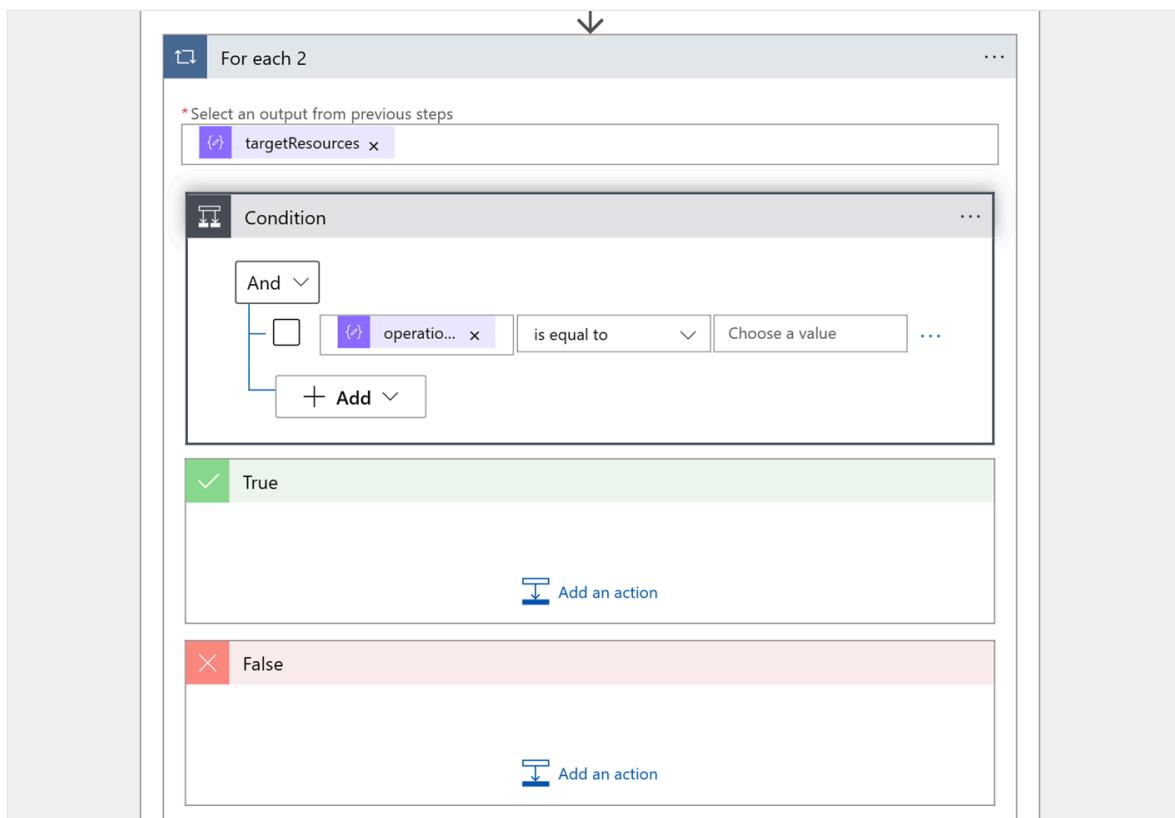
6. Inside Choose a value, search for and select **operationName**.



7. In **Choose a value**, type the exact name of the activity you want to send the security notification emails. For the full list of activities you can filter through and

send notifications for, see [Audit Log Activities](#).

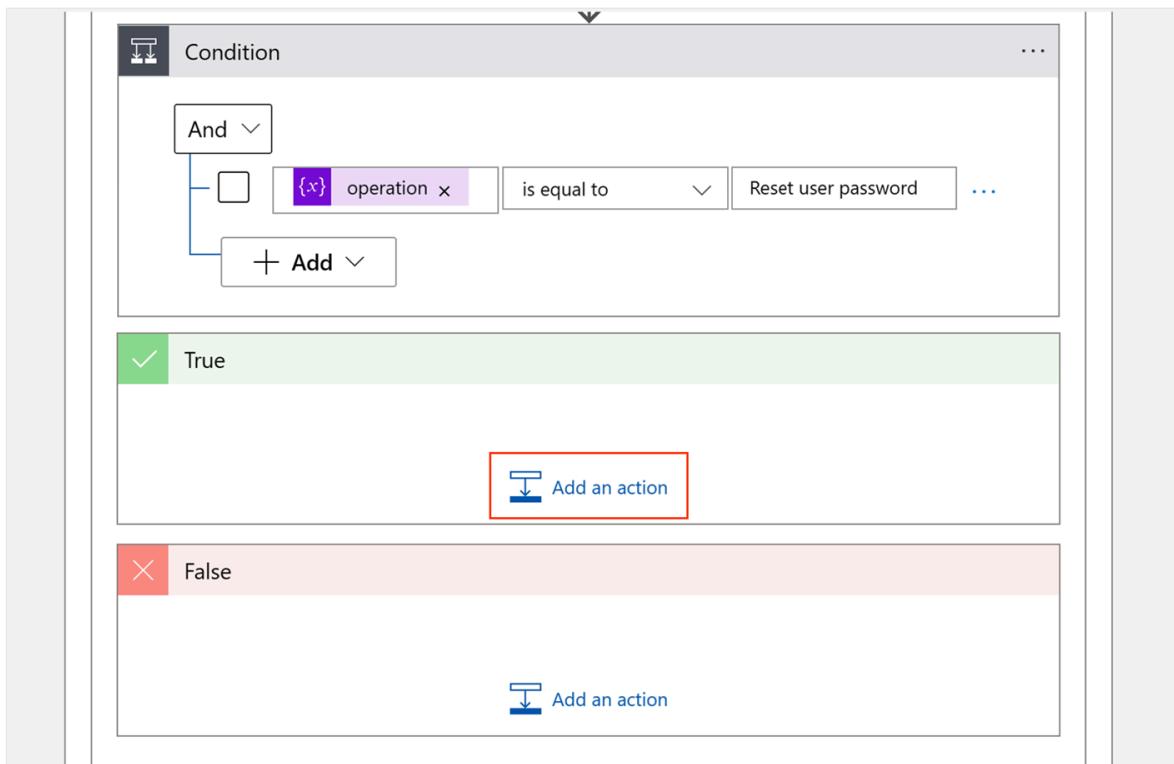
8. For this tutorial, we'll send email for the **Reset user password** activity.



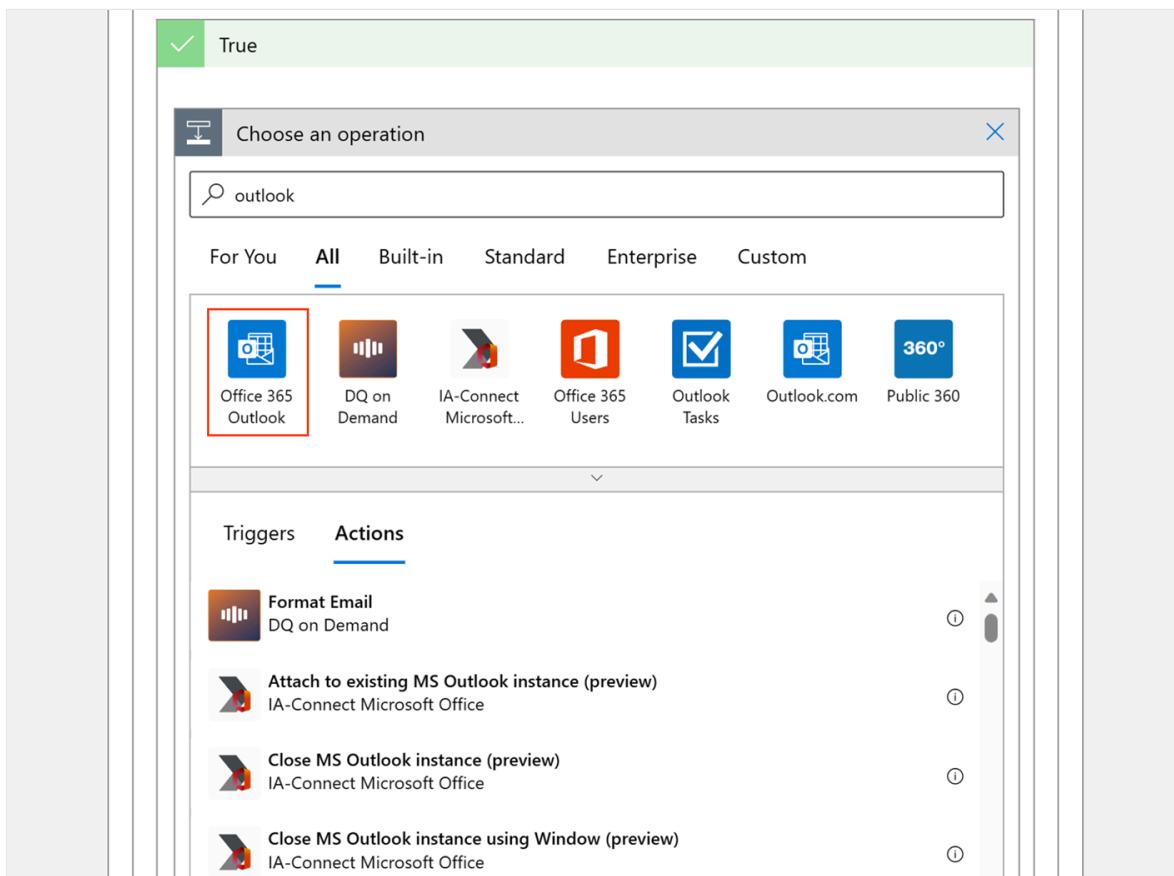
9. If you want to send security emails for multiple activities, select **Add** inside the **Condition** action block, then select **Add row**, and repeat those steps for different activity names in **Choose a value**.

Email notification setup

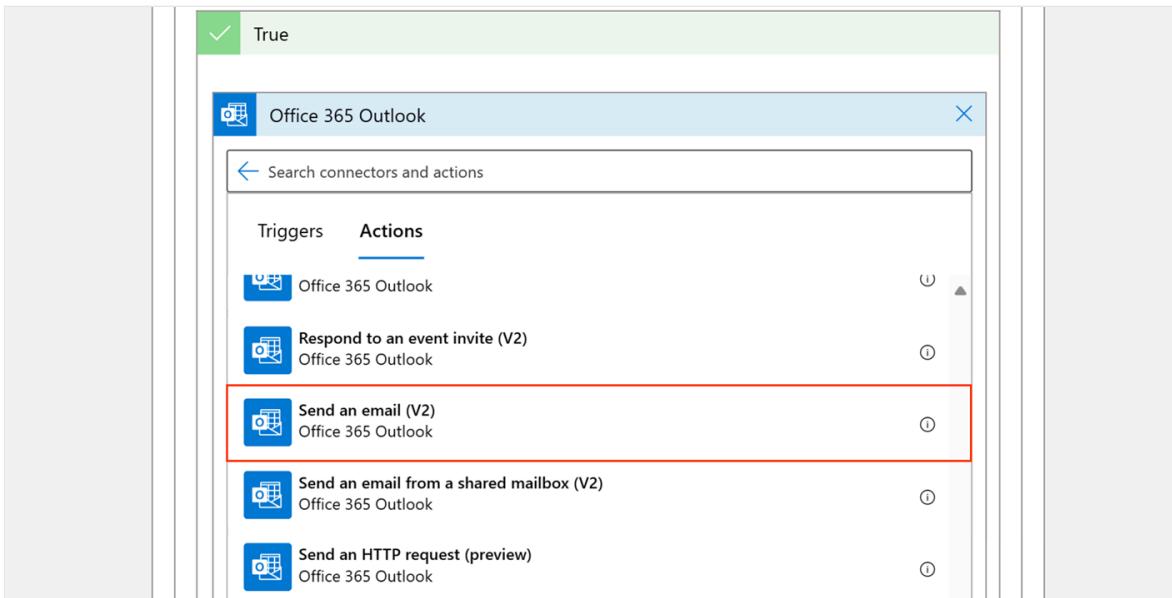
1. Under **Condition**, there are actions for **True** and **False**. Select **Add an action** inside the **True** action box.



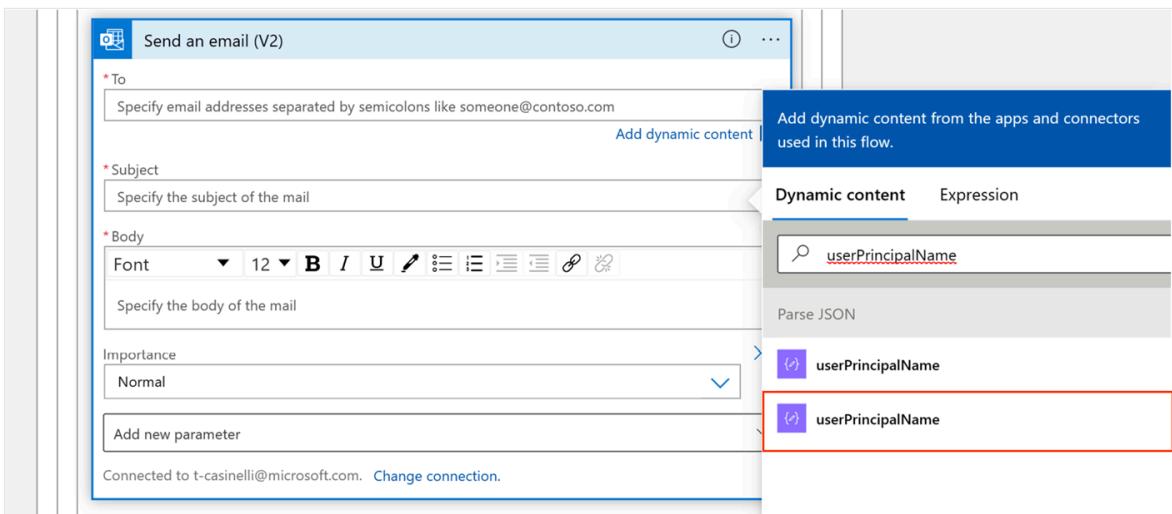
- Under **Choose an operation**, select **Built-in**. In the search box, enter **email**, and select **Office 365 Outlook**. Instead of Outlook emails, you can send notifications with different services. To find different services, go to the search bar in **Choose an operation** and search for the service you prefer.



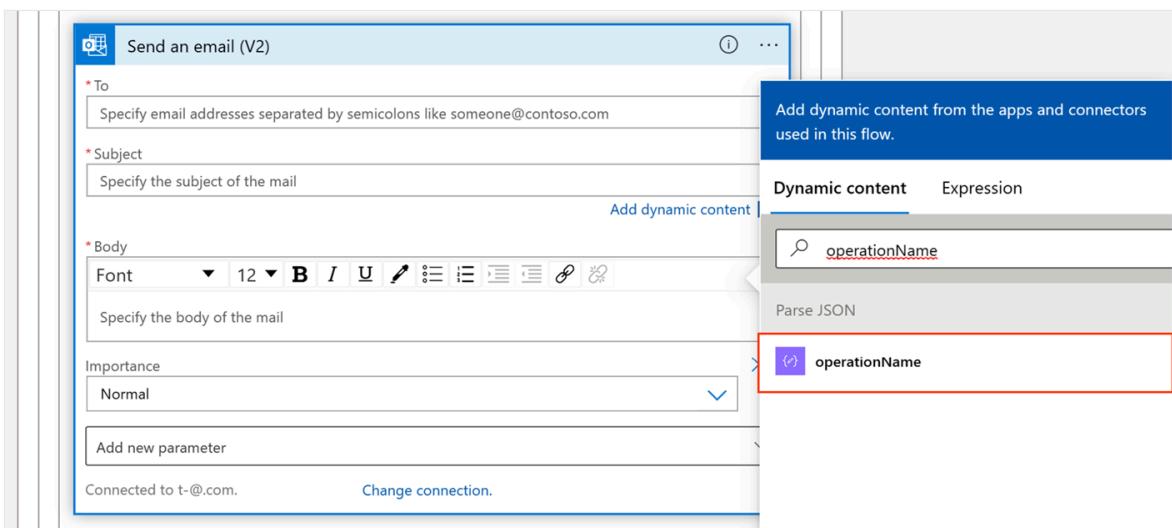
- Under **Actions**, scroll down and select **Send an email (V2)**.



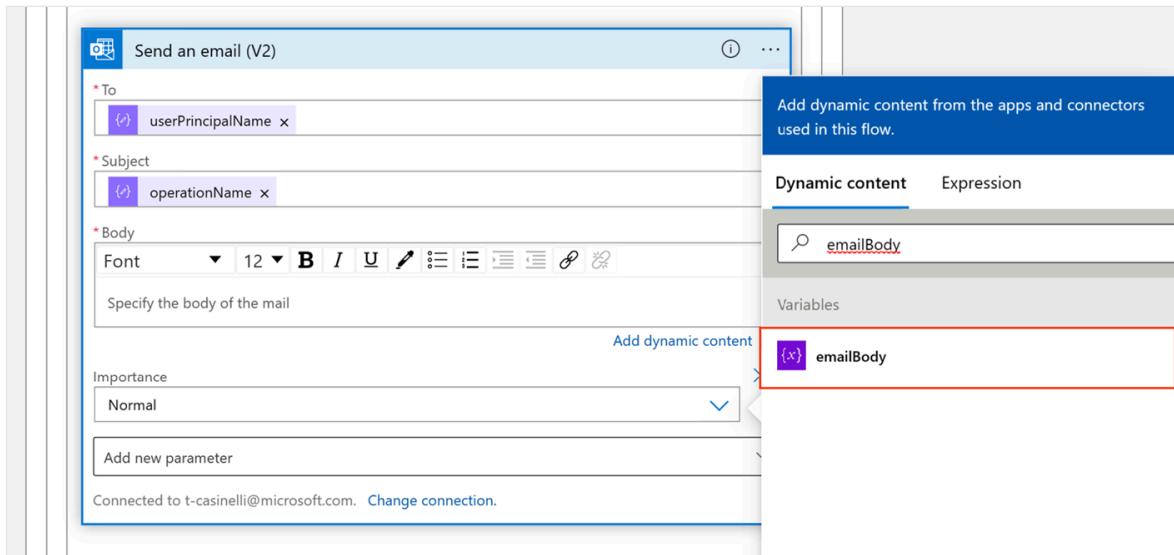
4. Inside the **To** field, search in **Dynamic content** for **userPrincipalName** and select the second option.



5. In the **Subject** field, search in **Dynamic content** for **operationName** and select it.



6. In the **Body** field, search in **Dynamic content** for **emailBody** and select it.



7. You can select **Importance** to change the importance of the email.

Run your workflow

To manually start your workflow, on the Designer toolbar, select **Run Trigger > Run**. When the audit logs stream to the event hub, they trigger the logic app to send the security notification.

This workflow can be customized to filter other logs and activities, or send notifications through different services such as Teams, to create the best experience to make your users aware of suspicious activities.

Next steps

- [How Microsoft Entra ID multifactor authentication works](#)

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Improve accessibility with multifactor authentication in Microsoft Entra ID

Article • 03/04/2025

As cybersecurity threats evolve, multifactor authentication (MFA) has become a cornerstone of secure digital identity. Microsoft Entra ID offers a range of MFA methods designed for robust security and diverse user needs, including those with accessibility constraints. Here's a closer look at how these MFA options enhance accessibility and inclusivity.

Microsoft Authenticator

The Microsoft Authenticator app provides either notifications for quick approval or generates time-based codes for more traditional MFA entry. This app is compatible with various assistive technologies, including screen readers, making it accessible for users with visual impairments. It also offers flexibility for individuals who prefer not to rely solely on SMS or voice calls.

[Download Microsoft Authenticator](#).

Text and voice calls

Text and voice call options cater to those who may not use a smartphone app. This can be beneficial for individuals with certain accessibility needs:

- **Text:** Allows users to receive a verification code via text message, which can be useful for those with hearing impairments or those who prefer text-based communication.
- **Voice calls:** Voice calls are a great option for users with visual impairments, as they provide audio cues rather than visual or tactile ones.

For more information, see [Phone authentication methods](#).

FIDO2 security keys

FIDO2 security keys are physical devices that offer a highly accessible and secure MFA option. These hardware keys support biometric authentication (such as fingerprint scans) or PINs, making them ideal for users who may find traditional passwords or other

authentication methods challenging. FIDO2 keys are beneficial for users with physical disabilities who may have difficulty typing complex passwords.

For more information, see [How to register passkey \(FIDO2\)](#).

Windows Hello for Business

Windows Hello for Business leverages biometric authentication (facial recognition or fingerprint) and PINs, offering a quick, secure, and accessible MFA option. This method eliminates the need for password input, which can be challenging for users with physical or cognitive disabilities. Biometric authentication allows for seamless access while maintaining strong security.

For more information, see [Windows Hello for Business](#).

Email verification

While not as secure as other MFA methods, email verification can be useful in certain accessibility scenarios, providing a fallback option. For users who experience difficulty with text, voice, or app-based authentication, email can offer a familiar and easily accessible alternative.

References:

- [Available verification methods](#)
- [How to enable MFA](#)

Conclusion

Microsoft Entra ID's range of MFA options enables individuals with diverse needs to access secure authentication without compromising on usability. To ensure that security measures remain accessible and inclusive for all users, Microsoft Entra ID offers various options like the Authenticator app, SMS and voice calls, FIDO2 keys, Windows Hello, and email verification.

Selecting the right MFA method depends on individual needs and constraints. Microsoft's commitment to flexible and inclusive authentication helps everyone stay secure, regardless of their physical or technological limitations. For those with specific accessibility requirements, it's worth exploring each MFA option to find the one that aligns best with personal preferences and usability needs.

Related content

- Available verification methods
 - How to enable MFA
-

Feedback

Was this page helpful?



Provide product feedback ↗

What authentication and verification methods are available in Microsoft Entra ID?

Article • 03/04/2025

Microsoft recommends passwordless authentication methods such as Windows Hello, Passkeys (FIDO2), and the Microsoft Authenticator app because they provide the most secure sign-in experience. Although a user can sign-in using other common methods such as a username and password, passwords should be replaced with more secure authentication methods.

Bad: Password	Good: Password and...	Better: Password and...	Best: Passwordless
123456 qwerty password iloveyou Password1	 SMS  Voice	 Authenticator (Push Notifications)  Software Tokens OTP  Hardware Tokens OTP (Preview)	 Authenticator (Phone Sign-in)  Window Hello  FIDO2 security key  Certificates

Microsoft Entra multifactor authentication adds another layer of security over only using a password when a user signs in. The user can be prompted for other forms of authentication, such as to respond to a push notification, enter a code from a software or hardware token, or respond to a text message or phone call.

To simplify the user on-boarding experience and register for both MFA and self-service password reset (SSPR), we recommend you [enable combined security information registration](#). For resiliency, we recommend that you require users to register multiple authentication methods. When one method isn't available for a user during sign-in or SSPR, they can choose to authenticate with another method. For more information, see [Create a resilient access control management strategy in Microsoft Entra ID](#).

How each authentication method works

Some authentication methods can be used as the primary factor when you sign in to an application or device, such as using a FIDO2 security key or a password. Other authentication methods are only available as a secondary factor when you use Microsoft Entra multifactor authentication or SSPR.

The following table outlines when an authentication method can be used during a sign-in event:

[+] Expand table

Method	Primary authentication	Secondary authentication
Windows Hello for Business	Yes	MFA ¹
Microsoft Authenticator push	No	MFA and SSPR
Microsoft Authenticator passwordless	Yes	No ²
Microsoft Authenticator passkey	Yes	MFA
Authenticator Lite	No	MFA
Passkey (FIDO2)	Yes	MFA
Certificate-based authentication (CBA)	Yes	MFA
Hardware OATH tokens (preview)	No	MFA and SSPR
Software OATH tokens	No	MFA and SSPR
External authentication methods (preview)	No	MFA
Temporary Access Pass (TAP)	Yes	MFA
Text	Yes	MFA and SSPR
Voice call	No	MFA and SSPR
QR code (preview)	Yes	No
Password	Yes	No

¹Windows Hello for Business can serve as a step-up MFA credential if it's used in FIDO2 authentication. Users need to be registered for passkey (FIDO2).

²Passwordless sign-in can be used for secondary authentication only if [CBA is used for primary authentication](#).

³Alternate phone methods can only be used for MFA.

All of these authentication methods can be configured in the Microsoft Entra admin center, and increasingly using the [Microsoft Graph REST API](#).

To learn more about how each authentication method works, see the following separate conceptual articles:

- [Windows Hello for Business](#)
- [Microsoft Authenticator app](#)
- [Authenticator Lite](#)
- [Passkey \(FIDO2\)](#)
- [Certificate-based authentication](#)
- [Hardware OATH tokens \(preview\)](#)
- [Software OATH tokens](#)
- [External authentication methods \(preview\)](#)
- [Temporary Access Pass \(TAP\)](#)
- [Short Message Service \(SMS\) sign-in and verification](#)
- [Voice call verification](#)
- [QR code \(preview\)](#)
- [Password](#)

 **Note**

In Microsoft Entra ID, a password is often one of the primary authentication methods. You can't disable the password authentication method. If you use a password as the primary authentication factor, increase the security of sign-in events using Microsoft Entra multifactor authentication.

These other verification methods can be used in certain scenarios:

- [App passwords](#) - used for old applications that don't support modern authentication and can be configured for per-user Microsoft Entra multifactor authentication.

- [Security questions](#) - only used for SSPR
- [Email address](#) - only used for SSPR

Usable and nonusable methods

Administrators can view user authentication methods in the Microsoft Entra admin center. Usable methods are listed first, followed by nonusable methods.

Each authentication method can become nonusable for different reasons. For example, a Temporary Access Pass may expire, or FIDO2 security key may fail attestation. The portal gets updated to provide the reason for why the method isn't usable.

Authentication methods that are no longer available due to **Require re-register multifactor authentication** also appear here.

Adele Vance | Authentication methods

[Add authentication method](#) | [Reset password](#) | [Require re-register multifactor authentication](#)

Want to switch back to the old user authentication methods experience? Click here to go back.

Authentication methods are the ways users sign into Microsoft Entra ID and perform self-service password reset (SSPR). The user's "default sign-in method" is the first one shown to the user when they are required to authenticate with a second factor - the user always can choose another registered, enabled authentication method to authenticate with. [Learn more](#)

Default sign-in method (Preview)	No default
Usable authentication methods	
No usable methods.	
Non-usable authentication methods	
No non-usuable methods.	
System preferred multifactor authentication method	
Feature status	System preferred MFA method
Enabled	No system preferred MFA method

Related content

To get started, see the [tutorial for self-service password reset \(SSPR\)](#) and [Microsoft Entra multifactor authentication](#).

To learn more about SSPR concepts, see [How Microsoft Entra self-service password reset works](#).

To learn more about MFA concepts, see [How Microsoft Entra multifactor authentication works](#).

Learn more about configuring authentication methods using the [Microsoft Graph REST API](#).

To review what authentication methods are in use, see [Microsoft Entra multifactor authentication authentication method analysis with PowerShell](#).

Feedback

Was this page helpful?



[Provide product feedback](#)

Manage authentication methods for Microsoft Entra ID

Article • 04/25/2025

Microsoft Entra ID allows the use of a range of authentication methods to support a wide variety of sign-in scenarios. Administrators can specifically configure each method to meet their goals for user experience and security. This topic explains how to manage authentication methods for Microsoft Entra ID, and how configuration options affect user sign-in and password reset scenarios.

Authentication methods policy

The Authentication methods policy is the recommended way to manage authentication methods, including modern methods like passwordless authentication. [Authentication Policy Administrators](#) can edit this policy to enable authentication methods for all users or specific groups.

Methods enabled in the Authentication methods policy can typically be used anywhere in Microsoft Entra ID, for both authentication and password reset scenarios. The exception is that some methods are inherently limited to use in authentication, such as FIDO2 and Windows Hello for Business, and others are limited to use in password reset, such as security questions. For more control over which methods are usable in a given authentication scenario, consider using the [Authentication Strengths](#) feature.

Most methods also have configuration parameters to more precisely control how that method can be used. For example, if you enable [Voice calls](#), you can also specify whether an office phone can be used in addition to a mobile phone.

Or let's say you want to enable passwordless authentication with Microsoft Authenticator. You can set extra parameters like showing the user sign-in location or the name of the app being signed into. These options provide more context for users when they sign-in and help prevent accidental MFA approvals.

To manage the Authentication methods policy, sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#) and browse to **Entra ID > Authentication methods > Policies**.

Home >

Authentication methods | Policies Contoso - Microsoft Entra ID Security

Search Got feedback? X

Manage

Policies Selected

>Password protection
Registration campaign
Authentication strengths
Settings

Monitoring

Activity
User registration details
Registration and reset events
Bulk operation results

Manage migration

On September 30th, 2025, the legacy multifactor authentication and self-service password reset policies will be deprecated and you'll manage all authentication methods here in the authentication methods policy. Use this control to manage your migration from the legacy policies to the new unified policy. [Learn more](#)

[Manage migration](#)

Method	Target	Enabled
FIDO2 security key	All users	Yes
Microsoft Authenticator	All users	Yes
SMS		No
Temporary Access Pass	All users	Yes
Hardware OATH tokens (Preview)		No
Third-party software OATH tokens	All users	Yes
Voice call	All users	Yes
Email OTP		Yes
Certificate-based authentication		No

Only the [converged registration experience](#) is aware of the Authentication methods policy. Users in scope of the Authentication methods policy but not the converged registration experience won't see the correct methods to register.

Legacy MFA and SSPR policies

Two other policies, located in **Multifactor authentication** settings and **Password reset** settings, provide a legacy way to manage some authentication methods for all users in the tenant. You can't control who uses an enabled authentication method, or how the method can be used.

i Important

In March 2023, we announced the deprecation of managing authentication methods in the legacy multifactor authentication and self-service password reset (SSPR) policies. Beginning September 30, 2025, authentication methods can't be managed in these legacy MFA and SSPR policies. We recommend customers use the manual migration control to migrate to the Authentication methods policy by the deprecation date.

To manage the legacy MFA policy, browse to **Entra ID > Authentication methods > Policies > Multifactor authentication > Additional cloud-based multifactor authentication settings**.

App passwords [Learn more](#)

- Allow users to create app passwords to sign in to non-browser apps
- Do not allow users to create app passwords to sign in to non-browser apps

Trusted IPs [Learn more](#)

Skip multifactor authentication for requests from federated users on my intranet



Skip multifactor authentication for requests from following range of IP address subnets:

1.1.1.1/16

Verification options [Learn more](#)

i Authentication methods for MFA and SSPR can now be managed in one converged policy. [Learn more](#)

Methods available to users:

- Call to phone
- Text message to phone
- Notification through mobile app
- Verification code from mobile app or hardware token

To manage authentication methods for self-service password reset (SSPR), browse to **Entra ID** > **Password reset** > **Authentication methods**. The **Mobile phone** option in this policy allows either voice calls or text message to be sent to a mobile phone. The **Office phone** option allows only voice calls.

The screenshot shows the 'Password reset | Authentication methods' page in the Microsoft Entra ID portal. The left sidebar has sections for 'Manage' (Properties, Authentication methods, Registration, Notifications, Customization, On-premises integration, Administrator Policy), 'Activity' (Audit logs, Usage & insights), and 'Troubleshooting + Support' (New support request). The main area shows a message: 'Authentication Methods for SSPR and Signin can now be managed in one converged policy.' Below it, a slider indicates 'Number of methods required to reset' set to 2. A list of available methods for users includes: Mobile app notification, Mobile app code, Email, Mobile phone, Office phone, and Security questions. A note at the bottom states: 'These settings only apply to end users in your organization. Admins are always enabled for self-service password reset and are required to use two authentication methods to reset their password. Click here to learn more about administrator password policies.'

How policies work together

Settings aren't synchronized between the policies, which allows administrators to manage each policy independently. Microsoft Entra ID respects the settings in all of the policies so a user who is enabled for an authentication method in *any* policy can register and use that method. To prevent users from using a method, it must be disabled in all policies.

Let's walk through an example where a user who belongs to the Accounting group wants to register Microsoft Authenticator. The registration process first checks the Authentication methods policy. If the Accounting group is enabled for Microsoft Authenticator, the user can register it.

If not, the registration process checks the legacy MFA policy. In that policy, any user can register Microsoft Authenticator if one of these settings is enabled for MFA:

- **Notification through mobile app**
- **Verification code from mobile app or hardware token**

If the user can't register Microsoft Authenticator based on either of those policies, the registration process checks the legacy SSPR policy. In that policy too, a user can register Microsoft Authenticator if the user is enabled for SSPR and any of these settings are enabled:

- **Mobile app notification**
- **Mobile app code**

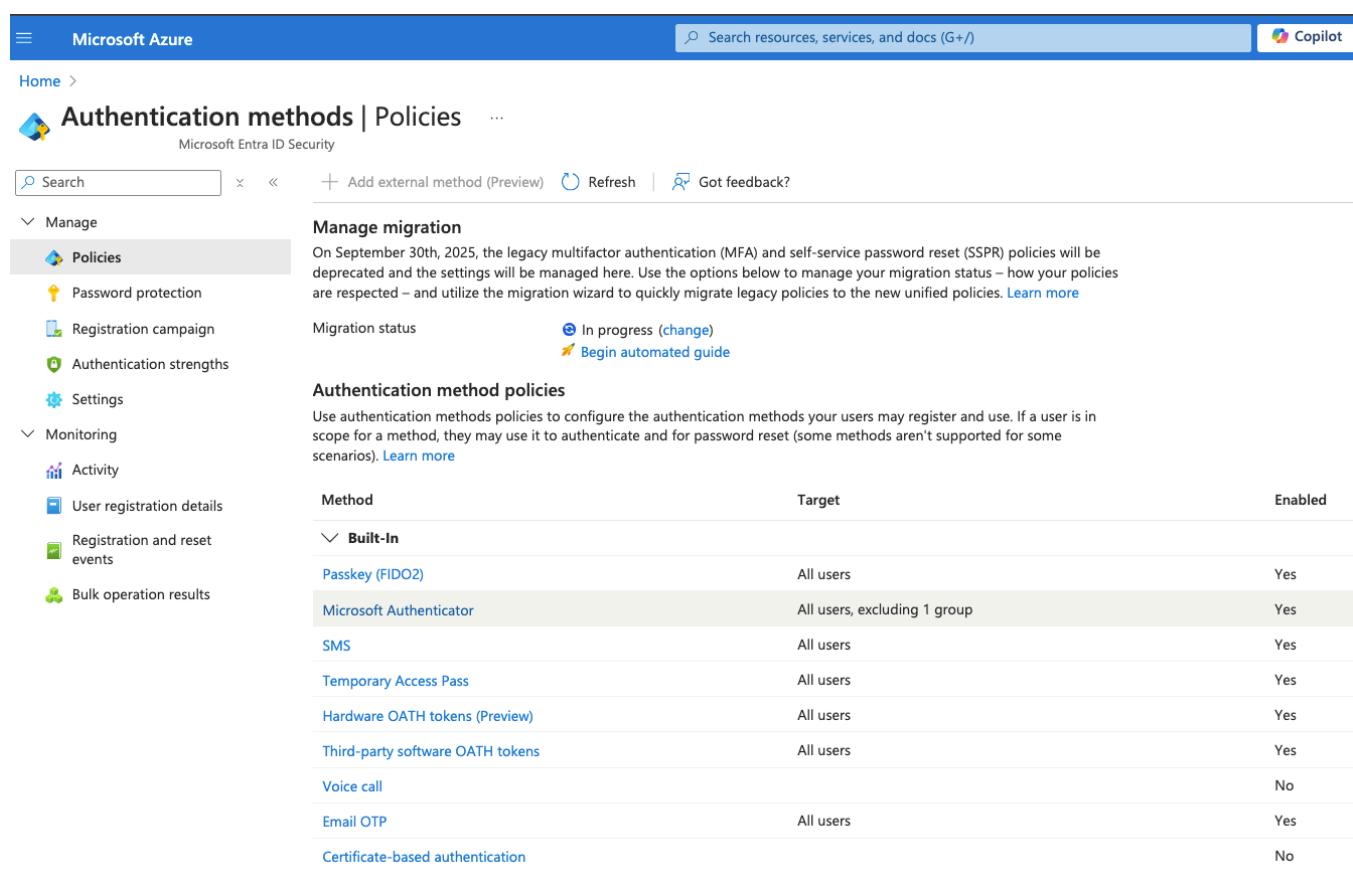
For users who are enabled for **Mobile phone** for SSPR, the independent control between policies can impact sign-in behavior. Where the other policies have separate options for text message and voice calls, the **Mobile phone** for SSPR enables both options. As a result, anyone

who uses **Mobile phone** for SSPR can also use voice calls for password reset, even if the other policies don't allow voice calls.

Similarly, let's suppose you enable **Voice calls** for a group. After you enable it, you find that even users who aren't group members can sign-in with a voice call. In this case, it's likely those users are enabled for **Mobile phone** in the legacy SSPR policy or **Call to phone** in the legacy MFA policy.

Migration between policies

The Authentication methods policy provides a migration guide to help unify administration of all authentication methods. All desired methods can be enabled in the Authentication methods policy if the policy targets intended user groups, or all users. The authentication methods migration guide automates the steps to audit your current policy settings for MFA and SSPR, and consolidate them in the Authentication methods policy. You can access the guide from the [Microsoft Entra admin center](#) by browsing to **Entra ID > Authentication methods > Policies**.



The screenshot shows the Microsoft Entra admin center interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and a Copilot icon. Below the navigation, the breadcrumb path shows 'Home > Authentication methods | Policies'. The main content area has a left sidebar with sections like 'Manage' (selected), 'Policies' (selected), 'Password protection', 'Registration campaign', 'Authentication strengths', 'Settings', 'Monitoring', 'Activity', 'User registration details', 'Registration and reset events', and 'Bulk operation results'. The main pane displays the 'Manage migration' section, which informs users about the deprecation of legacy MFA and SSPR policies and provides a migration wizard. It shows the 'Migration status' as 'In progress' and offers a 'Begin automated guide'. Below this, the 'Authentication method policies' section lists various methods with their targets and enablement status:

Method	Target	Enabled
Passkey (FIDO2)	All users	Yes
Microsoft Authenticator	All users, excluding 1 group	Yes
SMS	All users	Yes
Temporary Access Pass	All users	Yes
Hardware OATH tokens (Preview)	All users	Yes
Third-party software OATH tokens	All users	Yes
Voice call		No
Email OTP	All users	Yes
Certificate-based authentication		No

You can also migrate policy settings manually. The migration has three settings to let you move at your own pace, and avoid problems with sign-in or SSPR during the transition.

After migration is complete, methods in the legacy MFA and SSPR policies can be disabled. You can centralize control over authentication methods for both sign-in and SSPR in a single place, and the legacy MFA and SSPR policies will be disabled.

Note

Security questions can only be enabled today by using the legacy SSPR policy. If you're using security questions, and don't want to disable them, make sure to keep them enabled in the legacy SSPR policy until a migration control is available. You can migrate the remainder of your authentication methods and still manage security questions in the legacy SSPR policy.

To view the migration options, open the Authentication methods policy and click **Manage migration**.

Manage migration

On September 30th, 2025, the legacy multifactor authentication and self-service password reset policies will be deprecated and you'll manage all authentication methods here in the authentication methods policy. Use this control to manage your migration from the legacy policies to the new unified policy.

[Learn more !\[\]\(dca936d53d3420176de550a9113dbc36_img.jpg\)](#)

Pre-migration:

Use policy for authentication only, respect legacy policies.

Migration In Progress:

Use policy for authentication and SSPR, respect legacy policies.

Migration Complete:

Use policy for authentication and SSPR, ignore legacy policies.

The following table describes each option.

 [Expand table](#)

Option	Description
Pre-migration	The Authentication methods policy is used only for authentication. Legacy policy settings are respected.
Migration in Progress	The Authentication methods policy is used for authentication and SSPR. Legacy policy settings are respected.

Option	Description
Migration Complete	Only the Authentication methods policy is used for authentication and SSPR. Legacy policy settings are ignored.

Tenants are set to either Pre-migration or Migration in Progress by default, depending on their tenant's current state. If you start in Pre-migration, you can move to any of the states at any time. If you started in Migration in Progress, you can move between Migration in Progress and Microsoft Complete at any time, but won't be allowed to move to Pre-migration. If you move to Migration Complete, and then choose to roll back to an earlier state, we'll ask why so we can evaluate performance of the product.

Manage migration



On September 30th, 2025, the legacy multifactor authentication and self-service password reset policies will be deprecated and you'll manage all authentication methods here in the authentication methods policy. Use this control to manage your migration from the legacy policies to the new unified policy.

[Learn more](#)

Pre-migration:

Use policy for authentication only, respect legacy policies.

Migration In Progress:

Use policy for authentication and SSPR, respect legacy policies.

Migration Complete:

Use policy for authentication and SSPR, ignore legacy policies.

Tell us why you're re-enabling legacy policies: *

Not comfortable migrating yet



Tell us more...

Testing scenarios

Note

After all authentication methods are fully migrated, the following elements of the legacy SSPR policy remain active:

- The **Number of methods required to reset** control: admins can continue to change how many authentication methods must be verified before a user can perform SSPR.
- The SSPR administrator policy: admins can continue to register and use any methods listed under the legacy SSPR administrator policy or methods they're enabled to use in the Authentication methods policy.

In the future, both of these features will be integrated with the Authentication methods policy.

Known issues and limitations

- In recent updates, we removed the ability to target individual users. Previously targeted users will remain in the policy, but we recommend moving them to a targeted group.
- Registration of an authentication method can fail if many groups are included in the Authentication methods policy or a registration campaign. We recommend consolidating multiple groups into a single group for each authentication method. To maintain registration for users during consolidation, add the new group and remove current groups in the same operation.

Note

You might not be able to save updates to the Authentication methods policy if it targets many groups and the policy size exceeds 20 KB. While we work to increase the policy size limit, consolidate targeted groups as much as possible.

Next steps

- [How to migrate MFA and SSPR policy settings to the Authentication methods policy](#)
- [What authentication and verification methods are available in Microsoft Entra ID?](#)
- [How Microsoft Entra multifactor authentication works](#)
- [Microsoft Graph REST API](#)

Configure Temporary Access Pass to register passwordless authentication methods

Article • 03/04/2025

Passwordless authentication methods like a passkey (FIDO2) let users sign in securely without a password. Users can bootstrap passwordless methods in one of two ways:

- Use existing Microsoft Entra multifactor authentication methods
- Use a Temporary Access Pass

A Temporary Access Pass (TAP) is a time-limited passcode that can be configured for single use or multiple sign-ins. Users can sign in with a TAP to onboard other passwordless authentication methods. A TAP also makes recovery easier when a user loses or forgets a strong authentication method.

This article shows you how to enable and use a TAP using the [Microsoft Entra admin center](#). You can also perform these actions using REST APIs.

Enable the Temporary Access Pass policy

A TAP policy defines settings, such as the lifetime of passes created in the tenant, or the users and groups who can use a TAP to sign-in.

Before users can sign-in with a TAP, you need to enable this method in the Authentication methods policy and choose which users and groups can sign in by using a TAP.

Although you can create a TAP for any user, only users included in the policy can sign-in with it. You need the [Authentication Policy Administrator](#) role to update the TAP Authentication methods policy.

To configure TAP in the Authentication methods policy:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Policies**.
3. From the list of available authentication methods, select **Temporary Access Pass**.

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation sidebar with various options like User settings, Groups, Devices, Applications, Roles & admins, Billing, Settings, Protection, Identity Protection, Conditional Access, Security Center, Identity Secure Score, Multifactor authentication, and Authentication methods. The 'Authentication methods' link is highlighted with a red box. The main content area shows the 'Authentication methods | Policies' page for Contoso - Microsoft Entra ID Security. It includes sections for Manage (with Policies highlighted with a red box), Manage migration (with a note about legacy policies being deprecated), and a table of authentication methods. The 'Temporary Access Pass' method is highlighted with a red box in the table.

4. Select **Enable** and then select users to include or exclude from the policy.

The screenshot shows the 'Temporary Access Pass settings' page. The left sidebar has links for Home, Favorites, Identity (Overview, Users, Groups, Devices, Applications, Protection, Authentication methods, Password reset, Custom security attributes), and Learn & support. The main content area shows the 'Temporary Access Pass settings' page with a description of what TAP is. It has sections for 'Enable and Target' (with the 'Enable' button highlighted with a red box) and 'Include' (with 'All users' selected). A table lists the target users as 'All users' type 'Group'.

5. (Optional) Select **Configure** to modify the default Temporary Access Pass settings, such as setting maximum lifetime, or length, and select **Update**.

The screenshot shows the 'Temporary Access Pass settings' page with more detailed configuration options. The left sidebar is the same as the previous screenshot. The main content area shows the 'Temporary Access Pass settings' page with a description of TAP. It has sections for 'Enable and Target' (with 'Configure' highlighted with a red box), 'GENERAL' (with 'Edit' highlighted with a red box), and configuration options for 'Minimum lifetime', 'Maximum lifetime', 'Default lifetime', 'Length', 'Require one-time use', and 'Length (characters)'. The 'Update' button at the bottom right is highlighted with a red box.

6. Select **Save** to apply the policy.

The default value and the range of allowed values are described in the following table.

 Expand table

Setting	Default values	Allowed values	Comments
Minimum lifetime	1 hour	10 – 43,200 Minutes (30 days)	Minimum number of minutes that the TAP is valid.
Maximum lifetime	8 hours	10 – 43,200 Minutes (30 days)	Maximum number of minutes that the TAP is valid.
Default lifetime	1 hour	10 – 43,200 Minutes (30 days)	Individual passes within the minimum and maximum lifetime configured by the policy can override default value.
One-time use	False	True/False	When the policy is set to false, passes in the tenant can be used either once or more than once during its validity (maximum lifetime). By enforcing one-time use in the TAP policy, all passes created in the tenant are one-time use.
Length	8	8-48 characters	Defines the length of the passcode.

Create a Temporary Access Pass

After you enable a TAP policy, you can create a TAP policy for users in Microsoft Entra ID. The following roles can perform various actions related to a TAP.

- **Privileged Authentication Administrators** can create, delete, and view a TAP for admins and members (except themselves).
- **Authentication Administrators** can create, delete, and view a TAP for members (except themselves).
- **Authentication Policy Administrators** can enable TAP, include or exclude groups, and edit the Authentication methods policy.
- **Global Readers** can view TAP details for the user (without reading the code itself).

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Administrator**.

2. Browse to Identity > Users.

3. Select the user you would like to create a TAP for.

4. Select Authentication methods and select Add authentication method.

The screenshot shows the Microsoft Entra admin center interface. On the left, the navigation pane is visible with sections like Home, Favorites, Identity, Overview, Users, Groups, Devices, Applications, Roles & admins, Billing, Settings, Learn & support. Under 'Users', 'All users' is selected. In the main content area, 'Adele Vance | Authentication methods' is displayed. A red box highlights the '+ Add authentication method' button. To the right, a modal window titled 'Add authentication method' is open. It has a dropdown menu 'Choose method' set to 'Temporary Access Pass'. Below it, instructions say 'Create a Temporary Access Pass for Adele Vance. While the pass is valid, the user can use it to sign in and register strong credentials.' There are checkboxes for 'Delayed start time' (unchecked) and 'Activation duration' (set to 1 hour). A radio button 'One-time use' is set to 'Yes'. At the bottom right of the modal is a blue 'Add' button.

5. Select Temporary Access Pass.

6. Define a custom activation time or duration and select Add.

This screenshot shows the 'Add authentication method' dialog with the following configuration: 'Choose method' is set to 'Temporary Access Pass'. The activation duration is set to 1 hour. The 'One-time use' option is selected. A red box highlights the 'Add' button at the bottom.

7. Once added, the details of the TAP are shown.

Important

Make a note of the actual TAP value, because you provide this value to the user. You can't view this value after you select Ok.

Temporary Access Pass details

X

Provide Pass

Provide this Temporary Access Pass to the user so they can set their strong credentials.

%3&L&CRM



Secure registration

To register their credentials, have the user go to My Security Info.

<https://aka.ms/mysecurityinfo>



Additional information

Valid from 2/27/2024, 8:08:22 PM

Valid until 2/27/2024, 9:08:22 PM

Created 2/27/2024, 8:08:23 PM



Remove lost devices from the user's account. This is especially important for devices used for user authentication.

Ok

8. Select **OK** when you're done.

The following commands show how to create and get a TAP using PowerShell.

PowerShell

```
# Create a Temporary Access Pass for a user
$properties = @{}
$properties.isUsableOnce = $True
$properties.startDateTime = '2022-05-23 06:00:00'
$propertiesJSON = $properties | ConvertTo-Json

New-MgUserAuthenticationTemporaryAccessPassMethod -UserId user2@contoso.com
-BodyParameter $propertiesJSON
```

Id	CreatedDateTime	IsUsable	
IsUsableOnce	LifetimeInMinutes	MethodUsabilityReason	StartTime
TemporaryAccessPass			
--			

00aa00aa-bb11-cc22-dd33-44ee44ee44ee 5/22/2022 11:19:17 PM False True			
60	NotYetValid	23/05/2022 6:00:00 AM	TAP Rocks!
# Get a user's Temporary Access Pass			
Get-MgUserAuthenticationTemporaryAccessPassMethod -UserId user3@contoso.com			
Id	CreatedDateTime	IsUsable	
IsUsableOnce	LifetimeInMinutes	MethodUsabilityReason	StartTime
TemporaryAccessPass			
--			

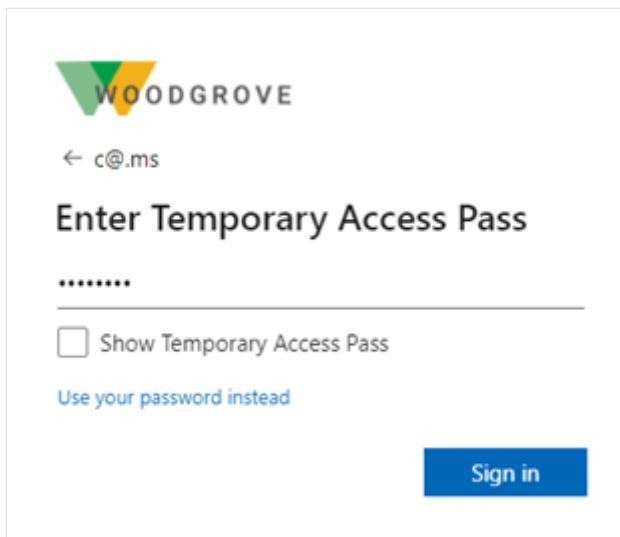
00aa00aa-bb11-cc22-dd33-44ee44ee44ee 5/22/2022 11:19:17 PM False True			
60	NotYetValid	23/05/2022 6:00:00 AM	

For more information, see [New-MgUserAuthenticationTemporaryAccessPassMethod](#) and [Get-MgUserAuthenticationTemporaryAccessPassMethod](#).

Use a Temporary Access Pass

The most common use for a TAP is for a user to register authentication details during the first sign-in or device setup, without the need to complete extra security prompts. Authentication methods are registered at <https://aka.ms/mysecurityinfo>. Users can also update existing authentication methods here.

1. Open a web browser to <https://aka.ms/mysecurityinfo>.
2. Enter the UPN of the account you created the TAP for, such as *tapuser@contoso.com*.
3. If the user is included in the TAP policy, they see a screen to enter their TAP.
4. Enter the TAP that was displayed in the Microsoft Entra admin center.



ⓘ Note

For federated domains, a TAP is preferred over federation. A user with a TAP completes the authentication in Microsoft Entra ID and isn't redirected to the federated Identity Provider (IdP).

The user is now signed in and can update or register a method such as FIDO2 security key. Users who update their authentication methods due to losing their credentials or device should make sure they remove the old authentication methods. Users can also continue to sign-in by using their password; a TAP doesn't replace a user's password.

User management of Temporary Access Pass

Users managing their security information at <https://aka.ms/mysecurityinfo> see an entry for the Temporary Access Pass. If a user doesn't have any other registered methods, they get a banner at the top of the screen that says to add a new sign-in method. Users can also see the TAP expiration time, and delete the TAP if it's no longer needed.

A screenshot of the 'My Sign-Ins' page in the Microsoft Entra portal. The left sidebar shows 'Overview', 'Security info' (which is selected), 'Organizations', 'Devices', and 'Privacy'. The main content area has a green banner at the top with the text 'ⓘ To maintain access to your account, add a sign in method.' Below this is a section titled 'Security info' with the sub-instruction 'These are the methods you use to sign into your account or reset your password.' It lists a single method: 'Temporary access pass' (with a delete link). At the bottom, there is a link 'Lost device? Sign out everywhere'.

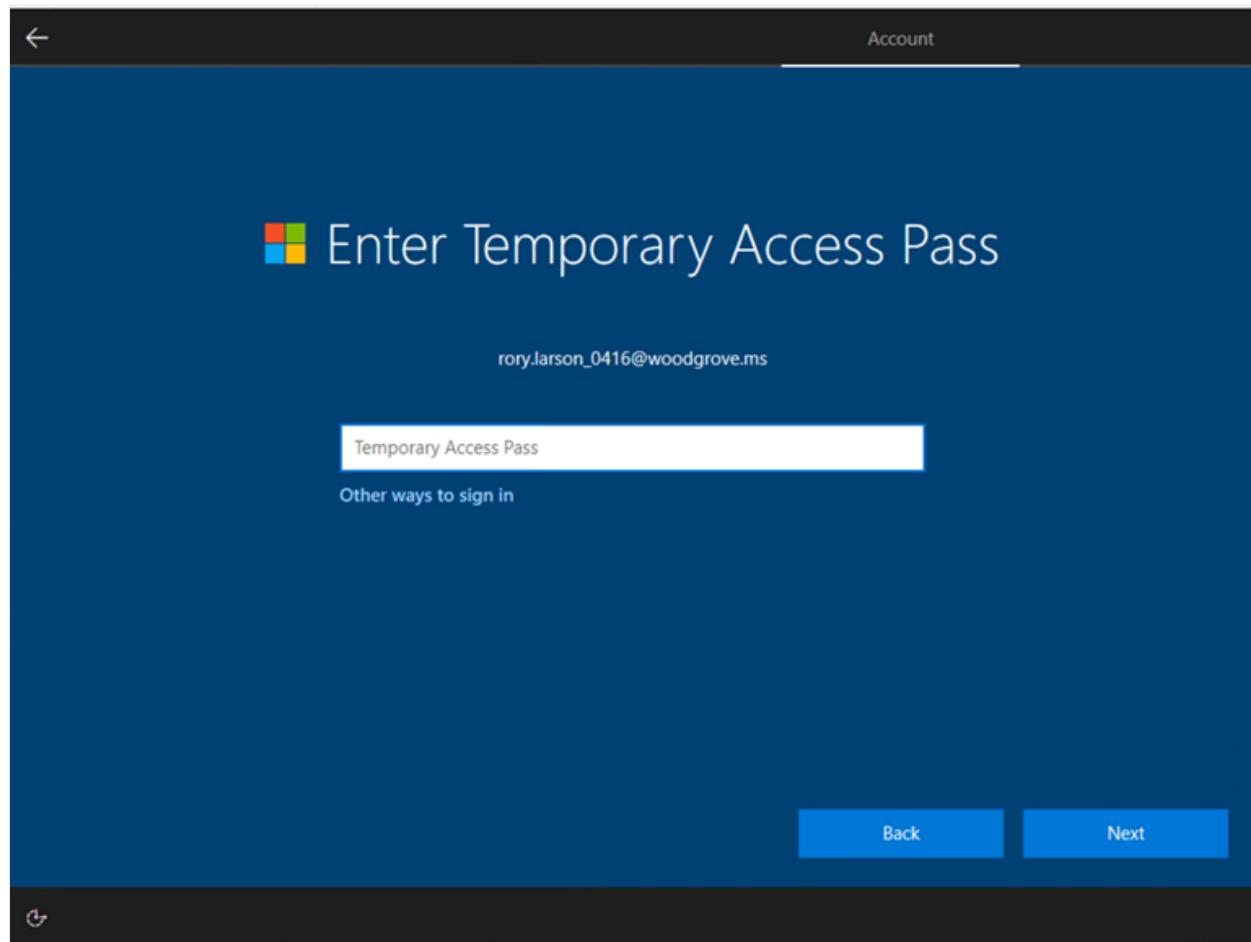
Windows device setup

Users with a TAP can navigate the setup process on Windows 10 and 11 to perform device join operations and configure Windows Hello for Business. TAP usage for setting up Windows Hello for Business varies based on the devices joined state.

For joined devices to Microsoft Entra ID:

- During the domain-join setup process, users can authenticate with a TAP (no password required) to join the device and register Windows Hello for Business.
- On already-joined devices, users must first authenticate with another method such as a password, smartcard, or FIDO2 key, before using TAP to set up Windows Hello for Business.
- If the [Web sign-in](#) feature on Windows is also enabled, the user can use TAP to sign into the device. This is intended only for completing initial device setup, or recovery when the user doesn't know or have a password.

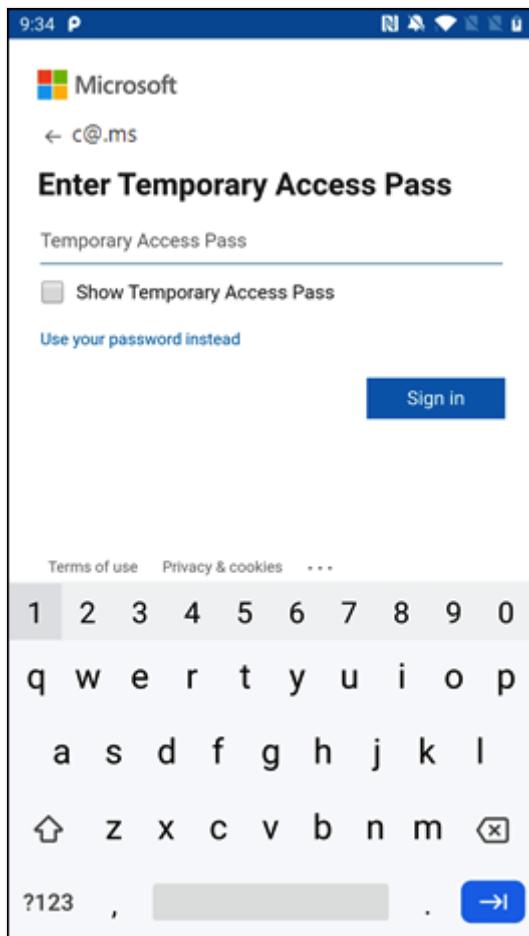
For hybrid-joined devices, users must first authenticate with another method such as a password, smartcard or FIDO2 key, before using TAP to set up Windows Hello for Business.



Using TAP with Microsoft Authenticator

Users can also use their TAP to register Microsoft Authenticator with their account. By adding a work or school account and signing in with a TAP users can register both passkeys and passwordless phone sign-in directly from the Authenticator app.

For more information, see [Add your work or school account to the Microsoft Authenticator app](#).



Guest access

You can add a TAP as a sign-in method to an internal guest, but not other types of guests. An internal guest has user object **UserType** set to **Guest**. They have authentication methods registered in Microsoft Entra ID. For more information about internal guests and other guest accounts, see [B2B guest user properties](#).

If you try to add a TAP to an external guest account in the Microsoft Entra admin center or in Microsoft Graph, you'll receive an error stating **Temporary Access Pass cannot be added to an external guest user**.

External guest users can sign-in to a resource tenant with a TAP issued by their home tenant if the TAP meets the home tenant authentication requirements and Cross Tenant Access policies have been configured to trust MFA from the users home tenant, see [Manage cross-tenant access settings for B2B collaboration](#).

Expiration

An expired or deleted TAP can't be used for interactive or non-interactive authentication.

Users need to reauthenticate with different authentication methods after the TAP is expired or deleted.

The token lifetime (session token, refresh token, access token, and so on) obtained by using a TAP sign-in is limited to the TAP lifetime. When a TAP expires, it leads to the expiration of the associated token.

Delete an expired Temporary Access Pass

Under the **Authentication methods** for a user, the **Detail** column shows when the TAP expired. You can delete an expired TAP using the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Administrator](#).
2. Browse to **Identity > Users**, select a user, such as *Tap User*, then choose **Authentication methods**.
3. On the right-hand side of the **Temporary Access Pass** authentication method shown in the list, select **Delete**.

You can also use PowerShell:

```
PowerShell
```

```
# Remove a user's Temporary Access Pass
Remove-MgUserAuthenticationTemporaryAccessPassMethod -UserId
user3@contoso.com -TemporaryAccessPassAuthenticationMethodId 00aa00aa-bb11-
cc22-dd33-44ee44ee44ee
```

For more information, see [Remove-MgUserAuthenticationTemporaryAccessPassMethod](#).

Replace a Temporary Access Pass

- Each user can only have one TAP. The passcode can be used during the start and end time of the TAP.
- If a user requires a new TAP:
 - If the existing TAP is valid, the admin can create a new TAP to override the existing valid TAP.
 - If the existing TAP has expired, a new TAP overrides the existing TAP.

For more information about NIST standards for onboarding and recovery, see [NIST Special Publication 800-63A](#).

Limitations

Keep these limitations in mind:

- When using a one-time TAP to register a passwordless method such as a FIDO2 security key or phone sign-in, the user must complete the registration within 10 minutes of sign-in with the one-time TAP. This limitation doesn't apply to a TAP that can be used more than once.
- Users in scope for self service password reset (SSPR) registration policy or [Microsoft Entra ID Protection multifactor authentication registration policy](#) are required to register authentication methods after they've signed in with a TAP using a browser. Users in scope for these policies are redirected to the [Interrupt mode of the combined registration](#). This experience doesn't currently support FIDO2 and phone sign-in registration.
- A TAP can't be used with the Network Policy Server (NPS) extension and Active Directory Federation Services (AD FS) adapter.
- It can take a few minutes for changes to replicate. Because of this, after a TAP is added to an account, it can take a while for the prompt to appear. For the same reason, after a TAP expires, users may still see a prompt for TAP.

Troubleshooting

- If a TAP isn't offered to a user during sign-in:
 - Make sure the user is in scope for TAP use in the Authentication methods policy.
 - Make sure the user has a valid TAP, and if it's one-time use, it wasn't used yet.
- If **Temporary Access Pass sign in was blocked due to User Credential Policy** appears during sign-in with a TAP:
 - Check that the user is in scope for the TAP policy
 - Make sure the user doesn't have a TAP for multiple use while the Authentication methods policy requires a one-time TAP.
 - Check if a one-time TAP was already used.
- If **Temporary Access Pass cannot be added to an external guest user** appears when you try to add a TAP to an account as an authentication method, the account is an external guest. Both internal and external guest accounts have an option to add a TAP for sign-in in the Microsoft Entra admin center and Microsoft Graph APIs. However, only internal guest accounts can be issued a TAP.

Next steps

- Plan a passwordless authentication deployment in Microsoft Entra ID
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Passwordless authentication options for Microsoft Entra ID

Article • 03/04/2025

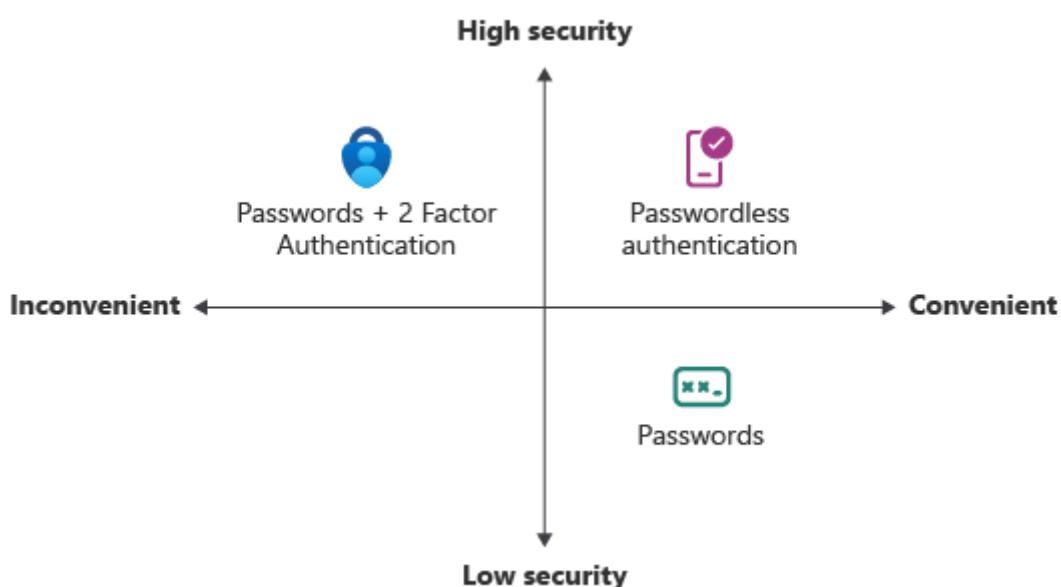
Features like multifactor authentication (MFA) are a great way to secure your organization, but users often get frustrated with the extra security layer on top of having to remember their passwords. Passwordless authentication methods are more convenient because the password is removed and replaced with something you have or something you are or know.

[\[...\] Expand table](#)

Authentication	Something you have	Something you are or know
Passwordless	Windows 10 Device, phone, or security key	Biometric or PIN

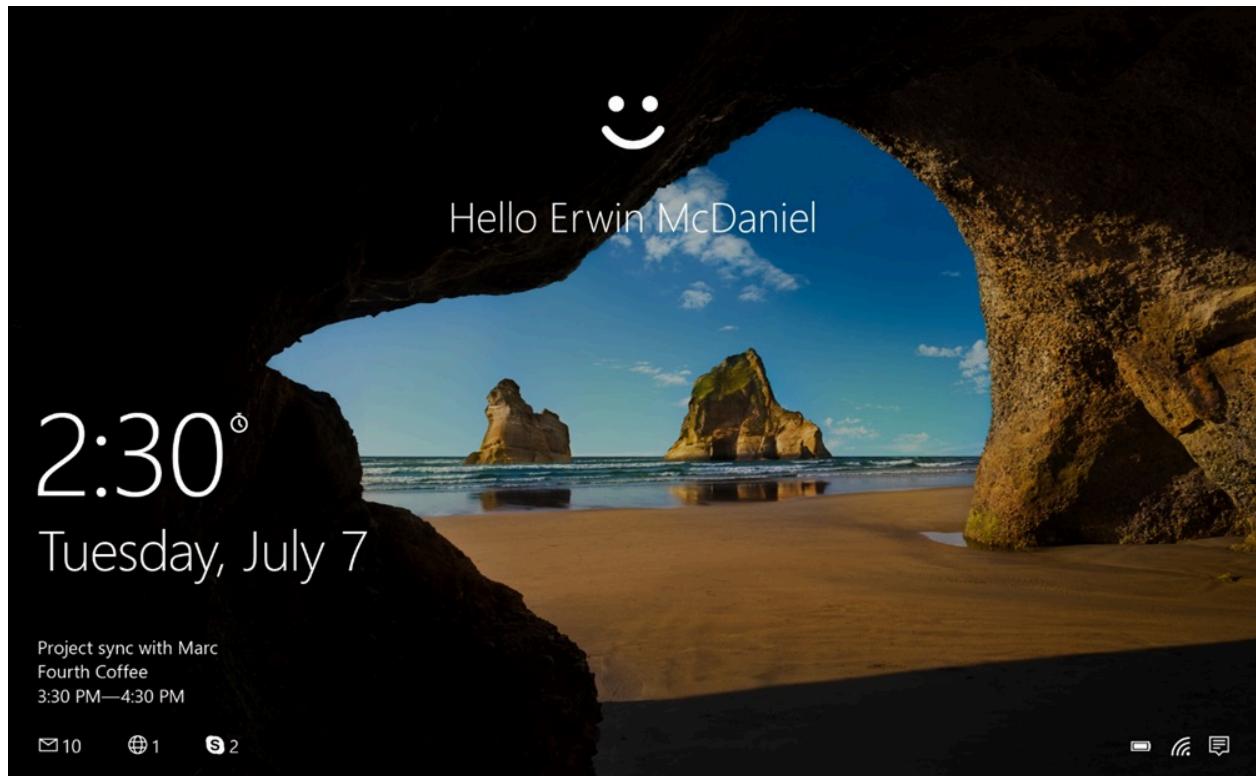
Each organization has different needs when it comes to authentication. Microsoft Entra ID and Azure Government integrate the following passwordless authentication options:

- Windows Hello for Business
- Platform Credential for macOS
- Platform single sign-on (PSSO) for macOS with smart card authentication
- Microsoft Authenticator
- Passkeys (FIDO2)
- Certificate-based authentication

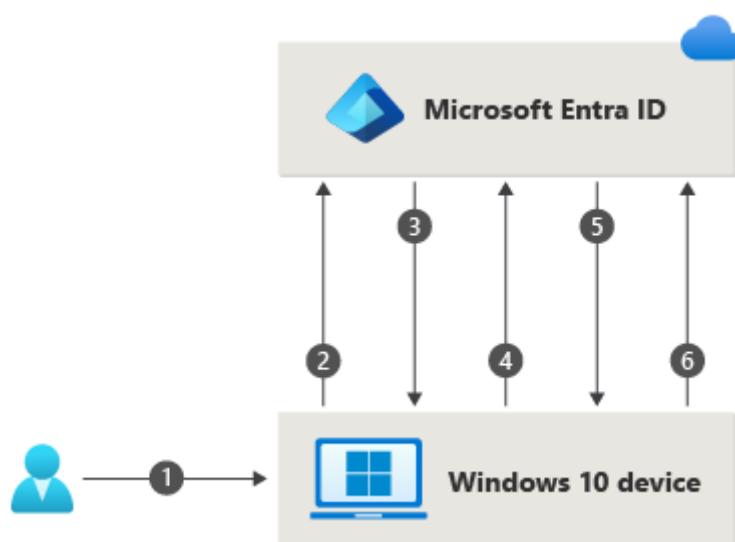


Windows Hello for Business

Windows Hello for Business is ideal for information workers that have their own designated Windows PC. The biometric and PIN credentials are directly tied to the user's PC, which prevents access from anyone other than the owner. With public key infrastructure (PKI) integration and built-in support for single sign-on (SSO), Windows Hello for Business provides a convenient method for seamlessly accessing corporate resources on-premises and in the cloud.



The following steps show how the sign-in process works with Microsoft Entra ID:



1. A user signs into Windows using biometric or PIN gesture. The gesture unlocks the Windows Hello for Business private key and is sent to the Cloud Authentication

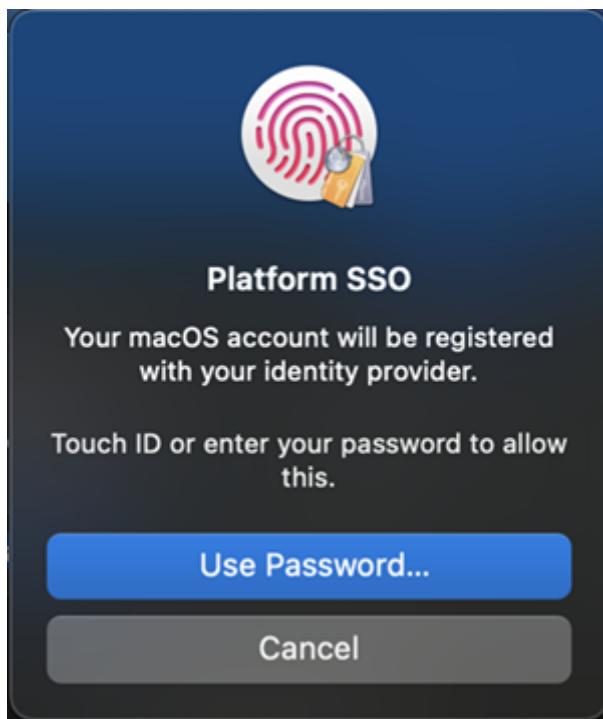
security support provider, called the *Cloud Authentication Provider* (*CloudAP*). For more information about CloudAP, see [What is a Primary Refresh Token?](#).

2. The CloudAP requests a nonce (a random arbitrary number that can be used once) from Microsoft Entra ID.
3. Microsoft Entra ID returns a nonce that's valid for 5 minutes.
4. The CloudAP signs the nonce using the user's private key and returns the signed nonce to the Microsoft Entra ID.
5. Microsoft Entra ID validates the signed nonce using the user's securely registered public key against the nonce signature. Microsoft Entra ID validates the signature, and then validates the returned signed nonce. When the nonce is validated, Microsoft Entra ID creates a primary refresh token (PRT) with session key that is encrypted to the device's transport key, and returns it to the CloudAP.
6. The CloudAP receives the encrypted PRT with session key. The CloudAP uses the device's private transport key to decrypt the session key, and protects the session key by using the device's Trusted Platform Module (TPM).
7. The CloudAP returns a successful authentication response to Windows. The user is then able to access Windows and cloud and on-premises applications by using seamless sign-on (SSO).

The Windows Hello for Business [planning guide](#) can be used to help you make decisions on the type of Windows Hello for Business deployment and the options you need to consider.

Platform Credential for macOS

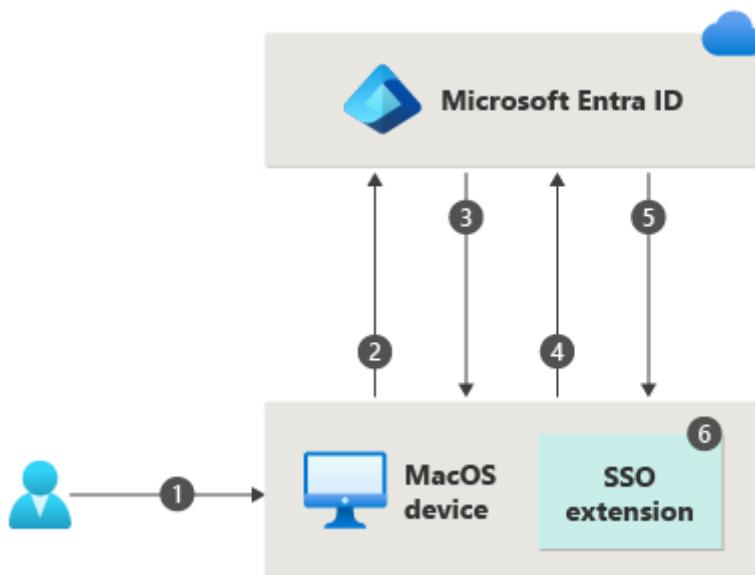
Platform Credential for macOS is a new capability on macOS that is enabled using the Microsoft Enterprise single sign-on Extension (SSOe). It provisions a secure enclave backed hardware-bound cryptographic key that is used for SSO across apps that use Microsoft Entra ID for authentication. The user's local account password is not affected and is required to log on to the Mac.



Platform Credential for macOS allows users to go passwordless by configuring Touch ID to unlock the device, and uses phish-resistant credentials, based on Windows Hello for Business technology. This saves customer organizations money by removing the need for security keys and advances Zero Trust objectives using integration with the Secure Enclave.

Platform Credential for macOS can also be used as a phishing-resistant credential for use in WebAuthn challenges, including browser re-authentication scenarios.

Authentication Policy Administrators need to enable the **Passkey (FIDO2)** authentication method to support Platform Credential for macOS as a phishing-resistant credential. If you use Key Restriction Policies in your FIDO policy, you need to add the AAGUID for the macOS Platform Credential to your list of allowed AAGUIDs: 7FD635B3-2EF9-4542-8D9D-164F2C771EFC.



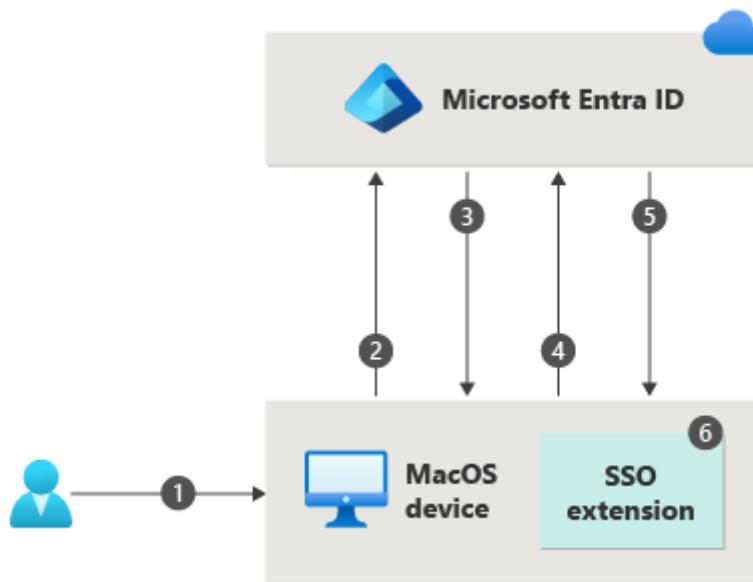
1. A user unlocks macOS using fingerprint or password gesture, which unlocks the key bag to provide access to UserSecureEnclaveKey.
2. The macOS requests a nonce (a random arbitrary number that can be used just once) from Microsoft Entra ID.
3. Microsoft Entra ID returns a nonce that's valid for 5 minutes.
4. The operating system (OS) sends a login request to Microsoft Entra ID with an embedded assertion signed with the UserSecureEnclaveKey that resides in the Secure Enclave.
5. Microsoft Entra ID validates the signed assertion using the user's securely registered public key of UserSecureEnclave key. Microsoft Entra ID validates the signature and nonce. Once the assertion is validated, Microsoft Entra ID creates a **primary refresh token (PRT)** encrypted with the public key of the UserDeviceEncryptionKey that is exchanged during registration and sends the response back to the OS.
6. The OS decrypts and validates the response, retrieves the SSO tokens, stores and shares it with the SSO extension for providing SSO. The user is able to access macOS, cloud and on-premises applications by using SSO.

Refer to [macOS Platform SSO](#) for more information on how to configure and deploy Platform Credential for macOS.

Platform single sign-on for macOS with SmartCard

Platform single sign-on (PSSO) for macOS allows users to go passwordless using the SmartCard authentication method. The user signs in to the machine using an external smart card, or smart card-compatible hard token (such as Yubikey). Once the device is unlocked, the smart card is used with Microsoft Entra ID to grant SSO across apps that use Microsoft Entra ID for authentication using [certificate-based authentication \(CBA\)](#). CBA needs to be configured and enabled for users for this feature to work. For configuring CBA, refer to [How to configure Microsoft Entra certificate-based authentication](#).

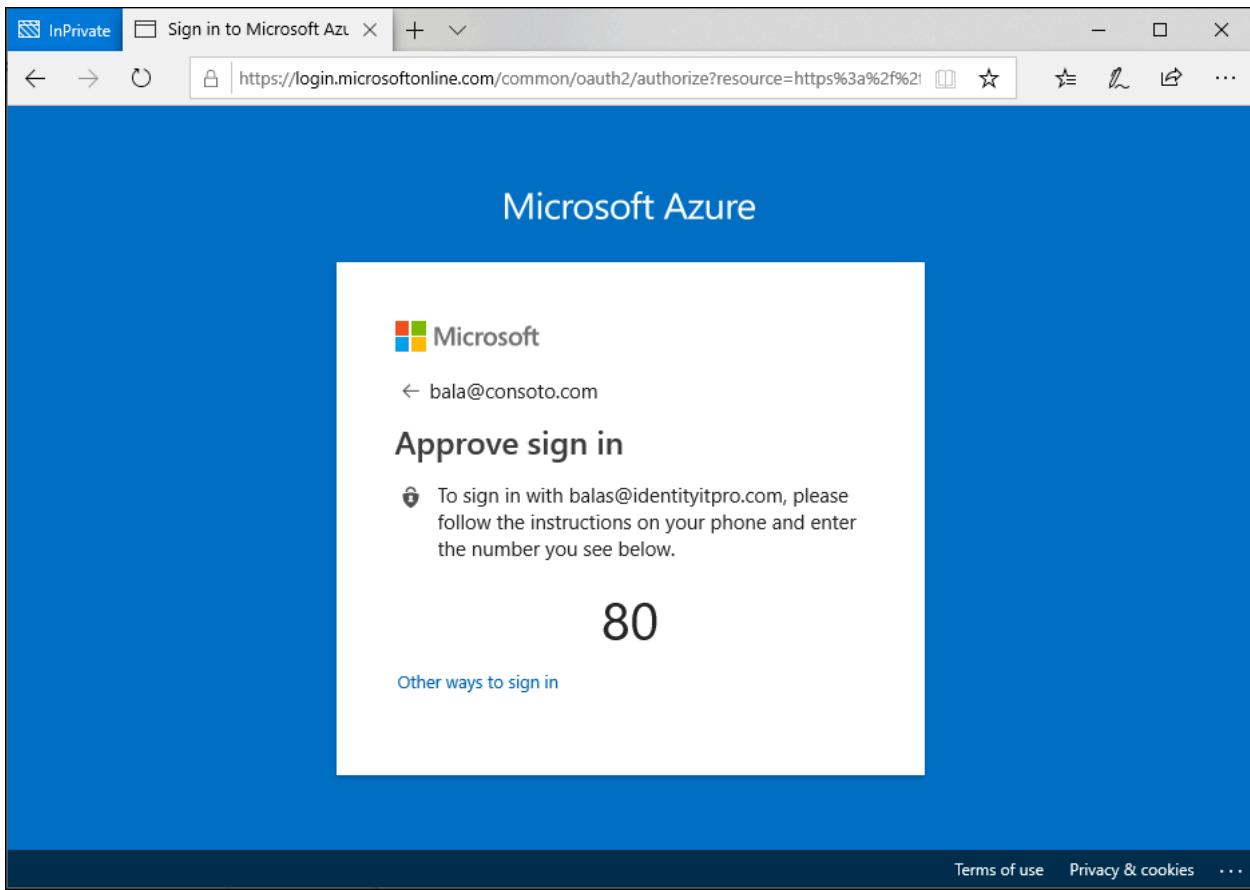
To enable it, an administrator needs to configure PSSO by using Microsoft Intune or another supported Mobile Device Management (MDM) solution.



1. A user unlocks macOS using smart card pin, which unlocks the smart card and the key bag to provide access to device registration keys present in Secure Enclave.
2. The macOS requests a nonce (a random arbitrary number that can be used only once) from Microsoft Entra ID.
3. Microsoft Entra ID returns a nonce that's valid for 5 minutes.
4. The operating system (OS) sends a login request to Microsoft Entra ID with an embedded assertion signed with the user's Microsoft Entra certificate from the smart card.
5. Microsoft Entra ID validates the signed assertion, signature and nonce. Once the assertion is validated, Microsoft Entra ID creates a [primary refresh token \(PRT\)](#) encrypted with the public key of the UserDeviceEncryptionKey that is exchanged during registration and sends the response back to the OS.
6. The OS decrypts and validates the response, retrieves the SSO tokens, stores and shares it with the SSO extension for providing SSO. The user is able to access macOS, cloud and on-premises applications by using SSO.

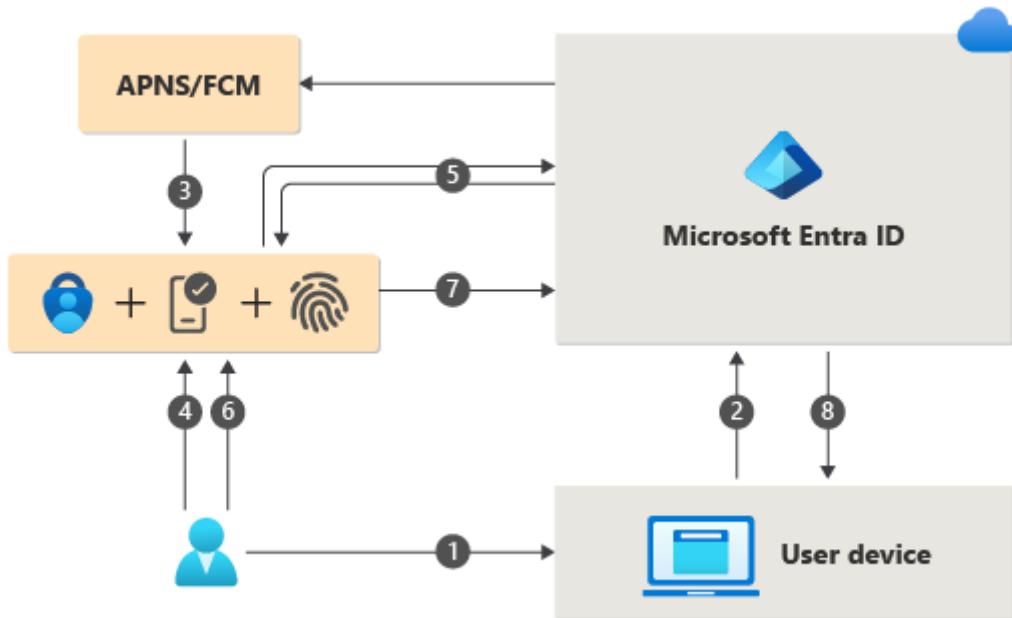
Microsoft Authenticator

You can also allow your employee's phone to become a passwordless authentication method. You could already be using the Authenticator app as a convenient multifactor authentication option in addition to a password. You can also use the Authenticator App as a passwordless option.



The Authenticator App turns any iOS or Android phone into a strong, passwordless credential. Users can sign in to any platform or browser by getting a notification to their phone, matching a number displayed on the screen to the one on their phone. Then they can use their biometric (touch or face) or PIN to confirm. For installation details, see [Download and install the Microsoft Authenticator](#).

Passwordless authentication using Microsoft Authenticator follows the same basic pattern as Windows Hello for Business. It's a little more complicated as the user needs to be identified so that Microsoft Entra ID can find the Authenticator app version being used:



1. The user enters their username.
2. Microsoft Entra ID detects that the user has a strong credential and starts the Strong Credential flow.
3. A notification is sent to the app via Apple Push Notification Service (APNS) on iOS devices, or via Firebase Cloud Messaging (FCM) on Android devices.
4. The user receives the push notification and opens the app.
5. The app calls Microsoft Entra ID and receives a proof-of-presence challenge and nonce.
6. The user completes the challenge by entering their biometric or PIN to unlock private key.
7. The nonce is signed with the private key and sent back to Microsoft Entra ID.
8. Microsoft Entra ID performs public/private key validation and returns a token.

To get started with passwordless sign-in, complete the following how-to:

[Enable passwordless sign using the Authenticator app](#)

Passkeys (FIDO2)

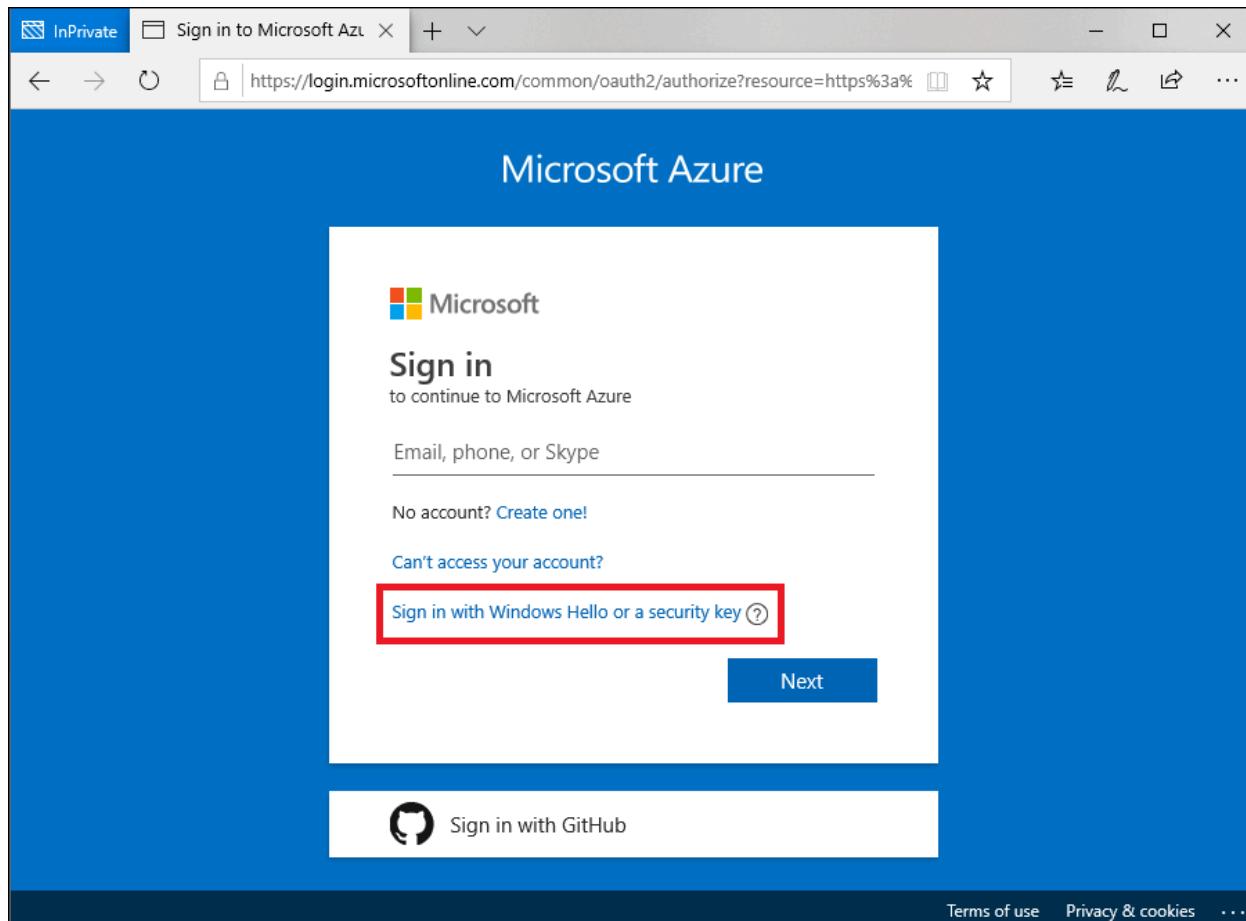
Users can register a passkey (FIDO2) and choose it as their primary sign-in method. With a hardware device that handles the authentication, the security of an account is increased as there's no password that can be exposed or guessed. Currently in preview, an Authentication Administrator can also [provision a FIDO2 security key](#) on behalf of a user by using Microsoft Graph API and a custom client. Provisioning on behalf of users is currently limited to security keys at this time.

The FIDO (Fast IDentity Online) Alliance helps to promote open authentication standards and reduce the use of passwords as a form of authentication. FIDO2 is the latest standard that incorporates the web authentication (WebAuthn) standard. FIDO allows organizations to apply the WebAuthn standard by using an external security key, or a platform key built into a device, to sign in without a username or password.

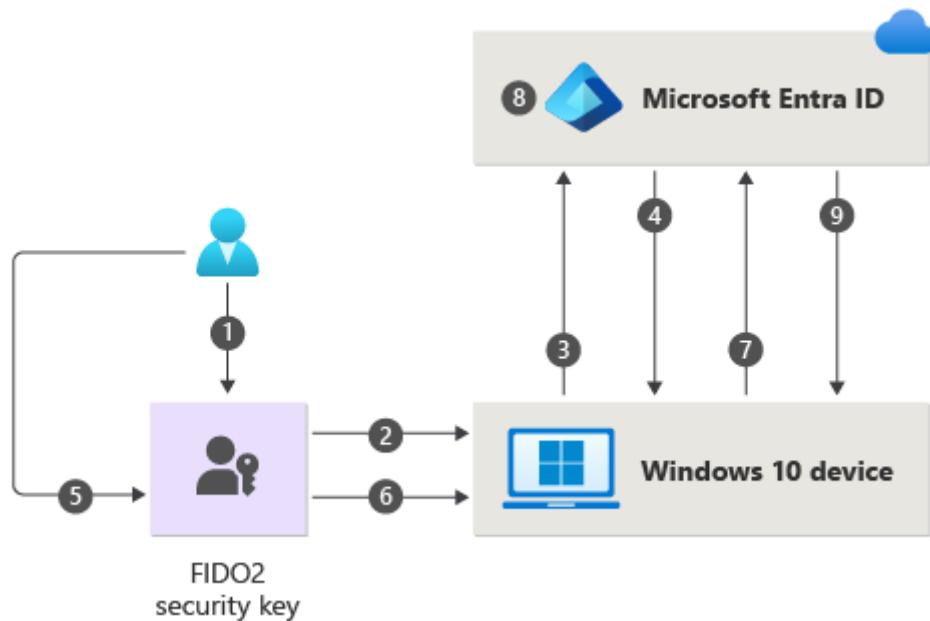
FIDO2 security keys are an unphishable standards-based passwordless authentication method that can come in any form factor. They're commonly USB devices, but they can also use Bluetooth or near-field communication (NFC). Passkeys (FIDO2) are based on the same WebAuthn standard and can be saved in Authenticator, or on mobile devices, tablets, or computers.

FIDO2 security keys can be used to sign in to their Microsoft Entra ID or Microsoft Entra hybrid joined Windows 10 devices and get single-sign on to their cloud and on-premises resources. Users can also sign in to supported browsers. FIDO2 security keys are a great option for enterprises who are very security sensitive or have scenarios or employees who aren't willing or able to use their phone as a second factor.

For more information about passkey (FIDO2) support, see [Support for passkey \(FIDO2\) authentication with Microsoft Entra ID](#). For developer best practices, see [Support FIDO2 auth in the applications they develop](#).



The following process is used when a user signs in with a FIDO2 security key:



1. The user plugs the FIDO2 security key into their computer.
2. Windows detects the FIDO2 security key.
3. Windows sends an authentication request.
4. Microsoft Entra ID sends back a nonce.
5. The user completes their gesture to unlock the private key stored in the FIDO2 security key's secure enclave.
6. The FIDO2 security key signs the nonce with the private key.
7. The primary refresh token (PRT) token request with signed nonce is sent to Microsoft Entra ID.
8. Microsoft Entra ID verifies the signed nonce using the FIDO2 public key.
9. Microsoft Entra ID returns PRT to enable access to on-premises resources.

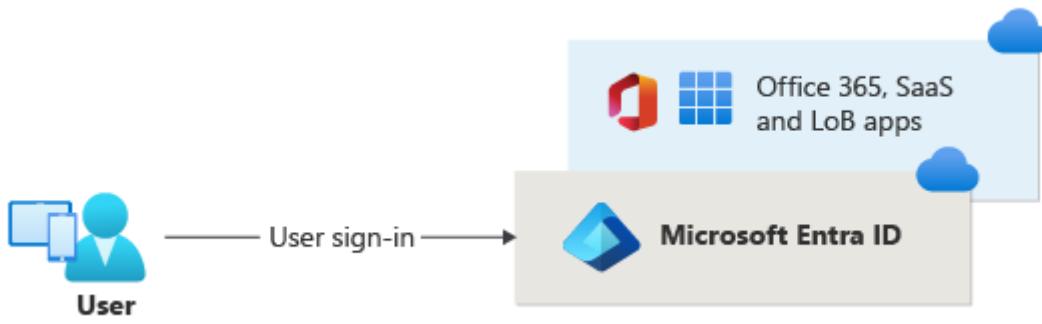
For a list FIDO2 security key providers, see [Become a Microsoft-compatible FIDO2 security key vendor](#).

To get started with FIDO2 security keys, complete the following how-to:

[Enable passwordless sign using FIDO2 security keys](#)

Certificate-based authentication

Microsoft Entra certificate-based authentication (CBA) enables customers to allow or require users to authenticate directly with X.509 certificates against their Microsoft Entra ID for applications and browser sign-in. CBA enables customers to adopt phishing-resistant authentication and sign in with an X.509 certificate against their Public Key Infrastructure (PKI).



Key benefits of using Microsoft Entra CBA

 Expand table

Benefits	Description
Great user experience	<ul style="list-style-type: none"> - Users who need certificate-based authentication can now directly authenticate against Microsoft Entra ID and not have to invest in federation. - Portal UI enables users to easily configure how to map certificate fields to a user object attribute to look up the user in the tenant (certificate username bindings) - Portal UI to configure authentication policies to help determine which certificates are single-factor versus multifactor.
Easy to deploy and administer	<ul style="list-style-type: none"> - Microsoft Entra CBA is a free feature, and you don't need any paid editions of Microsoft Entra ID to use it. - No need for complex on-premises deployments or network configuration. - Directly authenticate against Microsoft Entra ID.
Secure	<ul style="list-style-type: none"> - On-premises passwords don't need to be stored in the cloud in any form. - Protects your user accounts by working seamlessly with Microsoft Entra Conditional Access policies, including Phishing-Resistant multifactor authentication (MFA requires licensed edition) and blocking legacy authentication. - Strong authentication support where users can define authentication policies through the certificate fields, such as issuer or policy OID (object identifiers), to determine which certificates qualify as single-factor versus multifactor. - The feature works seamlessly with Conditional Access features and authentication strength capability to enforce MFA to help secure your users.

Supported scenarios

The following scenarios are supported:

- User sign-ins to web browser-based applications on all platforms.
- User sign-ins to Office mobile apps on iOS/Android platforms and Office native apps in Windows, including Outlook, OneDrive, and so on.

- User sign-ins on mobile native browsers.
- Support for granular authentication rules for multifactor authentication by using the certificate issuer **Subject** and **policy OIDs**.
- Configuring certificate-to-user account bindings by using any of the certificate fields:
 - Subject Alternate Name (SAN) PrincipalName and SAN RFC822Name
 - Subject Key Identifier (SKI) and SHA1PublicKey
- Configuring certificate-to-user account bindings by using any of the user object attributes:
 - User Principal Name
 - onPremisesUserPrincipalName
 - CertificateUserIds

Supported scenarios

The following considerations apply:

- Administrators can enable passwordless authentication methods for their tenant.
- Administrators can target all users or select users/Security groups within their tenant for each method.
- Users can register and manage these passwordless authentication methods in their account portal.
- Users can sign in with these passwordless authentication methods:
 - Authenticator app: Works in scenarios where Microsoft Entra authentication is used, including across all browsers, during Windows 10 setup, and with integrated mobile apps on any operating system.
 - Security keys: Work on lock screen for Windows 10 and the web in supported browsers like Microsoft Edge (both legacy and new Edge).
- Users can use passwordless credentials to access resources in tenants where they're a guest, but they could still be required to perform MFA in that resource tenant. For more information, see [Possible double multifactor authentication](#).
- Users can't register passwordless credentials within a tenant where they're a guest, the same way that they don't have a password managed in that tenant.

Unsupported scenarios

We recommend no more than 20 sets of keys for each passwordless method for any user account. As more keys are added, the user object size increases, and you could notice degradation for some operations. In that case, you should remove unnecessary keys. For more information and the PowerShell cmdlets to query and remove keys, see

Using WHfBTools PowerShell module for cleaning up orphaned Windows Hello for Business Keys [♂](#). Use the `/UserPrincipalName` optional parameter to query only keys for a specific user. The permissions required are to run as an administrator or the specified user.

When you use PowerShell to create a CSV file with all of the existing keys, carefully identify the keys that you need to keep, and remove those rows from the CSV. Then use the modified CSV with PowerShell to delete the remaining keys to bring the account key count under the limit.

It's safe to delete any key reported as "Orphaned"="True" in the CSV. An orphaned key is one for a device that isn't longer registered in Microsoft Entra ID. If removing all Orphans still doesn't bring the User account below the limit, it's necessary to look at the `DeviceId` and `CreationTime` columns to identify which keys to target for deletion. Be careful to remove any row in the CSV for keys you want to keep. Keys for any DeviceID corresponding to devices the user actively uses should be removed from the CSV before the deletion step.

Choose a passwordless method

The choice between these three passwordless options depends on your company's security, platform, and app requirements.

Here are some factors for you to consider when choosing Microsoft passwordless technology:

[\[+\] Expand table](#)

	Windows Hello for Business	Passwordless sign-in with the Authenticator app	FIDO2 security keys
Pre-requisite	Windows 10, version 1809 or later Microsoft Entra ID	Authenticator app Phone (iOS and Android devices)	Windows 10, version 1903 or later Microsoft Entra ID
Mode	Platform	Software	Hardware
Systems and devices	PC with a built-in Trusted Platform Module (TPM) PIN and biometrics recognition	PIN and biometrics recognition on phone	FIDO2 security devices that are Microsoft compatible
User experience	Sign in using a PIN or biometric recognition (facial,	Sign in using a mobile phone with	Sign in using FIDO2 security device (biometrics, PIN, and

	Windows Hello for Business	Passwordless sign-in with the Authenticator app	FIDO2 security keys
	iris, or fingerprint) with Windows devices. Windows Hello authentication is tied to the device; the user needs both the device and a sign-in component such as a PIN or biometric factor to access corporate resources.	fingerprint scan, facial or iris recognition, or PIN. Users sign in to work or personal account from their PC or mobile phone.	NFC) User can access device based on organization controls and authenticate based on PIN, biometrics using devices such as USB security keys and NFC-enabled smartcards, keys, or wearables.
Enabled scenarios	Password-less experience with Windows device. Applicable for dedicated work PC with ability for single sign-on to device and applications.	Password-less anywhere solution using mobile phone. Applicable for accessing work or personal applications on the web from any device.	Password-less experience for workers using biometrics, PIN, and NFC. Applicable for shared PCs and where a mobile phone isn't a viable option (such as for help desk personnel, public kiosk, or hospital team)

Use the following table to choose which method supports your requirements and users.

[\[+\] Expand table](#)

Persona	Scenario	Environment	Passwordless technology
Admin	Secure access to a device for management tasks	Assigned Windows 10 device	Windows Hello for Business and/or FIDO2 security key
Admin	Management tasks on non-Windows devices	Mobile or non Windows device	Passwordless sign-in with the Authenticator app
Information worker	Productivity work	Assigned Windows 10 device	Windows Hello for Business and/or FIDO2 security key
Information worker	Productivity work	Mobile or non Windows device	Passwordless sign-in with the Authenticator app
Frontline worker	Kiosks in a factory, plant, retail, or data entry	Shared Windows 10 devices	FIDO2 Security keys

Next steps

To get started with passwordless in Microsoft Entra ID, complete one of the following how-tos:

- [Enable FIDO2 security key passwordless sign-in](#)
- [Enable phone-based passwordless sign-in with the Authenticator app](#)

External Links

- [FIDO Alliance ↗](#)
 - [FIDO2 Client to Authenticator Protocol \(CTAP\) specification ↗](#)
-

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Authentication methods in Microsoft Entra ID - Microsoft Authenticator app

Article • 02/11/2025

Microsoft Authenticator provides another level of security to your Microsoft Entra work or school account or your Microsoft account. It's available for [Android](#) and [iOS](#).

With the Microsoft Authenticator app, users can authenticate in a passwordless way during sign-in. They can also use it as a verification option during self-service password reset (SSPR) or multifactor authentication (MFA) events.

Microsoft Authenticator supports passkey, passwordless sign in, and MFA by using notifications and verification codes.

- Users can sign in with a passkey in the Authenticator app and complete phishing-resistant authentication with their biometric sign-in or device PIN.
- Users can set up Authenticator notifications and sign in with Authenticator instead of their username and password.
- Users can receive an MFA request on their mobile device, and approve or deny the sign-in attempt from their phone.
- They can also use an OATH verification code in the Authenticator app and enter it in a sign-in interface.

For more information, see [Enable passwordless sign-in with the Microsoft Authenticator](#).

ⓘ Note

Android users with Company Portal versions below 2111 (5.0.5333.0) can't register Authenticator until they update their Company Portal application to a newer version.

Passkey sign-in

Authenticator is a free passkey solution that lets users do passwordless phishing-resistant authentications from their own phones. Some key benefits to using passkeys in the Authenticator app:

- Passkeys can be easily deployed at scale. Then passkeys are available on a user's phone for both mobile device management (MDM) and bring your own device (BYOD) scenarios.

- Passkeys in Authenticator come at no more cost and travel with the user wherever they go.
- Passkeys in Authenticator are device-bound which ensures the passkey doesn't leave the device on which it was created.
- Users stay up-to-date with latest passkey innovation based upon open WebAuthn standards.
- Enterprises can layer other capabilities on top of authentication flows such as [Federal Information Processing Standards \(FIPS\) 140 compliance](#).

Device-bound passkey

Passkeys in the Authenticator app are device-bound to ensure that they never leave the device they were created on. On an iOS device, Authenticator uses the Secure Enclave to create the passkey. On Android, we create the passkey in the Secure Element on devices that support it, or fall back to the Trusted Execution Environment (TEE).

How passkey attestation works with Authenticator

When attestation is enabled in the [Passkey \(FIDO2\)](#) policy, Microsoft Entra ID attempts to verify the legitimacy of the security key model or passkey provider where the passkey is being created. When a user registers a passkey in Authenticator, attestation verifies that the legitimate Microsoft Authenticator app created the passkey by using Apple and Google services. Here are details for how attestation works for each platform:

- iOS: Authenticator attestation uses the [iOS App Attest service](#) to ensure the legitimacy of the Authenticator app before registering the passkey.
- Android:
 - For Play Integrity attestation, Authenticator attestation uses the [Play Integrity API](#) to ensure the legitimacy of the Authenticator app before registering the passkey.
 - For Key attestation, Authenticator attestation uses [key attestation by Android](#) to verify that the passkey being registered is hardware-backed.

Note

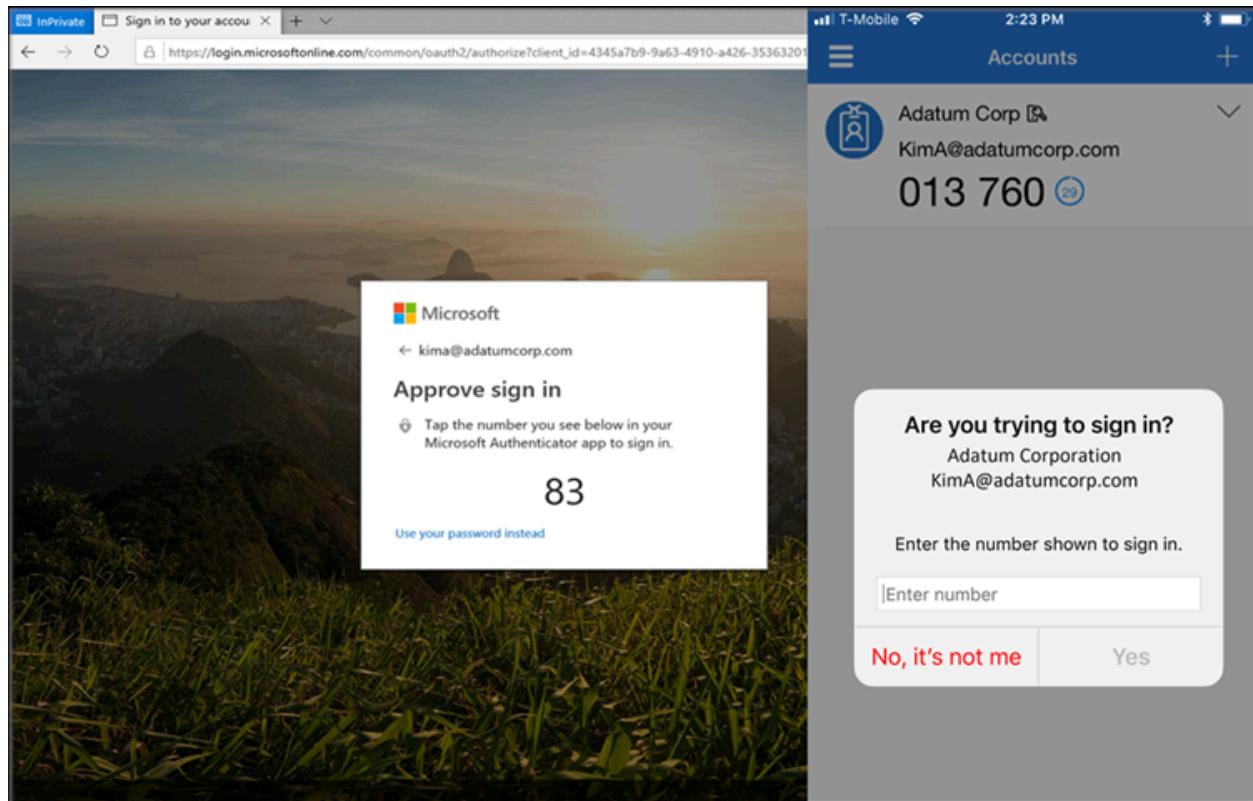
For both iOS and Android, Authenticator attestation relies upon Apple and Google services to verify the authenticity of the Authenticator app. Heavy service usage can make passkey registration fail, and users may need to try again. If Apple and Google services are down, Authenticator attestation blocks registration that requires attestation until services are restored. To monitor the status of Google Play

Integrity service, see [Google Play Status Dashboard](#). To monitor the status of the iOS App Attest service, see [System Status](#).

For more information about how to configure attestation, see [How to enable passkeys in Microsoft Authenticator for Microsoft Entra ID](#).

Passwordless sign-in via notifications

Instead of seeing a prompt for a password after entering a username, users who enable phone sign-in from the Authenticator app sees a message to enter a number in their app. When the correct number is selected, the sign-in process is complete.



This authentication method provides a high level of security, and removes the need for the user to provide a password at sign-in.

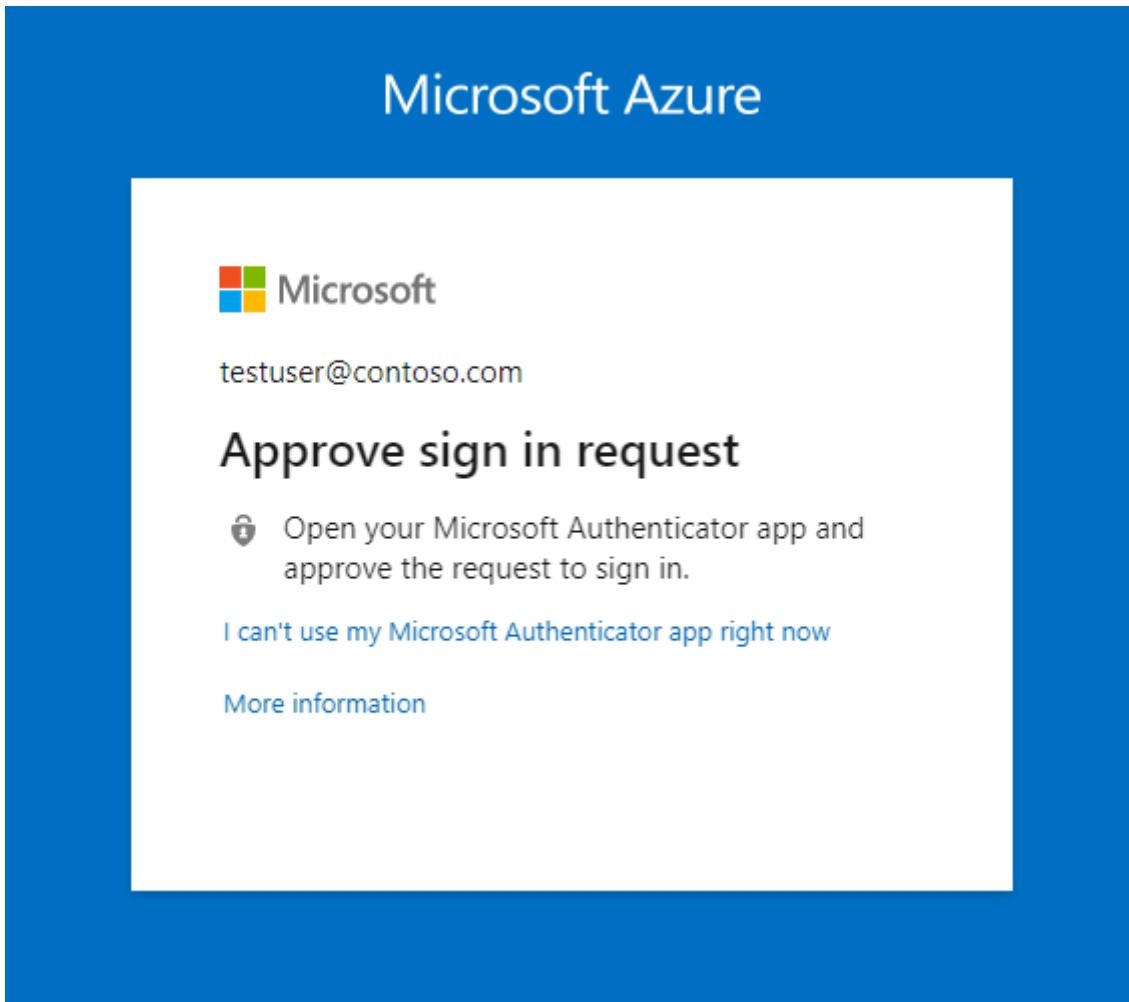
To get started with passwordless sign-in, see [Enable passwordless sign-in with the Microsoft Authenticator](#).

MFA via notifications through mobile app

The Authenticator app can help prevent unauthorized access to accounts and stop fraudulent transactions by pushing a notification to your smartphone or tablet. Users view the notification, and if it's legitimate, select **Verify**. Otherwise, they can select **Deny**.

Note

Starting in August, 2023, anomalous sign-ins don't generate notifications, similarly to how sign-ins from unfamiliar locations don't generate notifications. To approve an anomalous sign-in, users can open Microsoft Authenticator, or Authenticator Lite in a relevant companion app like Outlook. Then they can either pull down to refresh or tap **Refresh**, and approve the request.



In China, the *Notification through mobile app* method on Android devices doesn't work because as Google play services (including push notifications) are blocked in the region. However, iOS notifications do work. For Android devices, alternate authentication methods should be made available for those users.

Verification code from mobile app

The Authenticator app can be used as a software token to generate an OATH verification code. After entering your username and password, you enter the code provided by the Authenticator app into the sign-in interface. The verification code provides a second form of authentication.

Note

OATH verification codes generated by Authenticator aren't supported for certificate-based authentication.

Users can have a combination of up to five OATH hardware tokens or authenticator applications, such as the Authenticator app, configured for use at any time.

FIPS 140 compliant for Microsoft Entra authentication

Consistent with the guidelines outlined in [National Institute of Standards and Technologies \(NIST\) Special Publication 800-63B](#), authenticators used by US government agencies are required to use FIPS 140 validated cryptography. This guideline helps US government agencies meet the requirements of Executive Order (EO) 14028. Additionally, this guideline helps other regulated industries such as healthcare organizations working with [Electronic Prescriptions for Controlled Substances \(EPCS\)](#) meet their regulatory requirements.

FIPS 140 is a US government standard that defines minimum security requirements for cryptographic modules in information technology products and systems. The [Cryptographic Module Validation Program \(CMVP\)](#) maintains the testing against the FIPS 140 standard.

Microsoft Authenticator for iOS

Beginning with version 6.6.8, Microsoft Authenticator for iOS uses the native Apple CoreCrypto module for FIPS validated cryptography on Apple iOS FIPS 140 compliant devices. All Microsoft Entra authentications using phishing-resistant device-bound passkeys, push multifactor authentications (MFA), passwordless phone sign-in (PSI), and time-based one-time passcodes (TOTP) use the FIPS cryptography.

For more information about the FIPS 140 validated cryptographic modules that are used and compliant iOS devices, see [Apple iOS security certifications](#).

Microsoft Authenticator for Android

Beginning with version 6.2409.6094 on Microsoft Authenticator for Android, all authentications in Microsoft Entra ID, including passkeys, are considered FIPS-compliant. Authenticator uses the wolfSSL Inc. cryptographic module to achieve FIPS 140, Security

Level 1 compliance on Android devices. For more information about the certification, see [Cryptographic Module Validation Program](#).

Determining Microsoft Authenticator registration type in Security info

Users can access [Security info](#) (see the URLs in the next section) or by selecting Security info from MyAccount to manage and add more Microsoft Authenticator registrations. Specific icons are used to differentiate whether the Microsoft Authenticator registration is passwordless phone sign-in or MFA.

[+] Expand table

Authenticator registration type	Icon
Microsoft Authenticator: Passwordless phone sign-in	
Microsoft Authenticator: (Notification/Code)	

SecurityInfo links

[+] Expand table

Cloud	Security info URL
Azure commercial (includes Government Community Cloud (GCC))	https://aka.ms/MySecurityInfo
Azure for US Government (includes GCC High and DoD)	https://aka.ms/MySecurityInfo-us

Updates to Authenticator

Microsoft continuously updates Authenticator to maintain a high level of security. To ensure that your users are getting the best experience possible, we recommend having them continuously update their Authenticator App. In the case of critical security updates, app versions that aren't up-to-date may not work, and may block users from completing their authentication. If a user is using a version of the app that isn't supported, they're prompted to upgrade to the latest version before they proceed to sign in.

Microsoft also periodically retires older versions of the Authenticator App to maintain a high security bar for your organization. If a user's device doesn't support modern versions of Microsoft Authenticator, they can't sign in with the app. We recommend they sign in with an OATH verification code in Microsoft Authenticator to complete MFA.

Next steps

- To get started with passkeys, see [How to enable passkeys in Microsoft Authenticator for Microsoft Entra ID](#).
 - For more information about passwordless sign-in, see [Enable passwordless sign-in with the Microsoft Authenticator](#).
 - Learn more about configuring authentication methods using the [Microsoft Graph REST API](#).
-

Feedback

Was this page helpful?



Yes



No

[Provide product feedback ↗](#)

Overview of Microsoft Entra certificate-based authentication

Article • 03/04/2025

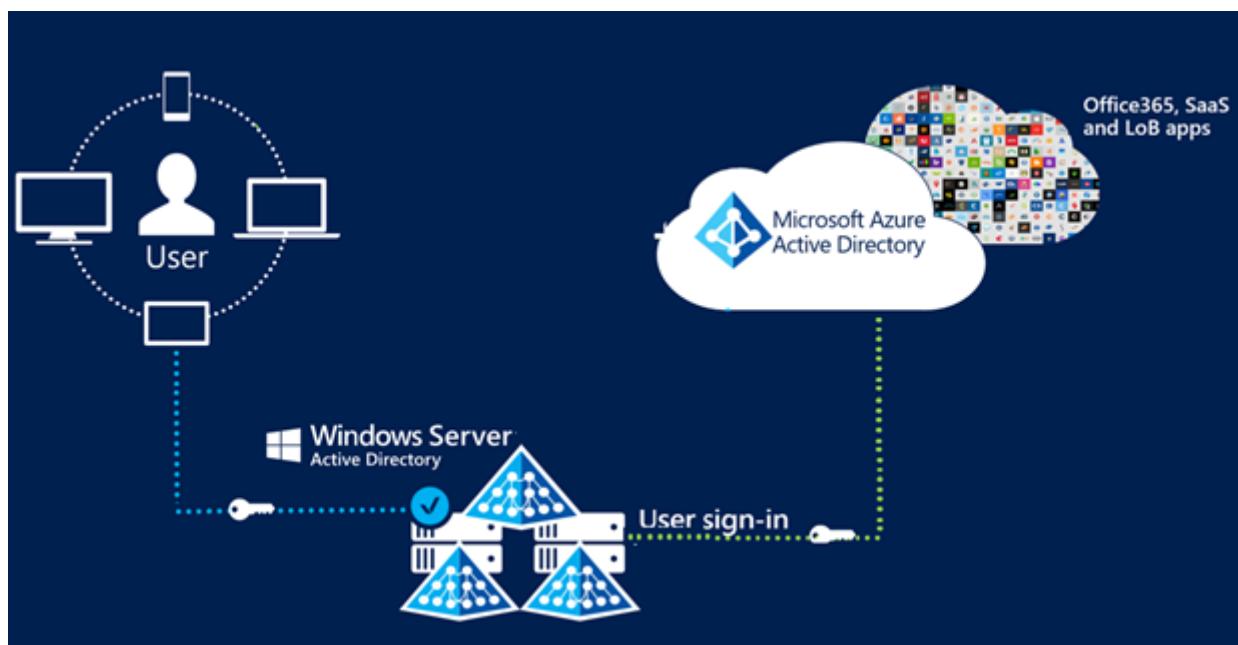
Microsoft Entra certificate-based authentication (CBA) enables customers to allow or require users to authenticate directly with X.509 certificates against their Microsoft Entra ID for applications and browser sign-in. This feature enables customers to adopt a phishing resistant authentication and authenticate with an X.509 certificate against their Public Key Infrastructure (PKI).

What is Microsoft Entra CBA?

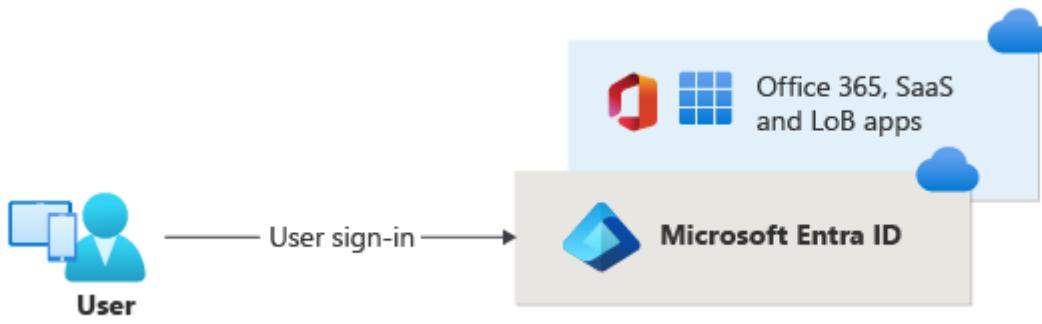
Before cloud-managed support for CBA to Microsoft Entra ID, customers had to implement federated certificate-based authentication, which requires deploying Active Directory Federation Services (AD FS) to be able to authenticate using X.509 certificates against Microsoft Entra ID. With Microsoft Entra certificate-based authentication, customers can authenticate directly against Microsoft Entra ID and eliminate the need for federated AD FS, with simplified customer environments and cost reduction.

The following images show how Microsoft Entra CBA simplifies the customer environment by eliminating federated AD FS.

Certificate-based authentication with federated AD FS



Microsoft Entra certificate-based authentication



Key benefits of using Microsoft Entra CBA

[Expand table](#)

Benefits	Description
Great user experience	<ul style="list-style-type: none"> - Users who need certificate-based authentication can now directly authenticate against Microsoft Entra ID and not have to invest in federated AD FS. - Portal UI enables users to easily configure how to map certificate fields to a user object attribute to look up the user in the tenant (certificate username bindings) - Portal UI to configure authentication policies to help determine which certificates are single-factor versus multifactor.
Easy to deploy and administer	<ul style="list-style-type: none"> - Microsoft Entra CBA is a free feature, and you don't need any paid editions of Microsoft Entra ID to use it. - No need for complex on-premises deployments or network configuration. - Directly authenticate against Microsoft Entra ID.
Secure	<ul style="list-style-type: none"> - On-premises passwords don't need to be stored in the cloud in any form. - Protects your user accounts by working seamlessly with Microsoft Entra Conditional Access policies, including Phishing-Resistant multifactor authentication (MFA requires licensed edition) and blocking legacy authentication. - Strong authentication support where users can define authentication policies through the certificate fields, such as issuer or policy OID (object identifiers), to determine which certificates qualify as single-factor versus multifactor. - The feature works seamlessly with Conditional Access features and authentication strength capability to enforce MFA to help secure your users.

Supported scenarios

The following scenarios are supported:

- User sign-ins to web browser-based applications on all platforms.

- User sign-ins to Office mobile apps on iOS/Android platforms as well as Office native apps in Windows, including Outlook, OneDrive, and so on.
- User sign-ins on mobile native browsers.
- Support for granular authentication rules for multifactor authentication by using the certificate issuer **Subject** and **policy OIDs**.
- Configuring certificate-to-user account bindings by using any of the certificate fields:
 - Subject Alternate Name (SAN) PrincipalName and SAN RFC822Name
 - Subject Key Identifier (SKI) and SHA1PublicKey
 - Issuer + Subject, Subject and Issuer + SerialNumber
- Configuring certificate-to-user account bindings by using any of the user object attributes:
 - User Principal Name
 - onPremisesUserPrincipalName
 - CertificateUserIds

Unsupported scenarios

The following scenarios aren't supported:

- Certificate Authority hints aren't supported, so the list of certificates that appears for users in the certificate picker UI isn't scoped.
- Only one CRL Distribution Point (CDP) for a trusted CA is supported.
- The CDP can be only HTTP URLs. We don't support Online Certificate Status Protocol (OCSP), or Lightweight Directory Access Protocol (LDAP) URLs.
- Password as an authentication method can't be disabled and the option to sign in using a password is displayed even with Microsoft Entra CBA method available to the user.

Known Limitation with Windows Hello For Business certificates

- While Windows Hello For Business (WHFB) can be used for multifactor authentication in Microsoft Entra ID, WHFB isn't supported for fresh MFA. Customers can choose to enroll certificates for your users using the WHFB key pair. When properly configured, these WHFB certificates can be used for multifactor authentication in Microsoft Entra ID. WHFB certificates are compatible with Microsoft Entra certificate-based authentication (CBA) in Microsoft Edge and Chrome browsers; however, at this time WHFB certificates aren't compatible with Microsoft Entra CBA in non-browser scenarios (such as Office 365 applications).

The workaround is to use the "Sign in Windows Hello or security key" option to sign in (when available) as this option doesn't use certificates for authentication and avoids the issue with Microsoft Entra CBA; however, this option may not be available in some older applications.

Out of Scope

The following scenarios are out of scope for Microsoft Entra CBA:

- Public Key Infrastructure for creating client certificates. Customers need to configure their own Public Key Infrastructure (PKI) and provision certificates to their users and devices.

Next steps

- [Technical deep dive for Microsoft Entra CBA](#)
- [How to configure Microsoft Entra CBA](#)
- [Microsoft Entra CBA on iOS devices](#)
- [Microsoft Entra CBA on Android devices](#)
- [Windows smart card sign in using Microsoft Entra CBA](#)
- [Certificate user IDs](#)
- [How to migrate federated users](#)
- [FAQ](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Authentication methods in Microsoft Entra ID - OATH tokens

Article • 03/04/2025

OATH time-based one-time password (TOTP) is an open standard that specifies how one-time password (OTP) codes are generated. OATH TOTP can be implemented using either software or hardware to generate the codes. Microsoft Entra ID doesn't support OATH HOTP, a different code generation standard.

Software OATH tokens

Software OATH tokens are typically applications such as the Microsoft Authenticator app and other authenticator apps. Microsoft Entra ID generates the secret key, or seed, that's input into the app and used to generate each OTP.

The Authenticator app automatically generates codes when set up to do push notifications so a user has a backup even if their device doesn't have connectivity. Third-party applications that use OATH TOTP to generate codes can also be used.

Some OATH TOTP hardware tokens are programmable, meaning they don't come with a secret key or seed preprogrammed. These programmable hardware tokens can be set up using the secret key or seed obtained from the software token setup flow. Customers can purchase these tokens from the vendor of their choice and use the secret key or seed in their vendor's setup process.

Hardware OATH tokens (preview)

Microsoft Entra ID supports the use of OATH-TOTP SHA-1 and SHA-256 tokens that refresh codes every 30 or 60 seconds. Customers can purchase these tokens from the vendor of their choice.

Microsoft Entra ID has a new Microsoft Graph API in preview for Azure. Administrators can access Microsoft Graph APIs with least privileged roles to manage tokens in the preview. There aren't any options to manage hardware OATH token in this preview refresh in the Microsoft Entra admin center.

You can continue to manage tokens from the original preview in **OATH tokens** in the Microsoft Entra admin center. On the other hand, you can only manage tokens in the preview refresh by using Microsoft Graph APIs.

Hardware OATH tokens that you add with Microsoft Graph for this preview refresh appear along with other tokens in the admin center. But you can only manage them by using Microsoft Graph.

Time drift correction

Microsoft Entra ID adjusts time drift of the tokens during activation and every authentication. The following table lists the time adjustment that Microsoft Entra ID makes for tokens during activation and sign in.

[\[+\] Expand table](#)

Token refresh interval	Activation time range	Authentication time range
30 seconds	+/- 1 day	+/- 1 minute
60 seconds	+/- 2 days	+/- 2 minutes

Improvements in the preview refresh

This hardware OATH token preview refresh improves flexibility and security for organizations by removing Global Administrator requirements. Organizations can delegate token creation, assignment, and activation to Privileged Authentication Administrators or Authentication Policy Administrators.

The following table lists the role requirements to manage hardware OATH tokens in the preview refresh.

[\[+\] Expand table](#)

Task	Preview refresh role
Create a new token in the tenant's inventory.	Authentication Policy Administrator
Read a token from the tenant's inventory; doesn't return the secret.	Authentication Policy Administrator
Update a token in the tenant. For example, update manufacturer or module; Secret can't be updated.	Authentication Policy Administrator
Delete a token from the tenant's inventory.	Authentication Policy Administrator

As part of the preview refresh, end users can also self-assign and activate tokens from their [Security info](#). In the preview refresh, a token can only be assigned to one user. The following table lists token and role requirements to assign and activate tokens.

[\[+\] Expand table](#)

Task	Token state	Role requirement
Assign a token from the inventory to a user in the tenant.	Assigned	Member (self) Authentication Administrator Privileged Authentication Administrator
Read the token of the user, doesn't return the secret.	Activated / Assigned (depends if the token was already activated or not)	Member (self) Authentication Administrator (only has restricted Read, not standard Read) Privileged Authentication Administrator
Update the token of the user, such as provide current 6-digit code for activation, or change token name.	Activated	Member (self) Authentication Administrator Privileged Authentication Administrator
Remove the token from the user. The token goes back to the token inventory.	Available (back to the tenant inventory)	Member (self) Authentication Administrator Privileged Authentication Administrator

In the legacy multifactor authentication (MFA) policy, hardware and software OATH tokens can only be enabled together. If you enable OATH tokens in the legacy MFA policy, end users see an option to add **Hardware OATH tokens** in their Security info page.

If you don't want end users to see an option to add **Hardware OATH tokens**, migrate to the Authentication methods policy. In the Authentication methods policy, hardware and software OATH tokens can be enabled and managed separately. For more information about how to migrate to the Authentication methods policy, see [How to migrate MFA and SSPR policy settings to the Authentication methods policy for Microsoft Entra ID](#).

Tenants with a Microsoft Entra ID P1 or P2 license can continue to upload hardware OATH tokens as in the original preview. For more information, see [Upload hardware OATH tokens in CSV format](#).

For more information about how to enable hardware OATH tokens and Microsoft Graph APIs that you can use to upload, activate, and assign tokens, see [How to manage OATH](#)

tokens.

OATH token icons

Users can add and manage OATH tokens at [Security info](#), or they can select **Security info** from **My account**. Software and hardware OATH tokens have different icons.

[] Expand table

Token registration type	Icon
OATH software token	
OATH hardware token	

Related content

Learn more about [how to manage OATH tokens](#). Learn about [FIDO2 security key providers](#) that are compatible with passwordless authentication.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Authentication methods in Microsoft Entra ID - phone options

Article • 03/04/2025

Microsoft recommends users move away from using text messages or voice calls for multifactor authentication. Modern authentication methods like [Microsoft Authenticator](#) are a recommended alternative. For more information, see [It's Time to Hang Up on Phone Transports for Authentication](#). Users can still verify themselves using a mobile phone or office phone as secondary form of authentication used for multifactor authentication or self-service password reset (SSPR).

You can [configure and enable users for SMS-based authentication](#) for direct authentication using text message. Text messages are convenient for Frontline workers. With text messages, users don't need to know a username and password to access applications and services. The user instead enters their registered mobile phone number, receives a text message with a verification code, and enters that in the sign-in interface.

ⓘ Note

Phone call verification isn't available for Microsoft Entra tenants with trial subscriptions. For example, if you sign up for a trial license Microsoft Enterprise Mobility and Security (EMS), phone call verification isn't available. Phone numbers must be provided in the format `+CountryCode PhoneNumber`, for example, `+1 4251234567`. There must be a space between the country/region code and the phone number.

Mobile phone verification

For Microsoft Entra multifactor authentication or SSPR, users can choose to receive a text message with a verification code to enter in the sign-in interface, or receive a phone call.

If users don't want their mobile phone number to be visible in the directory but want to use it for password reset, administrators shouldn't populate the phone number in the directory. Instead, users should populate their **Authentication Phone** at [My Sign-Ins](#). Administrators can see this information in the user's profile, but it's not published elsewhere.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below that, the breadcrumb navigation shows 'Home > Contoso | Users > Users > Cameron White'. The main content area is titled 'Cameron White | Authentication methods'. On the left, there's a sidebar with various user management options like 'Overview', 'Audit logs', 'Sign-in logs', etc., and a section for 'Manage' which includes 'Authentication methods' (which is selected and highlighted with a red box). Other sections like 'Troubleshooting + Support' and 'New support request' are also visible. The main content area displays information about authentication methods, including a note about switching back to the old experience, a table for 'Usable authentication methods' (showing 'No usable methods.'), a table for 'Non-usuable authentication methods' (showing 'No non-usuable methods.'), and a table for 'System preferred multifactor authentication method' (showing 'Enabled' and 'No system preferred MFA method').

ⓘ Note

Phone extensions are supported only for office phones.

Microsoft doesn't guarantee consistent text message or voice-based Microsoft Entra multifactor authentication prompt delivery by the same number. In the interest of our users, we may add or remove short codes at any time as we make route adjustments to improve text message deliverability. Microsoft doesn't support short codes for countries/regions besides the United States and Canada.

ⓘ Note

We apply delivery method optimizations such that tenants with a free or trial subscription may receive a text message or voice call.

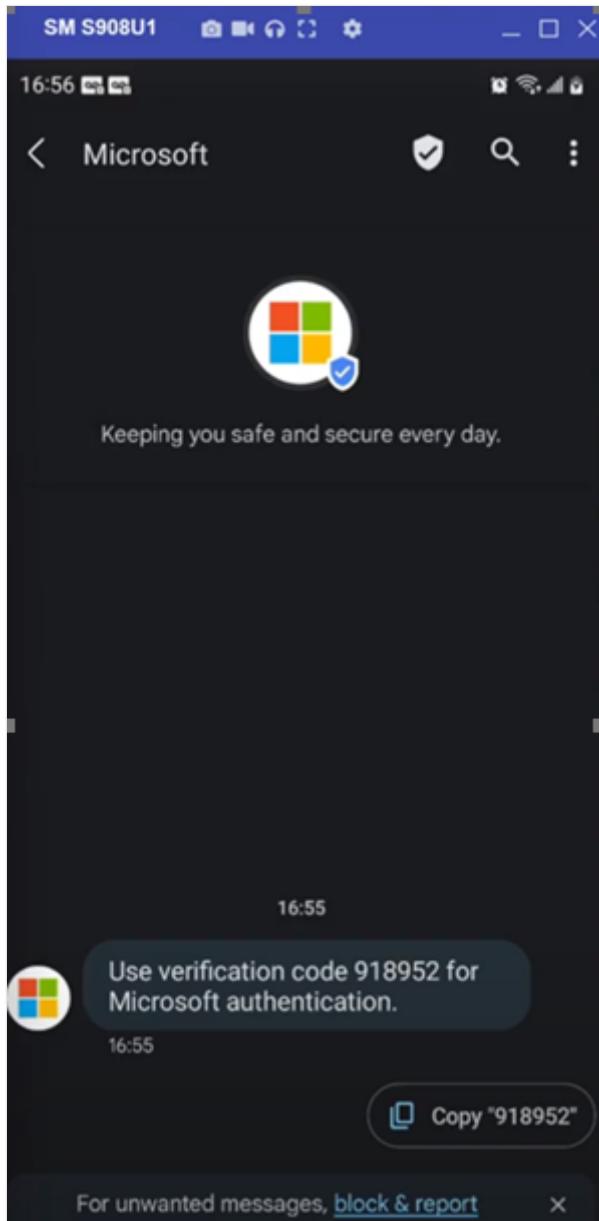
Text message verification

With text message verification during SSPR or Microsoft Entra multifactor authentication, a text message is sent to the mobile phone number containing a verification code. To complete the sign-in process, the verification code provided is entered into the sign-in interface.

Text messages can be sent over channels such as Short Message Service (SMS), Rich Communication Services (RCS), or WhatsApp.

Android users can enable RCS on their devices. RCS offers encryption and other improvements over SMS. For Android, MFA text messages may be sent over RCS rather

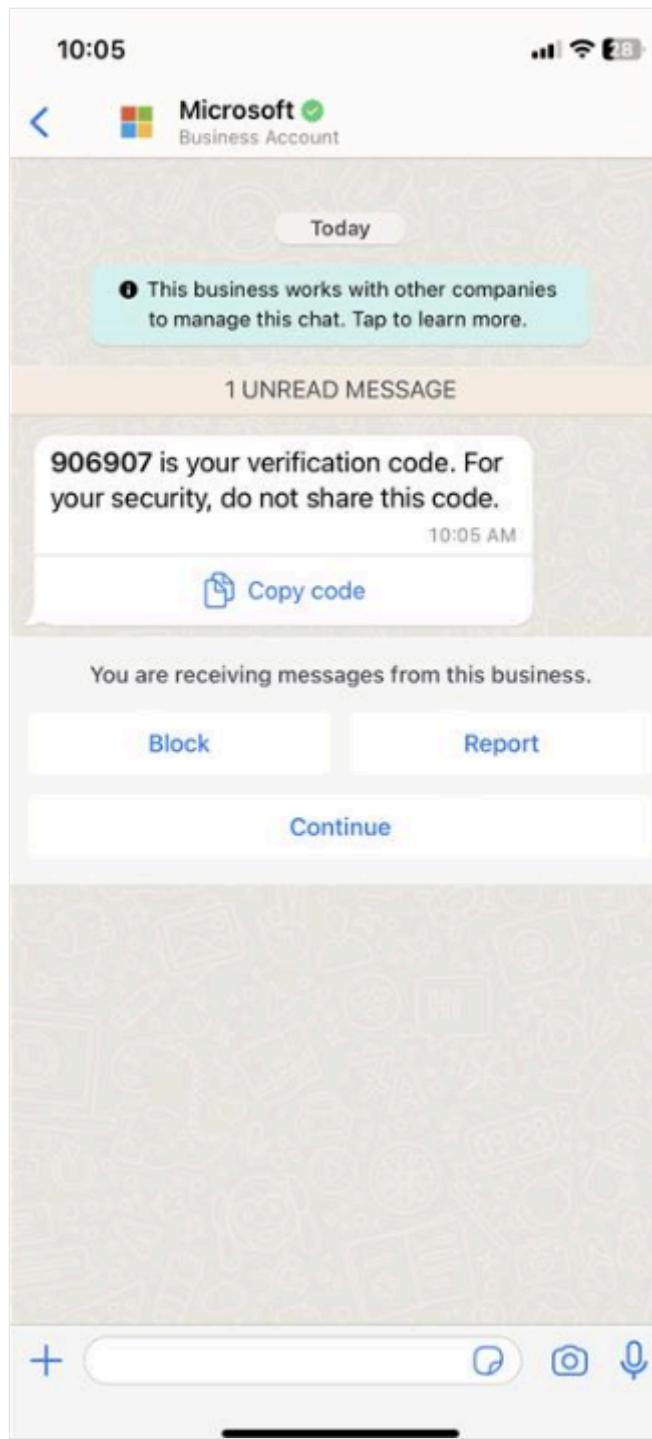
than SMS. The MFA text message is similar to SMS, but RCS messages have more Microsoft branding and a verified checkmark so users know they can trust the message.



Some users may receive their verification codes in WhatsApp. Like RCS, these messages are similar to SMS, but have more Microsoft branding and a verified checkmark. The first time a user receives a verification code in WhatsApp, they're notified by SMS text message of the changed behavior.

Only users that have WhatsApp receive verification codes through this channel. To check if a user has WhatsApp, we silently try to deliver them a message in the app by using the phone number they registered for text message verification.

If users don't have any internet connectivity or they uninstall WhatsApp, they receive SMS verification codes. The phone number associated with Microsoft's WhatsApp Business Agent is: +1 (217) 302 1989.



Phone call verification

With phone call verification during SSPR or Microsoft Entra multifactor authentication, an automated voice call is made to the phone number registered by the user. To complete the sign-in process, the user is prompted to press # on their keypad.

The calling number that a user receives the voice call from differs for each country. See [phone call settings](#) to view all possible voice call numbers.

Note

SSPR can only be completed with a primary phone method or an office phone method. Alternate phone methods are only available for MFA.

Office phone verification

With office phone call verification during SSPR or Microsoft Entra multifactor authentication, an automated voice call is made to the phone number registered by the user. To complete the sign-in process, the user is prompted to press # on their keypad.

Troubleshooting phone options

If you have problems with phone authentication for Microsoft Entra ID, review the following troubleshooting steps:

- "You've hit our limit on verification calls" or "You've hit our limit on text verification codes" error messages during sign-in
 - Microsoft may limit repeated authentication attempts that are performed by the same user or organization in a short period of time. This limitation doesn't apply to Microsoft Authenticator or verification codes. If you have hit these limits, you can use the Authenticator App, verification code or try to sign in again in a few minutes.
- "Sorry, we're having trouble verifying your account" error message during sign-in
 - Microsoft may limit or block voice or text message authentication attempts that are performed by the same user, phone number, or organization due to high number of voice or text message authentication attempts. If you experience this error, you can try another method, such as Authenticator or verification code, or reach out to your admin for support.
- Blocked caller ID on a single device.
 - Review any blocked numbers configured on the device.
- Wrong phone number or incorrect country/region code, or confusion between personal phone number versus work phone number.
 - Troubleshoot the user object and configured authentication methods. Make sure that the correct phone numbers are registered.
- Wrong PIN entered.
 - Confirm the user has used the correct PIN as registered for their account (MFA Server users only).

- Call forwarded to voicemail.
 - Ensure that the user has their phone turned on and that service is available in their area, or use alternate method.
- User is blocked
 - Have a Microsoft Entra administrator unlock the user in the Microsoft Entra admin center.
- Text messaging platforms like SMS, RCS, or WhatsApp aren't subscribed on the device.
 - Have the user change methods or activate a text messaging platform on the device.
- Faulty telecom providers, such as when no phone input is detected, missing DTMF tones issues, blocked caller ID on multiple devices, or blocked text messages across multiple devices.
 - Microsoft uses multiple telecom providers to route phone calls and text messages for authentication. If you see any of these issues, have a user attempt to use the method at least five times within 5 minutes and have that user's information available when contacting Microsoft support.
- Poor signal quality.
 - Have the user attempt to log in using a wi-fi connection by installing the Authenticator app.
 - Or use a text message instead of phone (voice) authentication.
- Phone number is blocked and unable to be used for Voice MFA
 - There are a few country codes blocked for voice MFA unless your Microsoft Entra administrator has opted in for those country codes. Have your Microsoft Entra administrator opt-in to receive MFA for those country codes.
 - Or, use Microsoft Authenticator instead of voice authentication.

Next steps

To get started, see the [tutorial for self-service password reset \(SSPR\)](#) and [Microsoft Entra multifactor authentication](#).

To learn more about SSPR concepts, see [How Microsoft Entra self-service password reset works](#).

To learn more about MFA concepts, see [How Microsoft Entra multifactor authentication works](#).

Learn more about configuring authentication methods using the [Microsoft Graph REST API](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Authentication methods in Microsoft Entra ID - QR code authentication method (Preview)

Article • 05/02/2025

QR code authentication method enables frontline workers to sign in efficiently in apps on shared devices. Users can use a unique QR code provided to them and enter their PIN to sign in, eliminating the need to enter intricate usernames and passwords. Currently, QR code authentication is supported only on mobile devices that run iOS/iPadOS or Android.

What is QR code authentication?

QR code authentication is a simple authentication method primarily designed for frontline workers. It consists of a unique QR code and a numeric PIN. The QR code serves as an identifier and is unique to the user. It can be downloaded and printed by using the Microsoft Entra admin center, My Staff, or Microsoft Graph. For convenience, the QR code can be attached to a badge or any other wearable item.

Authentication Administrators provide a temporary PIN to users, who then change it during sign-in. Only the user knows the PIN. It's exclusively bound to the QR code only. It can't be used with other user identifiers, such as a username or phone number. QR code authentication is a single-factor method in which the PIN (something you know) is a credential.

Benefits of QR code authentication

 Expand table

Benefit	Description
Easier and faster sign-in	Frontline workers don't have to enter complex usernames or passwords to sign in multiple times into shared devices throughout their shift.
Inexpensive	Printing a QR code costs less than a hardware key, which can be cost prohibitive for organizations with temporary frontline workers.

PIN properties

The following policies are applied when an Authentication Policy Administrator creates or resets a PIN.

Policy	Values
Allowed characters	Numbers (0-9)
Unallowed characters	<ul style="list-style-type: none"> - Characters (A-Z, a-z) - Symbols (- @ # \$ % ^ & * - _ ! + = [] { } \ : ' , . ? / ` ~ " () ; < >) - Unicode characters - Blank space
PIN length	8-20 digits
PIN complexity	<p>Enforced to avoid repetition and common sequences. The following patterns are checked:</p> <ul style="list-style-type: none"> - Don't contain 0123456789 or 9876543210. - Don't repeat a sequence of 2-3 digits in the PIN, like 121212, or 123123 or 342342. <p>An Invalid PIN error appears if the PIN includes unallowed characters or is less than the minimum PIN length.</p>

Best security practices to implement with QR code authentication

We recommend the following measures when you enable QR code authentication method as it's a single-factor authentication (something you know).

- QR code authentication is primarily for frontline workers (FLW) and not for information workers (IW). We recommend phishing-resistant authentication or MFA for IW.
- Don't enable QR code authentication for all the users in your tenant. Enable only for target users who will be using this auth method, for example, create a group for frontline workers and enable QR code auth only for them in Microsoft Entra Authentication Methods policies.
- Combine QR code authentication with Conditional Access policies as another security layer. We recommended policies such as compliant devices, access within network, allow for certain applications, and shared device mode.
- Enforce phishing-resistant authentication or MFA when users access resources from outside of the store or workplace network.
- Replace QR codes that are lost or stolen.
- Enforce [sign-in risk based Conditional Access policy](#) to block access.

QR code configurations in the Authentication method policy

Authentication Policy Administrators can enable QR code in Authentication methods in the Microsoft Entra admin center. QR code authentication is disabled by default.

In the Authentication method policy for QR code, you can configure:

- PIN length: 8-20 digits.
- Lifetime of standard QR code: 1-395 days. Default is 365 days. An Authentication Policy Administrator can change the default value when they add a standard QR code for a user.

For example, an admin can set the value to 30 days in the Authentication method policy. For every user in that tenant, the default expiration of a standard QR code is 30 days. An admin can change the default lifetime of the standard QR code for a specific user.

In this screenshot, the PIN length is set to the default of eight digits. The lifetime for the standard QR code is reduced to 200 days.

Home > Authentication methods | Policies >

QR code settings

QR code authentication is a simple and fast authentication method for frontline workers. To use it, you need both a QR code and PIN.

Enable and Target Configure

Defaults

QR PIN Length * ⓘ 8

Lifetime of standard QR code (days) * ⓘ 200

Functional details of QR code authentication method

When an Authentication Policy Administrator adds the QR code authentication method for a user, it generates a standard QR code and PIN. To create a temporary QR code, they need to edit the QR code authentication method.

A temporary QR code helps when a user forgets to bring their badge with standard QR code. It has a shorter lifetime, up to 12 hours. When a QR code authentication method is deleted for the user, they can't sign-in with their existing QR codes and PIN.

A PIN works with both standard and temporary QR codes because PIN is valid for the QR code authentication method. An Authentication Policy Administrator can provide a custom PIN or generate a PIN when they create a QR code authentication method. They can copy a temporary PIN only when they generate it. The PIN is then masked to prevent exposure.

The usability states for a standard QR code, a temporary QR code, and the PIN for a QR code authentication method aren't related to each other. For example, an active QR code authentication method can have a deleted or expired standard QR code, and an active temporary QR code. At any given point of time, there can be only a single active standard QR code and a single active temporary QR code.

The following table lists examples for combinations for the states for a standard QR code, a temporary QR code, and PIN. An active QR code and active PIN are required for successful authentication.

[] [Expand table](#)

Standard QR code	Temporary QR code	PIN for QR code authentication method
Active	Doesn't exist	Temporary, or user updated
Active	Active	Temporary, or user updated
Deleted	Doesn't exist	Temporary, or user updated
Expired	Active	Temporary, or user updated
Expired	Expired	Temporary, or user updated

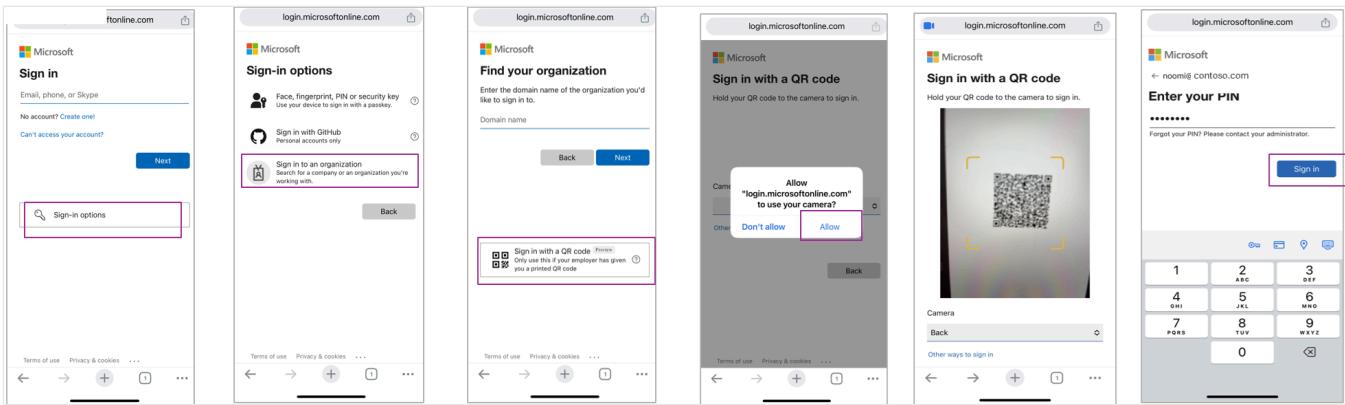
For more information about how to manage QR codes, see [How to enable the QR code authentication method in Microsoft Entra ID \(Preview\)](#).

User sign-in experience with QR code authentication

Users can sign in with a QR code by using the web sign-in experience or an optimized app sign-in experience.

Mobile web sign-in experience

You can use Microsoft's web browser sign-in experience (login.microsoft.com) to authenticate users. Users can click **Sign in options > Sign in to an organization > Sign in with a QR code**.

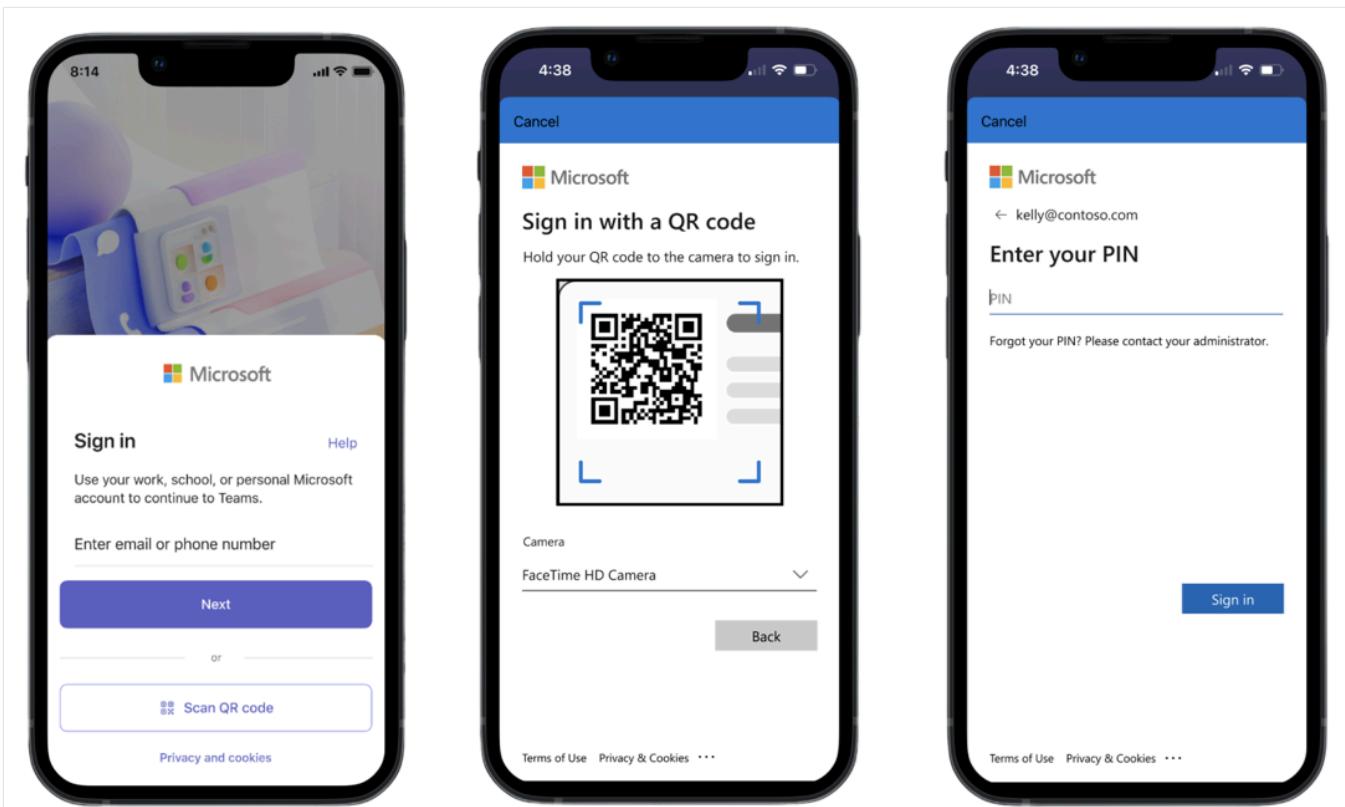


Mobile app sign-in experience

You can optimize sign-in for your apps by using Microsoft Authentication Library (MSAL) to add QR code as an option on the sign-in page. For example, you can add QR code sign-in just like Teams or Managed Home Screen (MHS). Then users can scan the QR code with two fewer clicks. This optimized sign-in experience is available in BlueFletch and Jamf app launchers.

For more information about how to optimize the sign-in experience, see:

- [Set up optimized QR code authentication experience in Android app](#)
- [Set up optimized QR code authentication experience in iOS app](#)





Unsupported user scenarios in current release

- Self-service PIN reset for users
- Bulk provisioning of QR code and PIN
- QR code scan by barcode scanners
- QR code authentication doesn't work with desktop apps or browsers
- Custom tenant endpoint for sign in
- Configurable PIN protection policies that define account lockout threshold, duration, or PIN complexity

Known issue

If you enable QR code authentication for a user, they need to sign-in with an existing authentication method before they can sign in with a QR code for the first time, or they see an **Incorrect QR code** error.

For example:

- You enable QR code authentication for a user.
- The user needs to sign in with their password or another sign-in method.
- For subsequent sign-ins, they can sign in with a QR code.

The user needs to sign in with another method because the cached user authentication method policy isn't updated until the user is authenticated again.

Related content

- [How to enable the QR code authentication method in Microsoft Entra ID \(Preview\)](#)
- [Best practices to protect frontline workers](#)
- [Manage your users with My Staff](#)
- [What authentication and verification methods are available in Microsoft Entra ID?](#)

Configure and enable users for SMS-based authentication using Microsoft Entra ID

Article • 03/04/2025

To simplify and secure sign-in to applications and services, Microsoft Entra ID provides multiple authentication options. SMS-based authentication lets users sign-in without providing, or even knowing, their user name and password. After their account is created by an identity administrator, they can enter their phone number at the sign-in prompt. They receive an SMS authentication code that they can provide to complete the sign-in. This authentication method simplifies access to applications and services, especially for Frontline workers.

This article shows you how to enable SMS-based authentication for select users or groups in Microsoft Entra ID. For a list of apps that support using SMS-based sign-in, see [App support for SMS-based authentication](#).

Before you begin

To complete this article, you need the following resources and privileges:

- An active Azure subscription.
 - If you don't have an Azure subscription, [create an account](#).
- A Microsoft Entra tenant associated with your subscription.
 - If needed, [create a Microsoft Entra tenant](#) or [associate an Azure subscription with your account](#).
- You need at least the [Authentication Policy Administrator](#) role in your Microsoft Entra tenant to enable SMS-based authentication.
- Each user that's enabled in the SMS authentication method policy must be licensed, even if they don't use it. Each enabled user must have one of the following Microsoft Entra ID, EMS, Microsoft 365 licenses:
 - [Microsoft 365 F1 or F3](#)
 - [Microsoft Entra ID P1 or P2](#)
 - [Enterprise Mobility + Security \(EMS\) E3 or E5](#) or [Microsoft 365 E3 or E5](#)
 - [Office 365 F3](#)

Known issues

Here are some known issues:

- SMS-based authentication isn't currently compatible with Microsoft Entra multifactor authentication.
- Except for Teams, SMS-based authentication isn't compatible with native Office applications.
- SMS-based authentication isn't supported for B2B accounts.
- Federated users won't authenticate in the home tenant. They only authenticate in the cloud.
- If a user's default sign-in method is a text or call to your phone number, then the SMS code or voice call is sent automatically during multifactor authentication. As of June 2021, some apps will ask users to choose **Text** or **Call** first. This option prevents sending too many security codes for different apps. If the default sign-in method is the Microsoft Authenticator app ([which we highly recommend](#) ↗), then the app notification is sent automatically.
- [Cross-tenant synchronization](#) does not support users with SMS sign-in enabled.

Enable the SMS-based authentication method

There are three main steps to enable and use SMS-based authentication in your organization:

- Enable the authentication method policy.
- Select users or groups that can use the SMS-based authentication method.
- Assign a phone number for each user account.
 - This phone number can be assigned in the Microsoft Entra admin center (which is shown in this article), and in *My Staff* or *My Account*.

First, let's enable SMS-based authentication for your Microsoft Entra tenant.

1. Sign in to the [Microsoft Entra admin center](#) ↗ as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Policies**.
3. From the list of available authentication methods, select **SMS**.

The screenshot shows the Microsoft Entra ID Security interface. On the left, there's a navigation pane with sections like Manage, Policies, Password protection, Registration campaign, Authentication strengths, Settings, Monitoring, Activity, User registration details, Registration and reset events, and Bulk operation results. The main area has a heading 'Authentication methods | Policies' and a sub-section 'Manage migration'. A note says: 'Use this policy to configure the authentication methods your users may register and use. If a user is in scope for a method, they may use it to authenticate and for password reset (some methods aren't supported for some scenarios). Learn more'. Below this is a table with columns 'Method', 'Target', and 'Enabled'. The methods listed are FIDO2 security key, Microsoft Authenticator, SMS (which is highlighted with a red box), Temporary Access Pass, Hardware OATH tokens (Preview), Third-party software OATH tokens, Voice call, Email OTP (which is marked as Yes), and Certificate-based authentication.

4. Select **Enable** and select **Target users**. You can choose to enable SMS-based authentication for *All users* or *Select users* and groups.

➊ Note

To configure SMS-based authentication for first-factor (that is, to allow users to sign in with this method), check the **Use for sign-in** checkbox. Leaving this unchecked makes SMS-based authentication available for multifactor authentication and Self-Service Password Reset only.

The screenshot shows the 'SMS settings' window. It has a header 'SMS settings' and a note: 'This authentication method delivers a one-time code via SMS to a user's phone, and the user then inputs that code to sign-in. Learn more. SMS is usable for multi-factor authentication and Self-Service Password Reset; it can also be configured to be used as a first factor.' Below this is a section titled 'Enable and Target' with a 'Enable' toggle switch (which is turned on) and a 'Target' section. In the 'Target' section, 'All users' is selected instead of 'Select groups'. There's a table with columns 'Name', 'Type', 'Use for sign-in', and 'Registration'. Under 'Name', 'All users' is listed as a 'Group'. Under 'Use for sign-in', there's a checked checkbox. Under 'Registration', there's a dropdown menu set to 'Optional'.

Assign the authentication method to users and groups

With SMS-based authentication enabled in your Microsoft Entra tenant, now select some users or groups to be allowed to use this authentication method.

1. In the SMS authentication policy window, set **Target** to *Select users*.
2. Choose to **Add users or groups**, then select a test user or group, such as *Contoso User* or *Contoso SMS Users*.

3. When you've selected your users or groups, choose **Select**, then **Save** the updated authentication method policy.

Each user that's enabled in SMS authentication method policy must be licensed, even if they don't use it. Make sure you have the appropriate licenses for the users you enable in the authentication method policy, especially when you enable the feature for large groups of users.

Set a phone number for user accounts

Users are now enabled for SMS-based authentication, but their phone number must be associated with the user profile in Microsoft Entra ID before they can sign-in. The user can [set this phone number themselves](#) in *My Account*, or you can assign the phone number using the Microsoft Entra admin center. Phone numbers can be set by those with at least the [Authentication Administrator](#) role.

When a phone number is set for SMS-based sign-in, it's also then available for use with [Microsoft Entra multifactor authentication](#) and [self-service password reset](#).

1. Search for and select **Microsoft Entra ID**.
2. From the navigation menu on the left-hand side of the Microsoft Entra window, select **Users**.
3. Select the user you enabled for SMS-based authentication in the previous section, such as *Contoso User*, then select **Authentication methods**.
4. Select **+ Add authentication method**, then in the *Choose method* drop-down menu, choose **Phone number**.

Enter the user's phone number, including the country code, such as +1 xxxxxxxxx. The Microsoft Entra admin center validates the phone number is in the correct format.

Then, from the *Phone type* drop-down menu, select *Mobile*, *Alternate mobile*, or *Other* as needed.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various user management options like Profile, Assigned roles, Groups, Applications, Licenses, Devices, and Azure role assignments. The 'Authentication methods' option is selected and highlighted with a red box. On the right, a modal window titled 'Add authentication method' is open. Inside, it says 'Choose method' with a dropdown set to 'Phone number'. Below that, there's a note about adding a phone number for authentication. A form is present with 'Phone number *' containing '+1 4251234567' and 'Phone type' set to 'Mobile'. At the bottom of the modal is a blue 'Add' button.

The phone number must be unique in your tenant. If you try to use the same phone number for multiple users, an error message is shown.

5. To apply the phone number to a user's account, select **Add**.

When successfully provisioned, a check mark appears for *SMS Sign-in enabled*.

Test SMS-based sign-in

To test the user account that's now enabled for SMS-based sign-in, complete the following steps:

1. Open a new InPrivate or Incognito web browser window to <https://www.office.com>
2. In the top right-hand corner, select **Sign in**.
3. At the sign-in prompt, enter the phone number associated with the user in the previous section, then select **Next**.



Sign in

+14251234567

No account? [Create one!](#)

[Can't access your account?](#)

[Sign-in options](#)

[Next](#)

4. An SMS message is sent to the phone number provided. To complete the sign-in process, enter the 6-digit code provided in the SMS message at the sign-in prompt.



← [+1 425-123-4567](#)

Enter code

We just sent a code to [+1 425-123-4567](#)

Enter code

[Sign in](#)

5. The user is now signed in without the need to provide a username or password.

Troubleshoot SMS-based sign-in

You can use the following scenarios and troubleshooting steps if you have problems with enabling and using SMS-based sign-in. For a list of apps that support using SMS-based sign-in, see [App support for SMS-based authentication](#).

Phone number already set for a user account

If a user has already registered for Microsoft Entra multifactor authentication and / or self-service password reset (SSPR), they already have a phone number associated with their account. This phone number isn't automatically available for use with SMS-based sign-in.

A user that has a phone number already set for their account is displayed a button to *Enable for SMS sign-in* in their **My Profile** page. Select this button, and the account is enabled for use with SMS-based sign-in and the previous Microsoft Entra multifactor authentication or SSPR registration.

For more information on the end-user experience, see [SMS sign-in user experience for phone number](#).

Error when trying to set a phone number on a user's account

If you receive an error when you try to set a phone number for a user account in the Microsoft Entra admin center, review the following troubleshooting steps:

1. Make sure that you're enabled for the SMS-based sign-in.
2. Confirm that the user account is enabled in the **SMS** authentication method policy.
3. Make sure you set the phone number with the proper formatting, as validated in the Microsoft Entra admin center (such as +1 4251234567).
4. Make sure that the phone number isn't used elsewhere in your tenant.
5. Check there's no voice number set on the account. If a voice number is set, delete and try to the phone number again.

Next steps

- For a list of apps that support using SMS-based sign-in, see [App support for SMS-based authentication](#).
- For more ways to sign-in to Microsoft Entra ID without a password, such as the Microsoft Authenticator App or FIDO2 security keys, see [Passwordless authentication options for Microsoft Entra ID](#).
- You can also use the Microsoft Graph REST API to [enable](#) or [disable](#) SMS-based sign-in.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Sign-in to Microsoft Entra ID with email as an alternate login ID (Preview)

Article • 04/17/2025

! Note

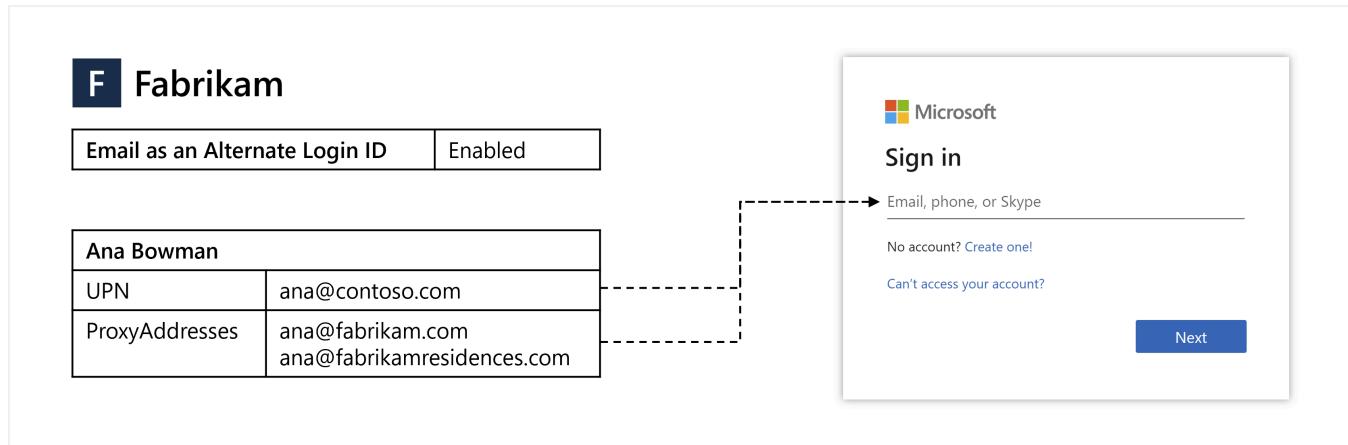
Sign-in to Microsoft Entra ID with email as an alternate login ID is a public preview feature of Microsoft Entra ID. For more information about previews, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

Many organizations want to let users sign in to Microsoft Entra ID using the same credentials as their on-premises directory environment. With this approach, known as hybrid authentication, users only need to remember one set of credentials.

Some organizations haven't moved to hybrid authentication for the following reasons:

- By default, the Microsoft Entra user Principal Name (UPN) is set to the same value as the on-premises UPN.
- Changing the Microsoft Entra UPN creates a mismatch between on-premises and Microsoft Entra environments that could cause problems with certain applications and services.
- Due to business or compliance reasons, the organization doesn't want to use the on-premises UPN to sign in to Microsoft Entra ID.

To move toward hybrid authentication, you can configure Microsoft Entra ID to let users sign in with their email as an alternate login ID. For example, if *Contoso* rebranded to *Fabrikam*, rather than continuing to sign in with the legacy `ana@contoso.com` UPN, email as an alternate login ID can be used. To access an application or service, users would sign in to Microsoft Entra ID using their non-UPN email, such as `ana@fabrikam.com`.



This article shows you how to enable and use email as an alternate login ID.

Before you begin

Here's what you need to know about email as an alternate login ID:

- The feature is available in Microsoft Entra ID Free edition and higher.
- The feature enables sign-in with *ProxyAddresses*, in addition to UPN, for cloud-authenticated Microsoft Entra users. More on how this applies to Microsoft Entra business-to-business (B2B) collaboration in the [B2B](#) section.
- When a user signs in with a non-UPN email, the `unique_name` and `preferred_username` claims (if present) in the [ID token](#) will return the non-UPN email.
 - If the non-UPN email in use becomes stale (no longer belongs to the user), these claims will return the UPN instead.
- The feature supports managed authentication with Password Hash Sync (PHS) or Pass-Through Authentication (PTA).
- There are two options for configuring the feature:
 - [Home Realm Discovery \(HRD\) policy](#) - Use this option to enable the feature for the entire tenant. At least the [Application Administrator](#) role is required.
 - [Staged rollout policy](#) - Use this option to test the feature with specific Microsoft Entra groups. When you first add a security group for staged rollout, you're limited to 200 users to avoid a UX time-out. After you've added the group, you can add more users directly to it, as required.

Preview limitations

In the current preview state, the following limitations apply to email as an alternate login ID:

- **User experience** - Users may see their UPN, even when they signed-in with their non-UPN email. The following example behavior may be seen:
 - User is prompted to sign in with UPN when directed to Microsoft Entra sign-in with `login_hint=<non-UPN_email>`.
 - When a user signs-in with a non-UPN email and enters an incorrect password, the "Enter your password" page changes to display the UPN.
 - On some Microsoft sites and apps, such as Microsoft Office, the *Account Manager* control typically displayed in the upper right may display the user's UPN instead of the non-UPN email used to sign in.
- **Unsupported flows** - Some flows are currently not compatible with non-UPN emails, such as the following:
 - Microsoft Entra ID Protection doesn't match non-UPN emails with *Leaked Credentials* risk detection. This risk detection uses the UPN to match credentials that have been leaked. For more information, see [How To: Investigate risk](#).

- When a user is signed-in with a non-UPN email, they cannot change their password. Microsoft Entra self-service password reset (SSPR) should work as expected. During SSPR, the user may see their UPN if they verify their identity using a non-UPN email.
- **Unsupported scenarios** - The following scenarios are not supported. Sign-in with non-UPN email for:
 - [Microsoft Entra hybrid joined devices](#)
 - [Microsoft Entra joined devices](#)
 - [Microsoft Entra registered devices](#)
 - [Single Sign-On and App Protection Policies on Mobile Platform](#)
 - Legacy authentication such as POP3 and SMTP
- **Unsupported apps** - Some third-party applications may not work as expected if they assume that the `unique_name` or `preferred_username` claims are immutable or will always match a specific user attribute, such as UPN.
- **Logging** - Changes made to the feature's configuration in HRD policy are not explicitly shown in the audit logs.
- **Staged rollout policy** - The following limitations apply only when the feature is enabled using staged rollout policy:
 - The feature does not work as expected for users that are included in other staged rollout policies.
 - Staged rollout policy supports a maximum of 10 groups per feature.
 - Staged rollout policy does not support nested groups.
 - Staged rollout policy does not support dynamic membership groups.
 - Contact objects inside the group will block the group from being added to a staged rollout policy.
- **Duplicate values** - Within a tenant, a cloud-only user's UPN can be the same value as another user's proxy address synced from the on-premises directory. In this scenario, with the feature enabled, the cloud-only user will not be able to sign in with their UPN. More on this issue in the [Troubleshoot](#) section.

Overview of alternate login ID options

To sign in to Microsoft Entra ID, users enter a value that uniquely identifies their account. Historically, you could only use the Microsoft Entra UPN as the sign-in identifier.

For organizations where the on-premises UPN is the user's preferred sign-in email, this approach was great. Those organizations would set the Microsoft Entra UPN to the exact same value as the on-premises UPN, and users would have a consistent sign-in experience.

Alternate Login ID for AD FS

However, in some organizations the on-premises UPN isn't used as a sign-in identifier. In the on-premises environments, you would configure the local AD DS to allow sign-in with an alternate login ID. Setting the Microsoft Entra UPN to the same value as the on-premises UPN isn't an option as Microsoft Entra ID would then require users to sign in with that value.

Alternate Login ID in Microsoft Entra Connect

The typical workaround to this issue was to set the Microsoft Entra UPN to the email address the user expects to sign in with. This approach works, though results in different UPNs between the on-premises AD and Microsoft Entra ID, and this configuration isn't compatible with all Microsoft 365 workloads.

Email as an Alternate Login ID

A different approach is to synchronize the Microsoft Entra ID and on-premises UPNs to the same value and then configure Microsoft Entra ID to allow users to sign in to Microsoft Entra ID with a verified email. To provide this ability, you define one or more email addresses in the user's *ProxyAddresses* attribute in the on-premises directory. *ProxyAddresses* are then synchronized to Microsoft Entra ID automatically using Microsoft Entra Connect.

 Expand table

Option	Description
Alternate Login ID for AD FS	Enable sign-in with an alternate attribute (such as Mail) for AD FS users.
Alternate Login ID in Microsoft Entra Connect	Synchronize an alternate attribute (such as Mail) as the Microsoft Entra UPN.
Email as an Alternate Login ID	Enable sign-in with verified domain <i>ProxyAddresses</i> for Microsoft Entra users.

Synchronize sign-in email addresses to Microsoft Entra ID

Traditional Active Directory Domain Services (AD DS) or Active Directory Federation Services (AD FS) authentication happens directly on your network and is handled by your AD DS infrastructure. With hybrid authentication, users can instead sign in directly to Microsoft Entra ID.

To support this hybrid authentication approach, you synchronize your on-premises AD DS environment to Microsoft Entra ID using [Microsoft Entra Connect](#) and configure it to use PHS or PTA. For more information, see [Choose the right authentication method for your Microsoft Entra hybrid identity solution](#).

In both configuration options, the user submits their username and password to Microsoft Entra ID, which validates the credentials and issues a ticket. When users sign in to Microsoft Entra ID, it removes the need for your organization to host and manage an AD FS infrastructure.

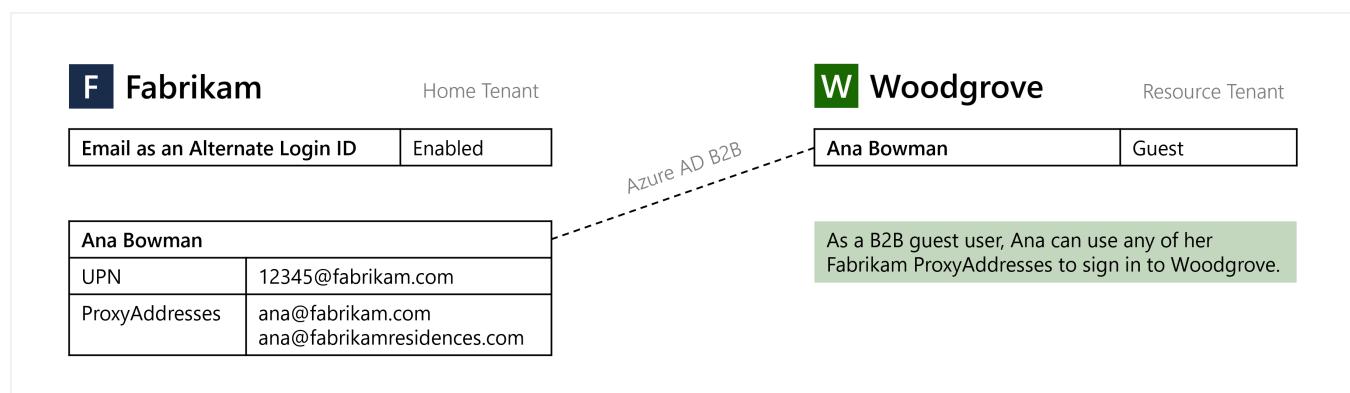
One of the user attributes that's automatically synchronized by Microsoft Entra Connect is *ProxyAddresses*. If users have an email address defined in the on-premises AD DS environment as part of the *ProxyAddresses* attribute, it's automatically synchronized to Microsoft Entra ID. This email address can then be used directly in the Microsoft Entra sign-in process as an alternate login ID.

Important

Only emails in verified domains for the tenant are synchronized to Microsoft Entra ID. Each Microsoft Entra tenant has one or more verified domains, for which you have proven ownership, and are uniquely bound to your tenant.

For more information, see [Add and verify a custom domain name in Microsoft Entra ID](#).

B2B guest user sign-in with an email address



Email as an alternate login ID applies to [Microsoft Entra B2B collaboration](#) under a "bring your own sign-in identifiers" model. When email as an alternate login ID is enabled in the home tenant, Microsoft Entra users can perform guest sign in with non-UPN email on the resource tenant endpoint. No action is required from the resource tenant to enable this functionality.

Note

When an alternate login ID is used on a resource tenant endpoint that does not have the functionality enabled, the sign-in process will work seamlessly, but SSO will be interrupted.

Enable user sign-in with an email address

! Note

This configuration option uses HRD policy. For more information, see [homeRealmDiscoveryPolicy resource type](#).

Once users with the *ProxyAddresses* attribute applied are synchronized to Microsoft Entra ID using Microsoft Entra Connect, you need to enable the feature for users to sign in with email as an alternate login ID for your tenant. This feature tells the Microsoft Entra login servers to not only check the sign-in identifier against UPN values, but also against *ProxyAddresses* values for the email address.

You can use either Microsoft Entra admin center or Graph PowerShell to set up the feature.

Microsoft Entra admin center

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Hybrid Identity Administrator](#).
2. Browse to **Entra ID > Entra Connect > Connect Sync**
3. Select **Email as alternate login ID****.

The screenshot shows the Microsoft Entra Connect | Connect Sync page. The left sidebar has a tree view with 'Microsoft Entra Connect' selected under 'User experiences'. The main content area has a breadcrumb path: Home > Authentication methods > Authentication strengths > Devices > All devices > Microsoft Entra Connect > Cloud Sync > Cloud sync > Configurations > Microsoft Entra Connect. The main content area has several sections: 'USER SIGN-IN' (with 'Email as alternate login ID' checked), 'STAGED ROLLOUT OF CLOUD AUTHENTICATION' (with a note to enable staged rollout for managed user sign-in), 'ON-PREMISES APPLICATIONS' (with a note to configure remote access for on-premises applications), and 'HEALTH AND ANALYTICS' (with a note to monitor on-premises identity infrastructure and synchronization services in the cloud).

4. Click the checkbox next to *Email as an alternate login ID*.

5. Click Save.

The screenshot shows a configuration page for enabling email as an alternate login ID. At the top, there's a breadcrumb trail: Home > Contoso | Microsoft Entra Connect > Microsoft Entra Connect | Connect Sync >. The main title is "Email as an alternate login ID". Below the title, a subtitle reads: "This feature allows cloud-authenticated users to sign in to Microsoft Entra ID with any of their proxy addresses, in addition to UPN." A "Learn more" link is provided. The next section states: "The option below controls the tenant-wide feature setting in Home Realm Discovery policy." Another "Learn more" link is available. A checkbox labeled "Email as an alternate login ID" is checked and highlighted with a red border. At the bottom, there are "Save" and "Cancel" buttons, with "Save" also highlighted with a red border.

With the policy applied, it can take up to one hour to propagate and for users to be able to sign in using their alternate login ID.

PowerShell

Note

This configuration option uses HRD policy. For more information, see [homeRealmDiscoveryPolicy resource type](#).

Once users with the *ProxyAddresses* attribute applied are synchronized to Microsoft Entra ID using Microsoft Entra Connect, you need to enable the feature for users to sign-in with email as an alternate login ID for your tenant. This feature tells the Microsoft Entra login servers to not only check the sign-in identifier against UPN values, but also against *ProxyAddresses* values for the email address.

1. Open a PowerShell session as an administrator, then install the *Microsoft.Graph* module using the `Install-Module` cmdlet:

```
PowerShell  
Install-Module Microsoft.Graph
```

For more information on installation, see [Install the Microsoft Graph PowerShell SDK](#).

2. Sign-in to your Microsoft Entra tenant using the `Connect-MgGraph` cmdlet:

```
PowerShell  
Connect-MgGraph -Scopes "Policy.ReadWrite.ApplicationConfiguration" -TenantId organizations
```

The command will ask you to authenticate using a web browser.

3. Check if a *HomeRealmDiscoveryPolicy* already exists in your tenant using the `Get-MgPolicyHomeRealmDiscoveryPolicy` cmdlet as follows:

```
PowerShell  
Get-MgPolicyHomeRealmDiscoveryPolicy
```

4. If there's no policy currently configured, the command returns nothing. If a policy is returned, skip this step and move on to the next step to update an existing policy.

To add the *HomeRealmDiscoveryPolicy* to the tenant, use the `New-MgPolicyHomeRealmDiscoveryPolicy` cmdlet and set the *AlternateIdLogin* attribute to "*Enabled*": *true* as shown in the following example:

```
PowerShell
```

```

$AzureADPolicyDefinition = @(
    @{
        "HomeRealmDiscoveryPolicy" = @{
            "AlternateIdLogin" = @{
                "Enabled" = $true
            }
        }
    } | ConvertTo-JSON -Compress
)

$AzureADPolicyParameters = @{
    Definition          = $AzureADPolicyDefinition
    DisplayName         = "BasicAutoAccelerationPolicy"
    AdditionalProperties = @{
        IsOrganizationDefault = $true
    }
}

New-MgPolicyHomeRealmDiscoveryPolicy @AzureADPolicyParameters

```

When the policy has been successfully created, the command returns the policy ID, as shown in the following example output:

PowerShell
<pre> Definition DeletedDateTime Description DisplayName Id IsOrganizationDefault ----- ----- ----- {{"HomeRealmDiscoveryPolicy": {"AlternateIdLogin": {"Enabled": true}}}} BasicAutoAccelerationPolicy HRD_POLICY_ID True </pre>

5. If there's already a configured policy, check if the *AlternateIdLogin* attribute is enabled, as shown in the following example policy output:

PowerShell
<pre> Definition DeletedDateTime Description DisplayName Id IsOrganizationDefault ----- ----- ----- {{"HomeRealmDiscoveryPolicy": {"AlternateIdLogin": {"Enabled": true}}}} BasicAutoAccelerationPolicy HRD_POLICY_ID True </pre>

If the policy exists but the *AlternateIdLogin* attribute that isn't present or enabled, or if other attributes exist on the policy you wish to preserve, update the existing policy using

the `Update-MgPolicyHomeRealmDiscoveryPolicy` cmdlet.

ⓘ Important

When you update the policy, make sure you include any old settings and the new *AlternateIdLogin* attribute.

The following example adds the *AlternateIdLogin* attribute and preserves the *AllowCloudPasswordValidation* attribute that was previously set:

PowerShell

```
$AzureADPolicyDefinition = @(
    @{
        "HomeRealmDiscoveryPolicy" = @{
            "AllowCloudPasswordValidation" = $true
            "AlternateIdLogin" = @{
                "Enabled" = $true
            }
        }
    } | ConvertTo-JSON -Compress
)

$AzureADPolicyParameters = @{
    HomeRealmDiscoveryPolicyId = "HRD_POLICY_ID"
    Definition                 = $AzureADPolicyDefinition
    DisplayName                = "BasicAutoAccelerationPolicy"
    AdditionalProperties       = @{
        "IsOrganizationDefault" = $true
    }
}

Update-MgPolicyHomeRealmDiscoveryPolicy @AzureADPolicyParameters
```

Confirm that the updated policy shows your changes and that the *AlternateIdLogin* attribute is now enabled:

PowerShell

```
Get-MgPolicyHomeRealmDiscoveryPolicy
```

ⓘ Note

With the policy applied, it can take up to an hour to propagate and for users to be able to sign-in using email as an alternate login ID.

Removing policies

To remove an HRD policy, use the `Remove-MgPolicyHomeRealmDiscoveryPolicy` cmdlet:

PowerShell

```
Remove-MgPolicyHomeRealmDiscoveryPolicy -HomeRealmDiscoveryPolicyId  
"HRD_POLICY_ID"
```

Enable staged rollout to test user sign-in with an email address

! Note

This configuration option uses staged rollout policy. For more information, see [featureRolloutPolicy resource type](#).

Staged rollout policy allows tenant administrators to enable features for specific Microsoft Entra groups. It is recommended that tenant administrators use staged rollout to test user sign-in with an email address. When administrators are ready to deploy this feature to their entire tenant, they should use [HRD policy](#).

1. Open a PowerShell session as an administrator, then install the *Microsoft.Graph.Beta* module using the `Install-Module` cmdlet:

PowerShell

```
Install-Module Microsoft.Graph.Beta
```

If prompted, select Y to install NuGet or to install from an untrusted repository.

2. Sign in to your Microsoft Entra tenant using the `Connect-MgGraph` cmdlet:

PowerShell

```
Connect-MgGraph -Scopes "Directory.ReadWrite.All"
```

The command returns information about your account, environment, and tenant ID.

3. List all existing staged rollout policies using the following cmdlet:

PowerShell

Get-MgBetaPolicyFeatureRolloutPolicy

4. If there are no existing staged rollout policies for this feature, create a new staged rollout policy and take note of the policy ID:

PowerShell

```
$MgPolicyFeatureRolloutPolicy = @{
    Feature      = "EmailAsAlternateId"
    DisplayName  = "EmailAsAlternateId Rollout Policy"
    IsEnabled    = $true
}
New-MgBetaPolicyFeatureRolloutPolicy @MgPolicyFeatureRolloutPolicy
```

5. Find the directoryObject ID for the group to be added to the staged rollout policy. Note the value returned for the *Id* parameter, because it will be used in the next step.

PowerShell

```
Get-MgBetaGroup -Filter "DisplayName eq 'Name of group to be added to the
staged rollout policy'"
```

6. Add the group to the staged rollout policy as shown in the following example. Replace the value in the *-FeatureRolloutPolicyId* parameter with the value returned for the policy ID in step 4 and replace the value in the *-OdataId* parameter with the *Id* noted in step 5. It may take up to 1 hour before users in the group can sign in to Microsoft Entra ID with email as an alternate login ID.

PowerShell

```
New-MgBetaDirectoryFeatureRolloutPolicyApplyToByRef ` 
    -FeatureRolloutPolicyId "ROLLOUT_POLICY_ID" ` 
    -OdataId
"https://graph.microsoft.com/v1.0/directoryObjects/{GROUP_OBJECT_ID}"
```

For new members added to the group, it may take up to 24 hours before they can sign in to Microsoft Entra ID with email as an alternate login ID.

Removing groups

To remove a group from a staged rollout policy, run the following command:

PowerShell

```
Remove-MgBetaPolicyFeatureRolloutPolicyApplyToByRef -FeatureRolloutPolicyId  
"ROLLOUT_POLICY_ID" -DirectoryObjectId "GROUP_OBJECT_ID"
```

Removing policies

To remove a staged rollout policy, first disable the policy then remove it from the system:

PowerShell

```
Update-MgBetaPolicyFeatureRolloutPolicy -FeatureRolloutPolicyId  
"ROLLOUT_POLICY_ID" -IsEnabled:$false  
Remove-MgBetaPolicyFeatureRolloutPolicy -FeatureRolloutPolicyId  
"ROLLOUT_POLICY_ID"
```

Test user sign-in with an email address

To test that users can sign in with email, go to <https://myprofile.microsoft.com> and sign in with a non-UPN email, such as `balas@fabrikam.com`. The sign-in experience should look and feel the same as signing-in with the UPN.

Troubleshoot

If users have trouble signing in with their email address, review the following troubleshooting steps:

1. Make sure it's been at least 1 hour since email as an alternate login ID was enabled. If the user was recently added to a group for staged rollout policy, make sure it's been at least 24 hours since they were added to the group.
2. If using HRD policy, confirm that the Microsoft Entra ID *HomeRealmDiscoveryPolicy* has the *AlternateIdLogin* definition property set to "*Enabled*": *true* and the *IsOrganizationDefault* property set to *True*:

PowerShell

```
Get-MgBetaPolicyHomeRealmDiscoveryPolicy | Format-List *
```

If using staged rollout policy, confirm that the Microsoft Entra ID *FeatureRolloutPolicy* has the *IsEnabled* property set to *True*:

PowerShell

[Get-MgBetaPolicyFeatureRolloutPolicy](#)

3. Make sure the user account has their email address set in the *ProxyAddresses* attribute in Microsoft Entra ID.

Sign-in logs

Activity Details: Sign-ins X

[Basic info](#) [Location](#) [Device info](#) [Authentication Details](#) [Conditional Access](#) [Report-only](#) [...](#)

Date
Request ID
Correlation ID
Authentication requirement
Status
Continuous access evaluation
Additional Details

Troubleshoot Event

User
Username
User ID
[Sign-in identifier](#)
User type
[Sign-in identifier type](#) proxyAddress
Cross tenant access type
Application

You can review the [sign-in logs in Microsoft Entra ID](#) for more information. Sign-ins with email as an alternate login ID will emit `proxyAddress` in the *Sign-in identifier type* field and the inputted username in the *Sign-in identifier* field.

Conflicting values between cloud-only and synced users

Within a tenant, a cloud-only user's UPN may take on the same value as another user's proxy address synced from the on-premises directory. In this scenario, with the feature enabled, the

cloud-only user will not be able to sign in with their UPN. Here are the steps for detecting instances of this issue.

1. Open a PowerShell session as an administrator, then install Microsoft Graph by using the [Install-Module](#) cmdlet:

```
PowerShell  
  
Install-Module Microsoft.Graph.Authentication
```

If prompted, select Y to install NuGet or to install from an untrusted repository.

2. Connect to Microsoft Graph:

```
PowerShell  
  
Connect-MgGraph -Scopes "User.Read.All"
```

3. Get affected users.

```
PowerShell  
  
# Get all users  
$allUsers = Get-MgUser -All  
  
# Get list of proxy addresses from all synced users  
$syncedProxyAddresses = $allUsers |  
    Where-Object {$_['ImmutableId']} |  
    Select-Object -ExpandProperty ProxyAddresses |  
    ForEach-Object {$_.Replace "smtp:", ""}  
  
# Get list of user principal names from all cloud-only users  
$cloudOnlyUserPrincipalNames = $allUsers |  
    Where-Object {!$_['ImmutableId']} |  
    Select-Object -ExpandProperty UserPrincipalName  
  
# Get intersection of two lists  
$duplicateValues = $syncedProxyAddresses |  
    Where-Object {$cloudOnlyUserPrincipalNames -Contains $_}
```

4. To output affected users:

```
PowerShell  
  
# Output affected synced users  
$allUsers |  
    Where-Object {$_['ImmutableId'] -And ($_.ProxyAddresses | Where-Object  
    {($duplicateValues | ForEach-Object {"smtp:$_"}) -Contains $_}).Length -GT 0}
```

```
|  
    Select-Object ObjectId, DisplayName, UserPrincipalName, ProxyAddresses,  
    ImmutableId, UserType  
  
# Output affected cloud-only users  
$allUsers |  
    Where-Object {!$_.ImmutableId -And $duplicateValues -Contains  
    $_.UserPrincipalName} |  
    Select-Object ObjectId, DisplayName, UserPrincipalName, ProxyAddresses,  
    ImmutableId, UserType
```

5. To output affected users to CSV:

PowerShell

```
# Output affected users to CSV  
$allUsers |  
    Where-Object {  
        ($_.ImmutableId -And ($_.ProxyAddresses | Where-Object  
        {($duplicateValues | ForEach-Object {"smtp:$_"}) -Contains $_}).Length -GT 0)  
        -Or  
        (!$.ImmutableId -And $duplicateValues -Contains  
        $_.UserPrincipalName)  
    } |  
    Select-Object ObjectId, DisplayName, UserPrincipalName,  
    @{n="ProxyAddresses"; e={$_.ProxyAddresses -Join ','}}, @{n="IsSyncedUser";  
e={$_.ImmutableId.Length -GT 0}}, UserType |  
    Export-Csv -Path .\AffectedUsers.csv -NoTypeInformation
```

Next steps

To learn more about hybrid identity, such as Microsoft Entra application proxy or Microsoft Entra Domain Services, see [Microsoft Entra hybrid identity for access and management of on-prem workloads](#).

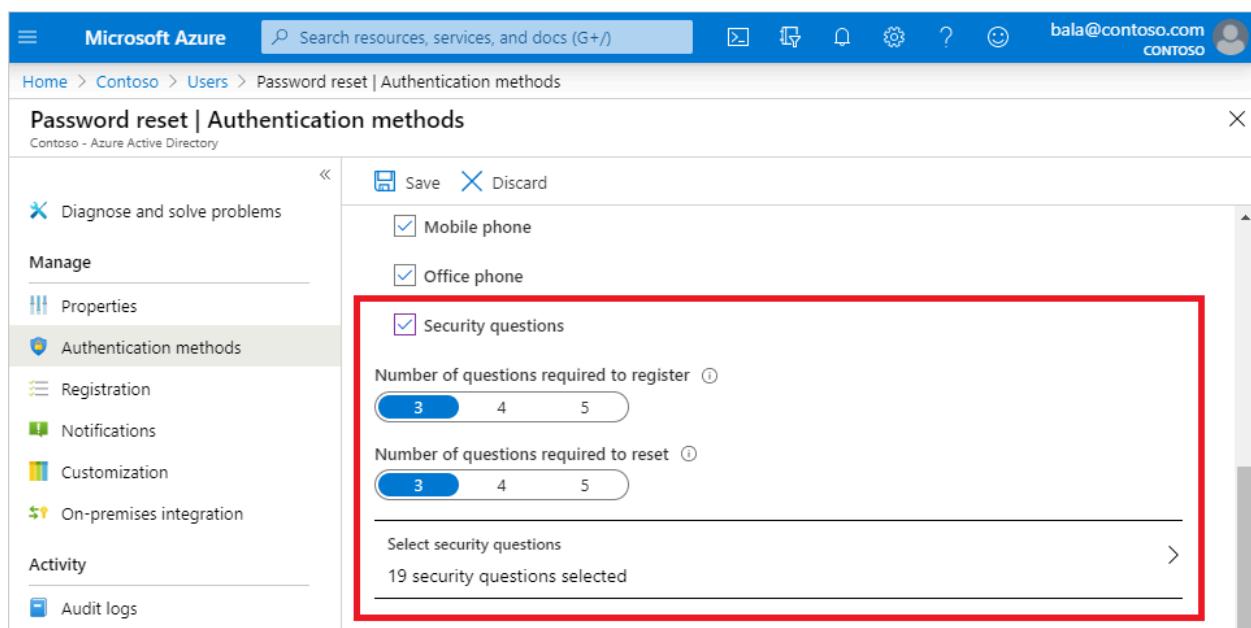
For more information on hybrid identity operations, see [how password hash sync or pass-through authentication synchronization work](#).

Authentication methods in Microsoft Entra ID - security questions

Article • 03/04/2025

Security questions aren't used as an authentication method during a sign-in event. Instead, security questions can be used during the self-service password reset (SSPR) process to confirm who you are. Administrator accounts can't use security questions as verification method with SSPR.

When users register for SSPR, they're prompted to choose the authentication methods to use. If they choose to use security questions, they pick from a set of questions to prompt for and then provide their own answers.



The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes 'Microsoft Azure', a search bar, and a user profile for 'bala@contoso.com'. Below the navigation is a breadcrumb trail: Home > Contoso > Users > Password reset | Authentication methods. The main content area is titled 'Password reset | Authentication methods' under 'Contoso - Azure Active Directory'. On the left, there's a sidebar with 'Manage' sections like 'Properties', 'Authentication methods' (which is selected and highlighted in grey), 'Registration', 'Notifications', 'Customization', 'On-premises integration', 'Activity', and 'Audit logs'. The main pane shows 'Save' and 'Discard' buttons. Under 'Authentication methods', there are checkboxes for 'Mobile phone', 'Office phone', and 'Security questions' (which is checked). A red box highlights the 'Security questions' section. It contains two dropdown menus for 'Number of questions required to register' (set to 3) and 'Number of questions required to reset' (set to 3). Below these is a link 'Select security questions' which shows '19 security questions selected'. At the bottom right of the main pane is a red 'X' button.

ⓘ Note

Security questions are stored privately and securely on a user object in the directory and can only be answered by users during registration. There's no way for an administrator to read or modify a user's questions or answers.

Security questions can be less secure than other methods because some people might know the answers to another user's questions. If you use security questions with SSPR, it's recommended to use them in along with another method. A user can be prompted to use the Microsoft Authenticator App or phone authentication to verify their identity during the SSPR process, and choose security questions only if they don't have their phone or registered device with them.

Predefined questions

The following predefined security questions are available for use as a verification method with SSPR. All of these security questions are translated and localized into the full set of Microsoft 365 languages based on the user's browser locale:

- In what city did you meet your first spouse/partner?
- In what city did your parents meet?
- In what city does your nearest sibling live?
- In what city was your father born?
- In what city was your first job?
- In what city was your mother born?
- What city were you in on New Year's 2000?
- What is the last name of your favorite teacher in high school?
- What is the name of a college you applied to but didn't attend?
- What is the name of the place in which you held your first wedding reception?
- What is your father's middle name?
- What is your favorite food?
- What is your maternal grandmother's first and last name?
- What is your mother's middle name?
- What is your oldest sibling's birthday month and year? (for example, November 1985)
- What is your oldest sibling's middle name?
- What is your paternal grandfather's first and last name?
- What is your youngest sibling's middle name?
- What school did you attend for sixth grade?
- What was the first and last name of your childhood best friend?
- What was the first and last name of your first significant other?
- What was the last name of your favorite grade school teacher?
- What was the make and model of your first car or motorcycle?
- What was the name of the first school you attended?
- What was the name of the hospital in which you were born?
- What was the name of the street of your first childhood home?
- What was the name of your childhood hero?
- What was the name of your favorite stuffed animal?
- What was the name of your first pet?
- What was your childhood nickname?
- What was your favorite sport in high school?
- What was your first job?
- What were the last four digits of your childhood telephone number?
- When you were young, what did you want to be when you grew up?

- Who is the most famous person you have ever met?

Custom security questions

For additional flexibility, you can define your own custom security questions. The maximum length of a custom security question is 200 characters.

Custom security questions aren't automatically localized like with the default security questions. All custom questions are displayed in the same language as they're entered in the administrative user interface, even if the user's browser locale is different. If you need localized questions, you should use the predefined questions.

Security question requirements

For both default and custom security questions, the following requirements and limitations apply:

- The minimum answer character limit is three characters.
- The maximum answer character limit is 40 characters.
- Users can't answer the same question more than one time.
- Users can't provide the same answer to more than one question.
- Any character set can be used to define the questions and the answers, including Unicode characters.
- The number of questions defined must be greater than or equal to the number of questions that were required to register.

Next steps

To get started, see the [tutorial for self-service password reset \(SSPR\)](#).

To learn more about SSPR concepts, see [How Microsoft Entra self-service password reset works](#).

Learn more about configuring authentication methods using the [Microsoft Graph REST API](#).

Feedback

Was this page helpful?



Provide product feedback ↗

Conditional Access authentication strength

Article • 03/04/2025

Authentication strength is a Conditional Access control that specifies which combinations of authentication methods can be used to access a resource. Users can satisfy the strength requirements by authenticating with any of the allowed combinations.

For example, an authentication strength can require that only phishing-resistant authentication methods be used to access a sensitive resource. To access a nonsensitive resource, administrators can create another authentication strength that allows less secure multifactor authentication (MFA) combinations, such as password + text message.

Authentication strength is based on the [Authentication methods policy](#), where administrators can scope authentication methods for specific users and groups to be used across Microsoft Entra ID federated applications. Authentication strength allows further control over the usage of these methods based upon specific scenarios such as sensitive resource access, user risk, location, and more.

Scenarios for authentication strengths

Authentication strengths can help customers address these scenarios:

- Require specific authentication methods to access a sensitive resource.
- Require a specific authentication method when a user takes a sensitive action within an application (in combination with Conditional Access authentication context).
- Require users to use a specific authentication method when they access sensitive applications outside of the corporate network.
- Require more secure authentication methods for users at high risk.
- Require specific authentication methods from guest users who access a resource tenant (in combination with cross-tenant settings).

Authentication strengths

Administrators can specify an authentication strength to access a resource by creating a Conditional Access policy with the **Require authentication strength** control. They can

choose from three built-in authentication strengths: **Multifactor authentication strength**, **Passwordless MFA strength**, and **Phishing-resistant MFA strength**. They can also create a custom authentication strength based on the authentication method combinations they want to allow.

The screenshot shows the 'Grant' configuration dialog box within the Microsoft Conditional Access interface. The left pane displays the policy settings:

- Name:** Phish resistant MFA required
- Assignments:** 0 users and groups selected
- Target resources:** No target resources selected
- Conditions:** 0 conditions selected
- Access controls:** 0 controls selected
- Session:** 0 controls selected
- Enable policy:** Report-only (selected), On, Off

The right pane shows the 'Grant' configuration:

- Control access enforcement:** Grant access (selected)
- Require multifactor authentication:** Unselected
- Require authentication strength:** Selected, set to Phishing-resistant MFA
- Other options:** Unselected (Require device to be marked as compliant, Require Microsoft Entra hybrid joined device, Require approved client app, Require app protection policy)

A warning message is displayed: "⚠️ 'Require authentication strength' cannot be used with 'Require multifactor authentication'. [Learn more](#)".

Built-in authentication strengths

Built-in authentication strengths are combinations of authentication methods that are predefined by Microsoft. Built-in authentication strengths are always available and can't be modified. Microsoft will update built-in authentication strengths when new methods become available.

For an example, the built-in **Phishing-resistant MFA strength** allows the following combinations:

- Windows Hello for Business

Or

- FIDO2 security key

Or

- Microsoft Entra certificate-based authentication (Multifactor)

The screenshot shows the Microsoft Entra ID Security portal. On the left, there's a navigation sidebar with options like Policies, Password protection, Registration campaign, Authentication strengths (which is selected), Settings, Monitoring, Activity, User registration details, Registration and reset events, and Bulk operation results. The main area is titled "Authentication methods | Authentication strengths" and shows a table of authentication strengths. The table has columns for "Authentication strength", "Type", and "Authentication methods". It lists three rows: "Multifactor authentication" (Built-in, Windows Hello For Business and 16 more), "Passwordless MFA" (Built-in, Windows Hello For Business and 3 more), and "Phishing-resistant MFA" (Built-in, Windows Hello For Business and 2 more). Above the table, there are buttons for "Search", "New authentication strength", "Refresh", "Type: All", "Authentication methods: All", and "Reset filters". A tooltip for "Learn more" is also visible. To the right, a modal window titled "View Authentication Strength" provides detailed information about each strength. For "Multifactor authentication", it says "Name: Phishing-resistant MFA", "Type: Built-in", and "Description: Include authentication methods that are phishing-resistant like Passkeys (FIDO2) and Windows Hello for Business". For "Passwordless MFA", it says "Name: Passkeys (FIDO2)", "Type: Built-in", and "Description: OR Certificate-based Authentication (Multifactor)". For "Phishing-resistant MFA", it says "Name: Windows Hello For Business", "Type: Built-in", and "Description: OR Windows Hello For Business".

The combinations of authentication methods for each built-in authentication strength are listed in the following table. These combinations include methods that need to be registered by users and enabled in the Authentication methods policy or the legacy MFA settings policy.

- **MFA strength** - the same set of combinations that could be used to satisfy the **Require multifactor authentication** setting.
- **Passwordless MFA strength** - includes authentication methods that satisfy MFA but don't require a password.
- **Phishing-resistant MFA strength** - includes methods that require an interaction between the authentication method and the sign-in surface.

[] Expand table

Authentication method combination	MFA strength	Passwordless MFA strength	Phishing-resistant MFA strength
FIDO2 security key	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Windows Hello for Business	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Certificate-based authentication (Multi-Factor)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Microsoft Authenticator (Phone Sign-in)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Temporary Access Pass (One-time use AND Multi-use)	<input checked="" type="checkbox"/>		
Password + something you have ¹	<input checked="" type="checkbox"/>		

Authentication method combination	MFA strength	Passwordless MFA strength	Phishing-resistant MFA strength
Federated single-factor + something you have ¹	<input checked="" type="checkbox"/>		
Federated Multi-Factor	<input checked="" type="checkbox"/>		
Certificate-based authentication (single-factor)			
SMS sign-in			
Password			
Federated single-factor			

¹ Something you have refers to one of the following methods: text message, voice, push notification, software OATH token, or hardware OATH token.

The following API call can be used to list definitions of all the built-in authentication strengths:

HTTP

GET

```
https://graph.microsoft.com/beta/identity/conditionalAccess/authenticationStrength/policies?$filter=policyType eq 'builtIn'
```

Conditional Access Administrators can also create custom authentication strengths to exactly suit their access requirements. For more information, see [Custom Conditional Access authentication strengths](#).

Limitations

- **Conditional Access policies are only evaluated after the initial authentication -** As a result, authentication strength doesn't restrict a user's initial authentication. Suppose you are using the built-in phishing-resistant MFA strength. A user can still type in their password, but they are required to sign in with a phishing-resistant method such as FIDO2 security key before they can continue.
- **Require multifactor authentication and Require authentication strength can't be used together in the same Conditional Access policy -** These two Conditional Access grant controls can't be used together because the built-in authentication

strength **Multifactor authentication** is equivalent to the **Require multifactor authentication** grant control.

- **Authentication methods that aren't currently supported by authentication strength** - The **Email one-time pass (Guest)** authentication method isn't included in the available combinations.
- **Windows Hello for Business** – If the user signed in with Windows Hello for Business as their primary authentication method, it can be used to satisfy an authentication strength requirement that includes Windows Hello for Business. However, if the user signed in with another method like password as their primary authentication method, and the authentication strength requires Windows Hello for Business, they aren't prompted to sign in with Windows Hello for Business. The user needs to restart the session, choose **Sign-in options**, and select a method required by the authentication strength.

Known issues

- **Authentication strength and sign-in frequency** - When a resource requires an authentication strength and a sign-in frequency, users can satisfy both requirements at two different times.

For example, let's say a resource requires passkey (FIDO2) for the authentication strength, and a 1-hour sign-in frequency. 24 hours ago, a user signed in with passkey (FIDO2) to access the resource.

When the user unlocks their Windows device using Windows Hello for Business, they can access the resource again. Yesterday's sign-in satisfies the authentication strength requirement, and today's device unlock satisfies the sign-in frequency requirement.

- **Authentication strength blade double representation** - Platform credentials, such as Windows Hello for Business and **Platform Credential for macOS** are both represented in authentication strength under **Windows Hello For Business**. To configure a custom authentication strength that allows the use of **Platform Credential for macOS**, use **Windows Hello For Business**.

FAQ

Should I use authentication strength or the Authentication methods policy?

Authentication strength is based on the Authentication methods policy. The Authentication methods policy helps to scope and configure authentication methods to be used across Microsoft Entra ID by specific users and groups. Authentication strength allows another restriction of methods for specific scenarios, such as sensitive resource access, user risk, location, and more.

For example, the administrator of Contoso wants to allow their users to use Microsoft Authenticator with either push notifications or passwordless authentication mode. The administrator goes to the Microsoft Authenticator settings in the Authentication methods policy, scopes the policy for the relevant users, and sets the **Authentication mode** to Any.

Then for Contoso's most sensitive resource, the administrator wants to restrict the access to only passwordless authentication methods. The administrator creates a new Conditional Access policy, using the built-in **Passwordless MFA strength**.

As a result, users in Contoso can access most of the resources in the tenant using password + push notification from the Microsoft Authenticator OR only using Microsoft Authenticator (phone sign-in). However, when the users in the tenant access the sensitive application, they must use Microsoft Authenticator (phone sign-in).

Prerequisites

- **Microsoft Entra ID P1** - Your tenant needs to have Microsoft Entra ID P1 license to use Conditional Access. If needed, you can enable a [free trial](#).

Next steps

- [Custom Conditional Access authentication strengths](#)
- [How authentication strength works for external users](#)
- [Troubleshoot authentication strengths](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

How Conditional Access authentication strength works

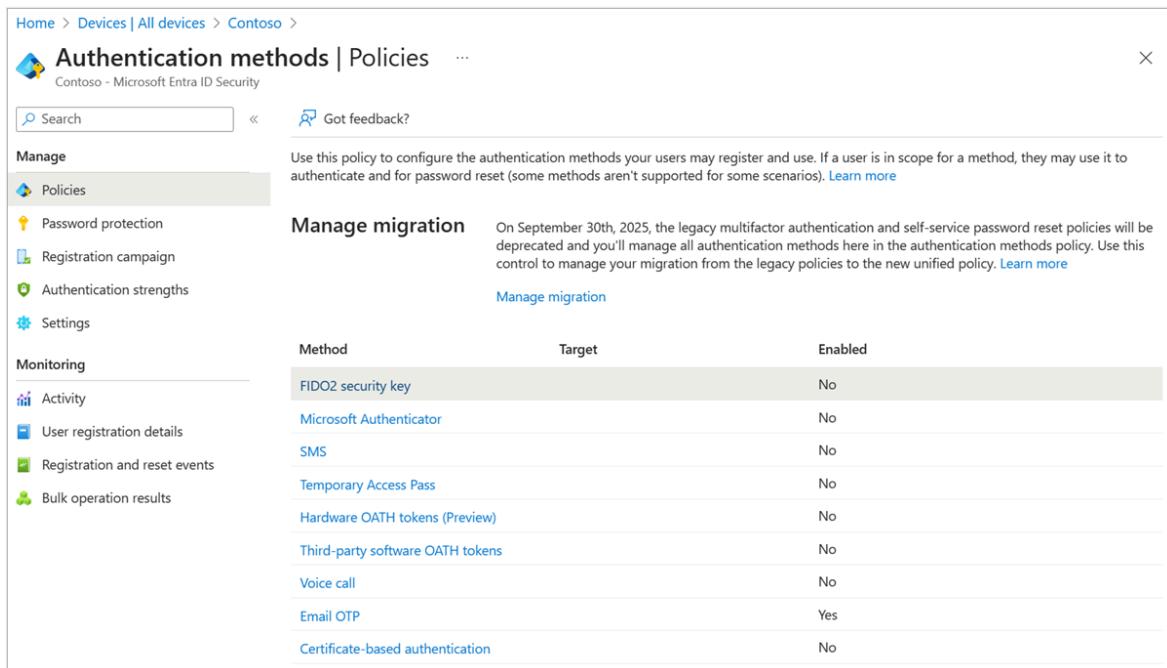
Article • 03/17/2025

This topic explains how Conditional Access authentication strength can restrict which authentication methods are allowed to access a resource.

How authentication strength works with the Authentication methods policy

There are two policies that determine which authentication methods can be used to access resources. If a user is enabled for an authentication method in either policy, they can sign in with that method.

- **Security > Authentication methods > Policies** is a more modern way to manage authentication methods for specific users and groups. You can specify users and groups for different methods. You can also configure parameters to control how a method can be used.



The screenshot shows the 'Authentication methods | Policies' page in the Microsoft Entra ID Security portal. The left sidebar has 'Manage' selected, with 'Policies' highlighted. Under 'Manage migration', it says: 'On September 30th, 2025, the legacy multifactor authentication and self-service password reset policies will be deprecated and you'll manage all authentication methods here in the authentication methods policy. Use this control to manage your migration from the legacy policies to the new unified policy.' A 'Manage migration' button is present. The main table lists authentication methods with columns: Method, Target, and Enabled. The methods listed are FIDO2 security key, Microsoft Authenticator, SMS, Temporary Access Pass, Hardware OATH tokens (Preview), Third-party software OATH tokens, Voice call, Email OTP, and Certificate-based authentication. The 'Enabled' column shows 'No' for most methods except 'Email OTP' which is 'Yes'.

Method	Target	Enabled
FIDO2 security key		No
Microsoft Authenticator		No
SMS		No
Temporary Access Pass		No
Hardware OATH tokens (Preview)		No
Third-party software OATH tokens		No
Voice call		No
Email OTP		Yes
Certificate-based authentication		No

- **Security > Multifactor authentication > Additional cloud-based multifactor authentication settings** is a legacy way to control multifactor authentication methods for all of the users in the tenant.

multi-factor authentication

users service settings

app passwords [\(learn more\)](#)

- Allow users to create app passwords to sign in to non-browser apps
- Do not allow users to create app passwords to sign in to non-browser apps

trusted ips [\(learn more\)](#)

- Skip multi-factor authentication for requests from federated users on my intranet

Skip multi-factor authentication for requests from following range of IP address subnets

192.168.0.0/16

verification options [\(learn more\)](#)

Methods available to users:

- Call to phone
- Text message to phone
- Notification through mobile app
- Verification code from mobile app or hardware token

remember multi-factor authentication on trusted device [\(learn more\)](#)

- Allow users to remember multi-factor authentication on devices they trust (between one to 365 days)

Number of days users can trust devices for

Users may register for authentication methods they're enabled for. An administrator can also configure a user's device with a method, such as certificate-based authentication.

How an authentication strength policy is evaluated during sign-in

The authentication strength Conditional Access policy defines which methods can be used. Microsoft Entra ID checks the policy during sign-in to determine the user's access to the resource. For example, an administrator configures a Conditional Access policy

with a custom authentication strength that requires a passkey (FIDO2 security key) or Password + text message. The user accesses a resource protected by this policy.

During sign-in, all settings are checked to determine which methods are allowed, which methods are registered, and which methods are required by the Conditional Access policy. To sign in, the method must be allowed, registered by the user (either before or as part of the access request), and satisfy the authentication strength.

How multiple Conditional Access authentication strength policies are evaluated

In general, when multiple Conditional Access policies apply for a sign-in, all conditions from all policies must be met. In the same vein, when multiple Conditional Access authentication strength policies apply to the sign-in, the user must satisfy all of the authentication strength conditions. For example, if two different authentication strength policies both require passkey (FIDO2), the user can use a FIDO2 security key to satisfy both policies. If the two authentication strength policies have different sets of methods, the user must use multiple methods to satisfy both policies.

How multiple Conditional Access authentication strength policies are evaluated for registering security info

For security info registration [Interrupt mode](#), the authentication strength evaluation is treated differently – authentication strengths that target the user action of **Registering security info** are preferred over other authentication strength policies that target **All resources (formerly 'All cloud apps')**. All other grant controls (such as **Require device to be marked as compliant**) from other Conditional Access policies in scope for the sign-in will apply as usual.

For example, let's assume Contoso would like to require their users to always sign in with a multifactor authentication method and from a compliant device. Contoso also wants to allow new employees to register these MFA methods using a Temporary Access Pass (TAP). TAP can't be used on any other resource. To achieve this goal, the admin can take the following steps:

1. Create a custom authentication strength named **Bootstrap and recovery** that includes the Temporary Access Pass authentication combination, it can also include any of the MFA methods.
2. Create a custom authentication strength named **MFA for sign-in** that includes all allowed MFA methods, without Temporary Access Pass.

3. Create a Conditional Access policy that targets **All resources (formerly 'All cloud apps')** and requires **MFA for sign-in** authentication strength AND **Require compliant device** grant controls.
4. Create a Conditional Access policy that targets the **Register security information** user action and requires the **Bootstrap and recovery** authentication strength.

As a result, users on a compliant device would be able to use a Temporary Access Pass to register any MFA method, and then use the newly registered method to authenticate to other resources like Outlook.

ⓘ Note

- If multiple Conditional Access policies target the **Register security information** user action, and they each apply an authentication strength, the user must satisfy all such authentication strengths to sign in.
- Some passwordless and phishing-resistant methods can't be registered from the Interrupt mode. For more information, see [Register passwordless authentication methods](#).

User experience

The following factors determine if the user gains access to the resource:

- Which authentication method was previously used?
- Which methods are available for the authentication strength?
- Which methods are allowed for user sign-in in the Authentication methods policy?
- Is the user registered for any available method?

When a user accesses a resource protected by an authentication strength Conditional Access policy, Microsoft Entra ID evaluates if the methods they have previously used satisfy the authentication strength. If a satisfactory method was used, Microsoft Entra ID grants access to the resource. For example, let's say a user signs in with password + text message. They access a resource protected by MFA authentication strength. In this case, the user can access the resource without another authentication prompt.

Let's suppose they next access a resource protected by Phishing-resistant MFA authentication strength. At this point, they'll be prompted to provide a phishing-resistant authentication method, such as Windows Hello for Business.

If the user hasn't registered for any methods that satisfy the authentication strength, they're redirected to [combined registration](#).

Users are required to register only one authentication method that satisfies the authentication strength requirement.

If the authentication strength doesn't include a method that the user can register and use, the user is blocked from sign-in to the resource.

Register passwordless authentication methods

The following authentication methods can't be registered as part of combined registration interrupt mode. Make sure users are registered for these methods before you apply a Conditional Access policy that can require them to be used for sign-in. If a user isn't registered for these methods, they can't access the resource until the required method is registered.

[] [Expand table](#)

Method	Registration requirements
Microsoft Authenticator (phone sign-in)	Can be registered from the Authenticator app.
Passkey(FIDO2)	Can be registered using combined registration managed mode and enforced by Authentication strengths using combined registration interrupt mode
Certificate-based authentication	Requires administrator setup; can't be registered by the user.
Windows Hello for Business	Can be registered in the Windows Out of Box Experience (OOBE) or the Windows Settings menu.

Federated user experience

For federated domains, MFA may be enforced by Microsoft Entra Conditional Access or by the on-premises federation provider by setting the `federatedIdpMfaBehavior`. If the `federatedIdpMfaBehavior` setting is set to `enforceMfaByFederatedIdp`, the user must authenticate on their federated IdP and can only satisfy the **Federated Multi-Factor** combination of the authentication strength requirement. For more information about the federation settings, see [Plan support for MFA](#).

If a user from a federated domain has multifactor authentication settings in scope for Staged Rollout, the user can complete multifactor authentication in the cloud and satisfy any of the **Federated single-factor + something you have** combinations. For more information about staged rollout, see [Enable Staged Rollout](#).

Next steps

- [Built-in authentication strengths](#)
 - [Custom authentication strengths](#)
 - [How authentication strength works for external users](#)
 - [Troubleshoot authentication strengths](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Conditional Access authentication strength

Article • 03/04/2025

Authentication strength is a Conditional Access control that specifies which combinations of authentication methods can be used to access a resource. Users can satisfy the strength requirements by authenticating with any of the allowed combinations.

For example, an authentication strength can require that only phishing-resistant authentication methods be used to access a sensitive resource. To access a nonsensitive resource, administrators can create another authentication strength that allows less secure multifactor authentication (MFA) combinations, such as password + text message.

Authentication strength is based on the [Authentication methods policy](#), where administrators can scope authentication methods for specific users and groups to be used across Microsoft Entra ID federated applications. Authentication strength allows further control over the usage of these methods based upon specific scenarios such as sensitive resource access, user risk, location, and more.

Scenarios for authentication strengths

Authentication strengths can help customers address these scenarios:

- Require specific authentication methods to access a sensitive resource.
- Require a specific authentication method when a user takes a sensitive action within an application (in combination with Conditional Access authentication context).
- Require users to use a specific authentication method when they access sensitive applications outside of the corporate network.
- Require more secure authentication methods for users at high risk.
- Require specific authentication methods from guest users who access a resource tenant (in combination with cross-tenant settings).

Authentication strengths

Administrators can specify an authentication strength to access a resource by creating a Conditional Access policy with the **Require authentication strength** control. They can

choose from three built-in authentication strengths: **Multifactor authentication strength**, **Passwordless MFA strength**, and **Phishing-resistant MFA strength**. They can also create a custom authentication strength based on the authentication method combinations they want to allow.

The screenshot shows the 'Grant' configuration dialog box within the Microsoft Conditional Access interface. The left pane displays the policy settings:

- Name:** Phish resistant MFA required
- Assignments:** 0 users and groups selected
- Target resources:** No target resources selected
- Conditions:** 0 conditions selected
- Access controls:** 0 controls selected
- Session:** 0 controls selected
- Enable policy:** Report-only (selected), On, Off

The right pane shows the 'Grant' configuration:

- Control access enforcement:** Grant access (selected)
- Require multifactor authentication:** Unselected
- Require authentication strength:** Selected, set to Phishing-resistant MFA
- Other options:** Unselected (Require device to be marked as compliant, Require Microsoft Entra hybrid joined device, Require approved client app, Require app protection policy)

A warning message is displayed: "⚠️ 'Require authentication strength' cannot be used with 'Require multifactor authentication'. [Learn more](#)".

Built-in authentication strengths

Built-in authentication strengths are combinations of authentication methods that are predefined by Microsoft. Built-in authentication strengths are always available and can't be modified. Microsoft will update built-in authentication strengths when new methods become available.

For an example, the built-in **Phishing-resistant MFA strength** allows the following combinations:

- Windows Hello for Business

Or

- FIDO2 security key

Or

- Microsoft Entra certificate-based authentication (Multifactor)

The screenshot shows the Microsoft Entra ID Security portal. On the left, there's a navigation sidebar with options like Policies, Password protection, Registration campaign, Authentication strengths (which is selected), Settings, Monitoring, Activity, User registration details, Registration and reset events, and Bulk operation results. The main area is titled "Authentication methods | Authentication strengths" and shows a table of authentication strengths. The table has columns for "Authentication strength", "Type", and "Authentication methods". It lists three rows: "Multifactor authentication" (Built-in, Windows Hello For Business and 16 more), "Passwordless MFA" (Built-in, Windows Hello For Business and 3 more), and "Phishing-resistant MFA" (Built-in, Windows Hello For Business and 2 more). Above the table, there are buttons for "Search", "New authentication strength", "Refresh", "Type: All", "Authentication methods: All", and "Reset filters". A tooltip for "Learn more" is also present. To the right, a modal window titled "View Authentication Strength" displays detailed information about the selected strength. It includes fields for "Name" (Phishing-resistant MFA), "Type" (Built-in), "Description" (Include authentication methods that are phishing-resistant like Passkeys (FIDO2) and Windows Hello for Business), "Authentication Flows" (Windows Hello For Business), and "OR" conditions (Passkeys (FIDO2) and Certificate-based Authentication (Multifactor)).

The combinations of authentication methods for each built-in authentication strength are listed in the following table. These combinations include methods that need to be registered by users and enabled in the Authentication methods policy or the legacy MFA settings policy.

- **MFA strength** - the same set of combinations that could be used to satisfy the **Require multifactor authentication** setting.
- **Passwordless MFA strength** - includes authentication methods that satisfy MFA but don't require a password.
- **Phishing-resistant MFA strength** - includes methods that require an interaction between the authentication method and the sign-in surface.

[] Expand table

Authentication method combination	MFA strength	Passwordless MFA strength	Phishing-resistant MFA strength
FIDO2 security key	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Windows Hello for Business	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Certificate-based authentication (Multi-Factor)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Microsoft Authenticator (Phone Sign-in)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Temporary Access Pass (One-time use AND Multi-use)	<input checked="" type="checkbox"/>		
Password + something you have ¹	<input checked="" type="checkbox"/>		

Authentication method combination	MFA strength	Passwordless MFA strength	Phishing-resistant MFA strength
Federated single-factor + something you have ¹	<input checked="" type="checkbox"/>		
Federated Multi-Factor	<input checked="" type="checkbox"/>		
Certificate-based authentication (single-factor)			
SMS sign-in			
Password			
Federated single-factor			

¹ Something you have refers to one of the following methods: text message, voice, push notification, software OATH token, or hardware OATH token.

The following API call can be used to list definitions of all the built-in authentication strengths:

HTTP

GET

```
https://graph.microsoft.com/beta/identity/conditionalAccess/authenticationStrength/policies?$filter=policyType eq 'builtIn'
```

Conditional Access Administrators can also create custom authentication strengths to exactly suit their access requirements. For more information, see [Custom Conditional Access authentication strengths](#).

Limitations

- **Conditional Access policies are only evaluated after the initial authentication -** As a result, authentication strength doesn't restrict a user's initial authentication. Suppose you are using the built-in phishing-resistant MFA strength. A user can still type in their password, but they are required to sign in with a phishing-resistant method such as FIDO2 security key before they can continue.
- **Require multifactor authentication and Require authentication strength can't be used together in the same Conditional Access policy -** These two Conditional Access grant controls can't be used together because the built-in authentication

strength **Multifactor authentication** is equivalent to the **Require multifactor authentication** grant control.

- **Authentication methods that aren't currently supported by authentication strength** - The **Email one-time pass (Guest)** authentication method isn't included in the available combinations.
- **Windows Hello for Business** – If the user signed in with Windows Hello for Business as their primary authentication method, it can be used to satisfy an authentication strength requirement that includes Windows Hello for Business. However, if the user signed in with another method like password as their primary authentication method, and the authentication strength requires Windows Hello for Business, they aren't prompted to sign in with Windows Hello for Business. The user needs to restart the session, choose **Sign-in options**, and select a method required by the authentication strength.

Known issues

- **Authentication strength and sign-in frequency** - When a resource requires an authentication strength and a sign-in frequency, users can satisfy both requirements at two different times.

For example, let's say a resource requires passkey (FIDO2) for the authentication strength, and a 1-hour sign-in frequency. 24 hours ago, a user signed in with passkey (FIDO2) to access the resource.

When the user unlocks their Windows device using Windows Hello for Business, they can access the resource again. Yesterday's sign-in satisfies the authentication strength requirement, and today's device unlock satisfies the sign-in frequency requirement.

- **Authentication strength blade double representation** - Platform credentials, such as Windows Hello for Business and **Platform Credential for macOS** are both represented in authentication strength under **Windows Hello For Business**. To configure a custom authentication strength that allows the use of **Platform Credential for macOS**, use **Windows Hello For Business**.

FAQ

Should I use authentication strength or the Authentication methods policy?

Authentication strength is based on the Authentication methods policy. The Authentication methods policy helps to scope and configure authentication methods to be used across Microsoft Entra ID by specific users and groups. Authentication strength allows another restriction of methods for specific scenarios, such as sensitive resource access, user risk, location, and more.

For example, the administrator of Contoso wants to allow their users to use Microsoft Authenticator with either push notifications or passwordless authentication mode. The administrator goes to the Microsoft Authenticator settings in the Authentication methods policy, scopes the policy for the relevant users, and sets the **Authentication mode** to Any.

Then for Contoso's most sensitive resource, the administrator wants to restrict the access to only passwordless authentication methods. The administrator creates a new Conditional Access policy, using the built-in **Passwordless MFA strength**.

As a result, users in Contoso can access most of the resources in the tenant using password + push notification from the Microsoft Authenticator OR only using Microsoft Authenticator (phone sign-in). However, when the users in the tenant access the sensitive application, they must use Microsoft Authenticator (phone sign-in).

Prerequisites

- **Microsoft Entra ID P1** - Your tenant needs to have Microsoft Entra ID P1 license to use Conditional Access. If needed, you can enable a [free trial](#).

Next steps

- [Custom Conditional Access authentication strengths](#)
- [How authentication strength works for external users](#)
- [Troubleshoot authentication strengths](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Custom Conditional Access authentication strengths

Article • 03/12/2025

Administrators can also create up to 15 of their own custom authentication strengths to exactly suit their requirements. A custom authentication strength can contain any of the supported combinations in the preceding table.

1. Sign in to the [Microsoft Entra admin center](#) as an Administrator.
2. Browse to **Protection > Authentication methods > Authentication strengths**.
3. Select **New authentication strength**.
4. Provide a descriptive **Name** for your new authentication strength.
5. Optionally provide a **Description**.
6. Select any of the available methods you want to allow.
7. Choose **Next** and review the policy configuration.

The screenshot shows the Microsoft Entra admin center interface. On the left, the navigation menu includes Home, Policies, Password protection, Registration campaign, Authentication strengths (which is selected), and Settings. The main content area displays the 'Authentication methods | Authentication strengths' page. It features a search bar, a 'New authentication strength' button, and a refresh button. Below this is a table with columns: Authentication strength, Type, and Authentication methods. The table lists three built-in authentication strengths: Multifactor authentication (Type: Built-in, Methods: Windows Hello For Business and 16 more), Passwordless MFA (Type: Built-in, Methods: Windows Hello For Business and 3 more), and Phishing-resistant MFA (Type: Built-in, Methods: Windows Hello For Business and 2 more). To the right, a modal window titled 'New authentication strength' is open. It has tabs for 'Configure' (selected) and 'Review'. Under 'Configure', there is a 'Name' field containing 'Contoso Require FIDO2' and a 'Description' field with the placeholder 'Add a description for your authentication strength'. Below these fields is a 'Search authentication combinations' input. A list of available methods is shown, with 'Passkeys (FIDO2)' checked. Other options include 'Phishing-resistant MFA (3)', 'Windows Hello For Business', 'Advanced options', 'Certificate-based Authentication (Multifactor)', 'Passwordless MFA (1)', 'Microsoft Authenticator (Phone Sign-in)', and 'Multifactor authentication (13)'. At the bottom of the modal are 'Previous' and 'Next' buttons.

Update and delete custom authentication strengths

You can edit a custom authentication strength. If it's referenced by a Conditional Access policy, it can't be deleted, and you need to confirm any edit. To check if an authentication strength is referenced by a Conditional Access policy, click the **Conditional Access policies** column.

FIDO2 security key advanced options

You can restrict the usage of FIDO2 security keys based on their Authenticator Attestation GUIDs (AAGUIDs). This capability allows administrators to require a FIDO2 security key from a specific manufacturer in order to access the resource. To require a specific FIDO2 security key, first create a custom authentication strength. Then select **FIDO2 Security Key**, and click **Advanced options**.

New authentication strength

X

Custom

Configure

Review

Name *

Require FIDO2 vendor

Description

Custom authentication strength that requires a FIDO2 security key from a specific vendor



Search authentication combinations



▼ Phishing-resistant MFA (3)



Windows Hello For Business



FIDO2 Security Key

[Advanced options](#)



Certificate-based Authentication (Multifactor)



▼ Passwordless MFA (1)



Microsoft Authenticator (Phone Sign-in)



▼ Multifactor authentication (13)



Temporary Access Pass (One-time use)



Temporary Access Pass (Multi-use)



Password + Microsoft Authenticator (Push Notification)



Password + Software OATH token



Password + Hardware OATH token



Password + SMS

Previous

Next

Next to Allowed FIDO2 Keys click +, copy the AAGUID value, and click Save.

FIDO2 Key advanced options

X

Enter a list of Authenticator Attestation GUIDs (AAGUIDs) that can be used to satisfy this authentication strength. Security keys with AAGUIDs not in this list will not be usable to satisfy this authentication strength.

[Learn more](#)

Allowed FIDO2 Keys +

4ea759d2-ab32-4ea6-ae4f-5b6f920cb511



Certificate-based authentication advanced options

In the [Authentication methods policy](#), you can configure whether certificates are bound in the system to single-factor or multifactor authentication protection levels, based on the certificate issuer or policy OID. You can also require single-factor or multifactor authentication certificates for specific resources, based on Conditional Access authentication strength policy.

By using authentication strength advanced options, you can require a specific certificate issuer or policy OID to further restrict sign-ins to an application.

For example, Contoso issues smart cards to employees with three different types of multifactor certificates. One certificate is for confidential clearance, another for secret clearance, and a third is for top secret clearance. Each one is distinguished by properties of the certificate, such as policy OID or issuer. Contoso wants to ensure that only users with the appropriate multifactor certificate can access data for each classification.

The next sections show how to configure advanced options for CBA by using the Microsoft Entra admin center and Microsoft Graph.

Microsoft Entra admin center

1. Sign in to the [Microsoft Entra admin center](#) as an Administrator.
2. Browse to **Protection > Authentication methods > Authentication strengths**.
3. Select **New authentication strength**.
4. Provide a descriptive **Name** for your new authentication strength.

5. Optionally provide a **Description**.
6. Below Certificate-based authentication (either single-factor or multifactor), click **Advanced options**.

New authentication strength

Custom

Configure **Review**

Name *

Name your authentication strength

Description

Add a description for your authentication strength

Search authentication combinations

- Phishing-resistant MFA (3)
- Windows Hello For Business
- FIDO2 Security Key
[Advanced options](#)
- Certificate-based Authentication (Multifactor)
[Advanced options](#)

7. You can select certificate issuers from the drop-down menu, type the certificate issuers and type the allowed policy OIDs. The drop-down menu lists all certificate authorities from the tenant irrespective of whether they're single-factor or multifactor. Certificate issuers can be configured either by using the drop down **Certificate issuers from the certificate authorities in your tenant** or by using **Other certificate issuer by SubjectkeyIdentifier** for scenarios where the certificate you would like to use is not uploaded to the Certificate authorities in your tenant. One such example is external user scenarios, where the user could be authenticating in their home tenant and auth strength is being enforced on the resource tenant.

Certificate-based authentication

X

When configured, one of the allowed certificate issuers and one of the allowed policy OIDs will be required during sign in.

[Learn more](#) 

Certificate issuers from the certificate authorities in your tenant

Select a certificate issuer



OR

Other Certificate Issuers by SubjectKeyIdentifier



Enter certificate issuer



AND

Custom policy OIDs



Enter policy OID



- If both attributes Certificate issuers AND Policy OIDs are configured, there's a AND relationship and the user has to use a certificate that has atleast one of the issuers AND one of the policy OID from the list to satisfy the authentication strength.
- If only Certificate issuers attribute is configured then the user has to use a certificate that has atleast one of the issuers to satisfy the authentication strength .
- If only Policy OIDs attribute is configured then the user has to use a certificate that has atleast one of the policy OIDs to satisfy the authentication strength.

Note

We allow a max of 5 issuers and 5 OIDs to be configured in authentication strengths configuration.

1. Click **Next** to review the configuration, then click **Create**.

Microsoft Graph

To create a new Conditional Access authentication strength policy with certificate combinationConfiguration:

JSON

```
POST /beta/identity/conditionalAccess/authenticationStrength/policies
{
    "displayName": "CBA Restriction",
    "description": "CBA Restriction with both IssuerSki and OIDs",
    "allowedCombinations": [
        "x509CertificateMultiFactor"
    ],
    "combinationConfigurations": [
        {
            "@odata.type":
"#microsoft.graph.x509CertificateCombinationConfiguration",
            "appliesToCombinations": [
                "x509CertificateMultiFactor"
            ],
            "allowedIssuerSkis":
["9A4248C6AC8C2931AB2A86537818E92E7B6C97B6"],
            "allowedPolicyOIDs": [
                "1.2.3.4.6",
                "1.2.3.4.5.6"
            ]
        }
    ]
}
```

To add a new combinationConfiguration to an existing policy:

JSON

```
POST
beta/identity/conditionalAccess/authenticationStrength/policies/{authenticationStrengthPolicyId}/combinationConfigurations

{
    "@odata.type":
"#microsoft.graph.x509CertificateCombinationConfiguration",
    "allowedIssuerSkis": [
        "9A4248C6AC8C2931AB2A86537818E92E7B6C97B6"
    ],
    "allowedPolicyOIDs": [],
    "appliesToCombinations": [
        "x509CertificateSingleFactor"
    ]
}
```

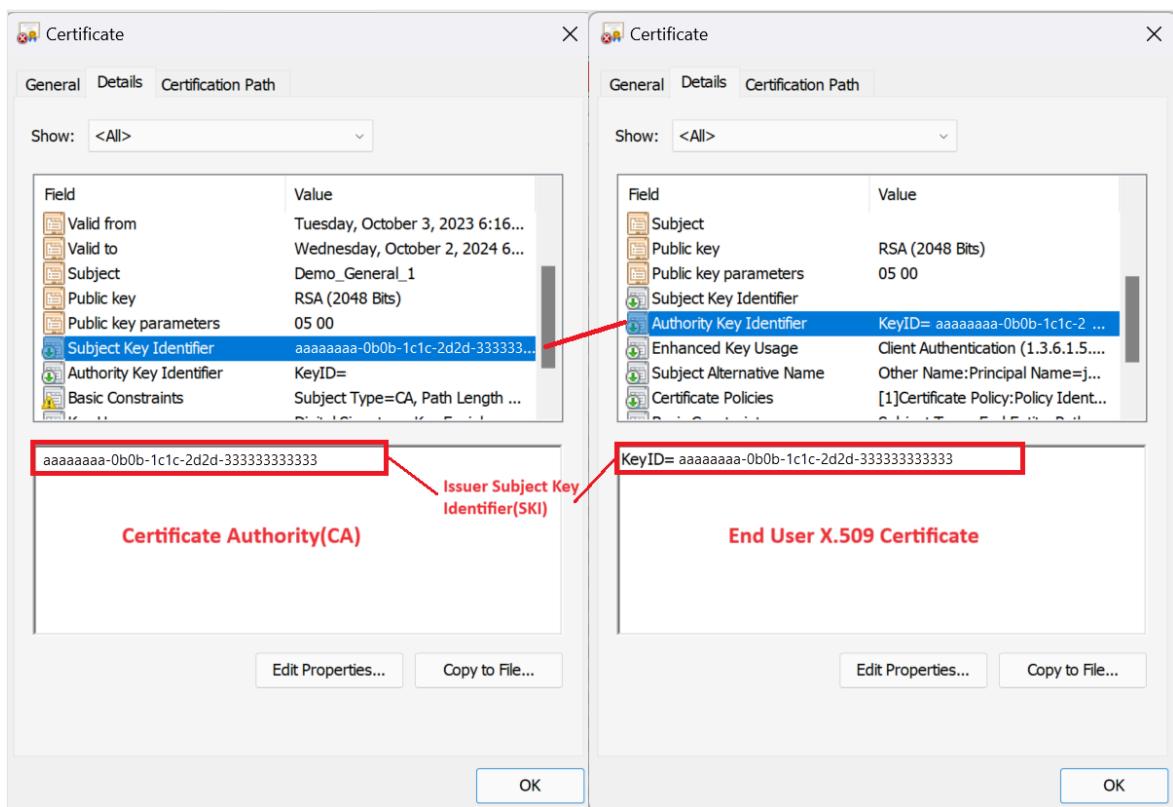
Limitations

FIDO2 security key advanced options

- FIDO2 security key Advanced options - Advanced options aren't supported for external users with a home tenant that is located in a different Microsoft cloud than the resource tenant.

Certificate-based authentication advanced options

- Only one certificate can be used in each browser session. After you sign in with a certificate, it's cached in the browser for the duration of the session. You won't be prompted to choose another certificate if it doesn't meet the authentication strength requirements. You need to sign out and sign back in to restart the session. Then choose the relevant certificate.
- Certificate Authorities and user certificates should conform to the X.509 v3 standard. Specifically, to enforce issuer SKI CBA restrictions, certificates need valid AKIs:



! Note

If the certificate doesn't conform, user authentication might succeed, but not satisfy the issuerSki restrictions for the authentication strength policy.

- During sign-in, the first 5 policy OIDs from the end user certificate are considered, and compared with the policy OIDs configured in the authentication strength policy. If the end user certificate has more than 5 policy OIDs, the first 5 policy

OIDs in lexical order that match the authentication strength requirements are taken into account.

- For B2B users, let's take an example where Contoso has invited users from Fabrikam to their tenant. In this case, Contoso is the resource tenant and Fabrikam is the home tenant.
 - When cross-tenant access setting is **Off** (Contoso doesn't accept MFA that was performed by the home tenant) - Using certificate-based authentication on the resource tenant isn't supported.
 - When cross-tenant access setting is **On**, Fabrikam and Contoso are on the same Microsoft cloud – meaning, both Fabrikam and Contoso tenants are on the Azure commercial cloud or on the Azure for US Government cloud. In addition, Contoso trusts MFA that was performed on the home tenant. In this case:
 - Access to a specific resource can be restricted by using the policy OIDs or the "other certificate issuer by SubjectkeyIdentifier" in the custom authentication strength policy.
 - Access to specific resources can be restricted by using the "Other certificate issuer by SubjectkeyIdentifier" setting in the custom authentication strength policy.
 - When cross-tenant access setting is **On**, Fabrikam and Contoso aren't on the same Microsoft cloud – for example, Fabrikam's tenant is on the Azure commercial cloud and Contoso's tenant is on the Azure for US Government cloud – access to specific resources can't be restricted by using the issuer ID or policy OIDs in the custom authentication strength policy.

Troubleshooting authentication strength advanced options

Users can't use their FIDO2 security key to sign in

A Conditional Access Administrator can restrict access to specific security keys. When a user tries to sign in by using a key they can't use, this **You can't get there from here** message appears. The user has to restart the session, and sign-in with a different FIDO2 security key.



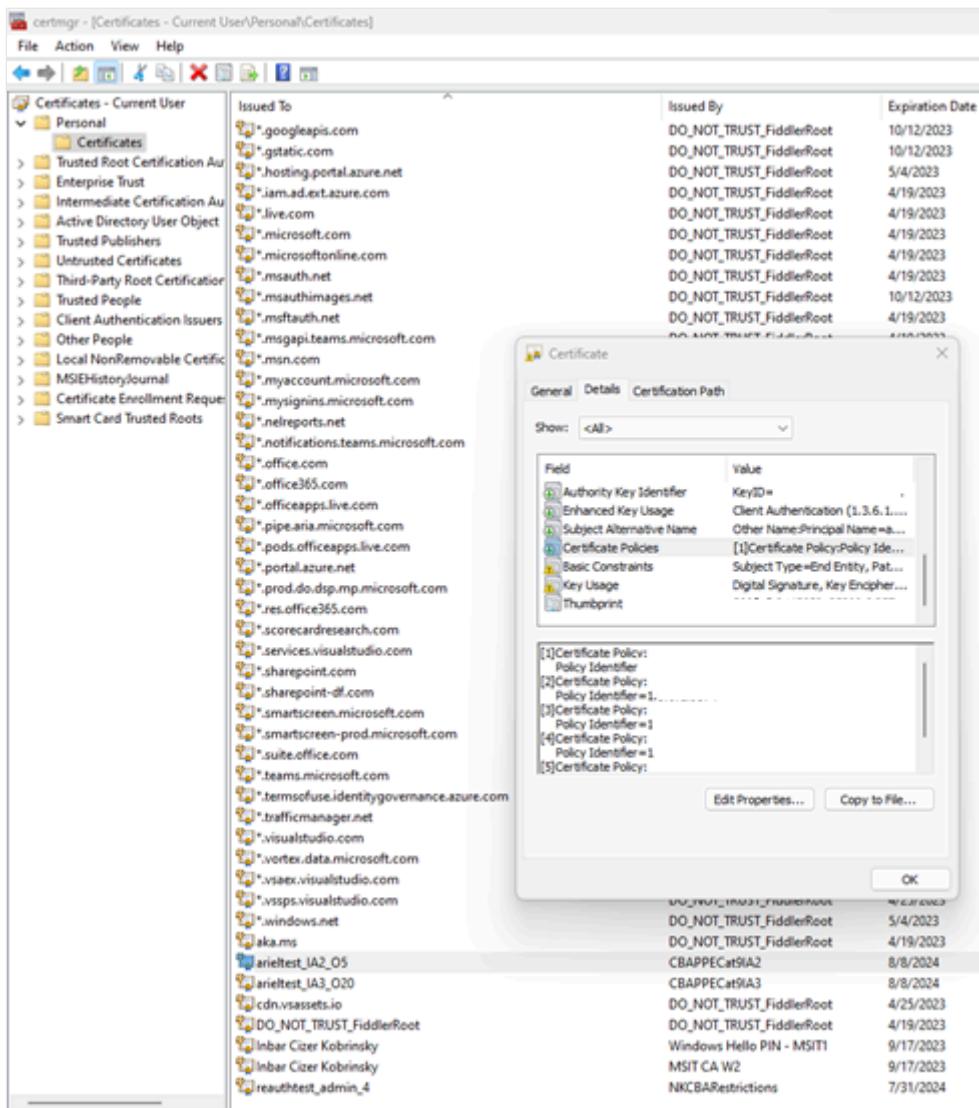
You can't get there from here

Your sign-in was successful but this security key does not meet the criteria to access this resource. Try signing in with a different key or contact your admin for help.

[More details](#)

How to check certificate policy OIDs and issuer

You can confirm the personal certificate properties match the configuration in authentication strength advanced options. On the user's device, sign in as an Administrator. Click **Run**, type `certmgr.msc`, and press Enter. To check policy OIDs, click **Personal**, right-click the certificate and click **Details**.



Next steps

- Built-in Conditional Access authentication strengths
- How authentication strength works for external users
- Troubleshoot authentication strengths

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

How Conditional Access authentication strength works for external users

Article • 03/04/2025

The Authentication methods policy is especially useful for restricting external access to sensitive apps in your organization because you can enforce specific authentication methods, such as phishing-resistant methods, for external users.

When you apply an authentication strength Conditional Access policy to external Microsoft Entra users, the policy works together with MFA trust settings in your cross-tenant access settings to determine where and how the external user must perform MFA. A Microsoft Entra user authenticates in their home Microsoft Entra tenant. Then when they access your resource, Microsoft Entra ID applies the policy and checks to see if you've enabled MFA trust. Note that enabling MFA trust is optional for B2B collaboration but is *required* for [B2B direct connect](#).

In external user scenarios, the authentication methods that can satisfy authentication strength vary, depending on whether the user is completing MFA in their home tenant or the resource tenant. The following table indicates the allowed methods in each tenant. If a resource tenant has opted to trust claims from external Microsoft Entra organizations, only those claims listed in the "Home tenant" column below will be accepted by the resource tenant for MFA. If the resource tenant has disabled MFA trust, the external user must complete MFA in the resource tenant using one of the methods listed in the "Resource tenant" column.

[] Expand table

Authentication method	Home tenant	Resource tenant
Text message as second factor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Voice call	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Microsoft Authenticator push notification	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Microsoft Authenticator phone sign-in	<input checked="" type="checkbox"/>	
OATH software token	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OATH hardware token	<input checked="" type="checkbox"/>	
FIDO2 security key	<input checked="" type="checkbox"/>	
Windows Hello for Business	<input checked="" type="checkbox"/>	

Authentication method	Home tenant	Resource tenant
Certificate-based Authentication	<input checked="" type="checkbox"/>	

For more information about how to set authentication strengths for external users, see [Conditional Access: Require an authentication strength for external users](#).

User experience for external users

An authentication strength Conditional Access policy works together with [MFA trust settings](#) in your cross-tenant access settings. First, a Microsoft Entra user authenticates with their own account in their home tenant. Then when this user tries to access your resource, Microsoft Entra ID applies the authentication strength Conditional Access policy and checks to see if you've enabled MFA trust.

- If **MFA trust is enabled**, Microsoft Entra ID checks the user's authentication session for a claim that indicates MFA was fulfilled in the user's home tenant. See the preceding table for authentication methods that are acceptable for MFA when completed in an external user's home tenant. If the session contains a claim that indicates the MFA policies are already met in the user's home tenant, and the methods satisfy the authentication strength requirements, the user is allowed access. Otherwise, Microsoft Entra ID presents the user with a challenge to complete MFA in the home tenant using an acceptable authentication method.
- If **MFA trust is disabled**, Microsoft Entra ID presents the user with a challenge to complete MFA in the resource tenant using an acceptable authentication method. See preceding table for authentication methods that are acceptable for MFA by an external user.

Next steps

- [Conditional Access authentication strength](#)
- [How authentication strength works](#)
- [Built-in Conditional Access authentication strengths](#)
- [Custom Conditional Access authentication strengths](#)
- [Troubleshoot authentication strengths](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Troubleshoot Conditional Access authentication strength

Article • 03/04/2025

This topic covers errors you might see when you use Microsoft Entra Conditional Access authentication strength and how to resolve them.

A user is asked to sign in with another method, but they don't see a method they expect

For sign in, the authentication method needs to be:

- Registered for the user
- Enabled by the Authentication methods policy

For more information, see [How Conditional Access authentication strength works](#).

To verify if a method can be used:

1. Check which authentication strength is required. Click **Security > Authentication methods > Authentication strengths**.
2. Check if the user is enabled for a required method:
 - a. Check the Authentication methods policy to see if the user is enabled for any method required by the authentication strength. Click **Security > Authentication methods > Policies**.
 - b. As needed, check if the tenant is enabled for any method required for the authentication strength. Click **Security > Multifactor Authentication > Additional cloud-based multifactor authentication settings**.
3. Check which authentication methods are registered for the user in the Authentication methods policy. Click **Users and groups > *username* > Authentication methods**.

If the user is registered for an enabled method that meets the authentication strength, they might need to use another method that isn't available after primary authentication, such as Windows Hello for Business. For more information, see [How each authentication method works](#). The user needs to restart the session, choose **Sign-in options**, and select a method required by the authentication strength.



← chrisgreen@woodgrove.com

Let's try something else

Another sign-in method is required to access this resource. Close your browser and try again, but choose another way to sign in:

- Use my password

[Sign out and sign in with a different account](#)

[More details](#)

A user can't access a resource

If an authentication strength requires a method that a user can't use, the user is blocked from sign-in. To check which method is required by an authentication strength, and which method the user is registered and enabled to use, follow the steps in the [previous section](#).

How to check which authentication strength was enforced during sign-in

Use the **Sign-ins** log to find more information about the sign-in:

- Under the **Authentication details** tab, the **Requirement** column shows the name of the authentication strength policy.

Activity Details: Sign-ins

X

Basic info Location Device info Authentication Details Conditional Access Report-only Additional Details

Authentication Policies Applied Session Lifetime Policies Applied

Conditional Access Remember multifactor authentication
Authentication Strength(s)

Date	Authentication meth...	Authentication meth...	Succeeded	Result detail	Requirement
9/7/2022, 1:46:57 PM	Password	Password in the cloud	true	Correct password	Phishing resistant MFA
9/7/2022, 1:46:57 PM	FIDO2 security key	Andres' key - f8a011f3...	true		Phishing resistant MFA
9/7/2022, 1:46:57 PM	Other		true	MFA requireme...	Phishing resistant MFA

- Under the **Conditional Access** tab, you can see which Conditional Access policy was applied. Click the name of the policy, and look for **Grant controls** to see the authentication strength that was enforced.

Conditional Access Policy details

X

↑ Previous ↓ Next

Policy: MSGraph - Phishing resistant

Policy state: Enabled

Result: Success

Assignments

User

Andres User ✓ Matched

Application

Graph Explorer ✓ Matched

Conditions

Sign-in risk

None ● Not configured

Device platform

Windows 10 ● Not configured

Location

Redmond, US ● Not configured

Client app

Browser ● Not configured

Device

Unknown ● Not configured

User risk

● Not configured

Access controls

Grant Controls

✓ Satisfied

Require Authentication Strength -
Phishing-resistant multifactor
authentication

Session Controls

● Not configured

A user can't register a new method during sign-in

Some methods can't be registered during sign-in, or they need more setup beyond the combined registration. For more information, see [Register passwordless authentication methods](#).

Keep your account secure

Your organization requires you to set up the following methods of proving who you are.

Additional authentication is required to complete this sign-in. Learn how to set up [a security key \(FIDO2\)](#), then go to <https://aka.ms/mysecurityinfo> to add the authentication method to your account.

You can also contact your admin to register one or more of these authentication methods for you:

- Certificate-based authentication (multi-factor)

Next steps

- [Built-in Conditional Access authentication strengths](#)
- [Custom Conditional Access authentication strengths](#)
- [How authentication strength works for external users](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

How it works: Microsoft Entra self-service password reset

Article • 03/04/2025

Microsoft Entra self-service password reset (SSPR) gives users the ability to change or reset their password, with no administrator or help desk involvement. If a user's account is locked or they forget their password, they can follow prompts to unblock themselves and get back to work. This ability reduces help desk calls and loss of productivity when a user can't sign in to their device or an application. We recommend this video on [how to enable and configure SSPR in Microsoft Entra ID](#).

Important

This conceptual article explains to an administrator how self-service password reset works. If you're an end user already registered for self-service password reset and need to get back into your account, go to <https://aka.ms/sspr>.

If your IT team hasn't enabled the ability to reset your own password, reach out to your helpdesk for additional assistance.

How does the password reset process work?

A user can reset or change their password using the [SSPR portal](#). They must first register their desired authentication methods. When a user accesses the SSPR portal, the Microsoft Entra platform considers the following factors:

- How should the page be localized?
- Is the user account valid?
- What organization does the user belong to?
- Where is the user's password managed?

When a user selects the **Can't access your account** link from an application or page, or goes directly to <https://aka.ms/sspr>, the language used in the SSPR portal is based on the following options:

- By default, the browser locale is used to display the SSPR in the appropriate language. The password reset experience is localized into the same languages that [Microsoft 365 supports](#).

- If you want to link to the SSPR in a specific localized language, append `?mkt=` to the end of the password reset URL along with the required locale.
 - For example, to specify the Spanish `es-us` locale, use `?mkt=es-us` - <https://passwordreset.microsoftonline.com/?mkt=es-us>.

After the SSPR portal is displayed in the required language, the user is prompted to enter a user ID and pass a captcha. Microsoft Entra ID now verifies that the user is able to use SSPR by doing the following checks:

- Checks that the user has SSPR enabled.
 - If the user isn't enabled for SSPR, the user is asked to contact their administrator to reset their password.
- Checks that the user has the right authentication methods defined on their account in accordance with administrator policy.
 - If the policy requires only one method, check that the user has the appropriate data defined for at least one of the authentication methods enabled by the administrator policy.
 - If the authentication methods aren't configured, the user is advised to contact their administrator to reset their password.
 - If the policy requires two methods, check that the user has the appropriate data defined for at least two of the authentication methods enabled by the administrator policy.
 - If the authentication methods aren't configured, the user is advised to contact their administrator to reset their password.
- Checks to see if the user's password is managed on-premises, such as if the Microsoft Entra tenant is using federated, pass-through authentication, or password hash synchronization:
 - If SSPR writeback is configured and the user's password is managed on-premises, the user is allowed to proceed to authenticate and reset their password.
 - If SSPR writeback isn't deployed and the user's password is managed on-premises, the user is asked to contact their administrator to reset their password.

If all of the previous checks are successfully completed, the user is guided through the process to reset or change their password.

 **Note**

SSPR may send email notifications to users as part of the password reset process.

These emails are sent using the SMTP relay service, which operates in an active-active mode across several regions.

SMTP relay services receive and process the email body, but don't store it. The body of the SSPR email that may potentially contain customer provided info isn't stored in the SMTP relay service logs. The logs only contain protocol metadata.

To get started with SSPR, complete the following tutorial:

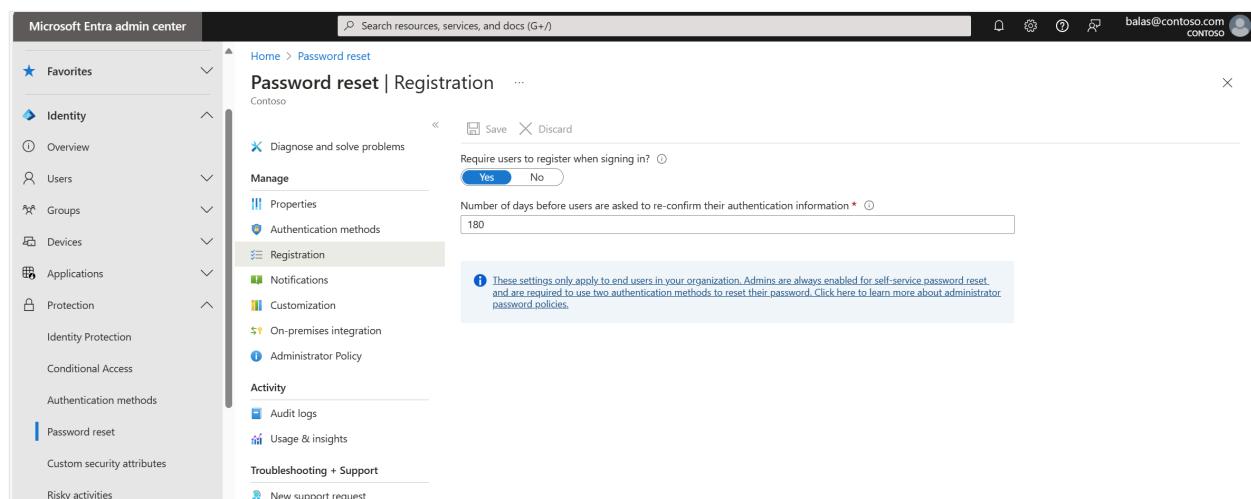
Tutorial: Enable self-service password reset (SSPR)

Require users to register when they sign in

You can enable the option to require a user to complete the SSPR registration if they use modern authentication or web browser to sign in to any applications using Microsoft Entra ID. This workflow includes the following applications:

- Microsoft 365
- Microsoft Entra admin center
- Access Panel
- Federated applications
- Custom applications using Microsoft Entra ID

When you don't require registration, users aren't prompted during sign-in, but they can manually register. Users can either visit <https://aka.ms/ssprsetup> or select the **Register for password reset** link under the **Profile** tab in the Access Panel.



The screenshot shows the Microsoft Entra admin center interface. The left sidebar has a 'Favorites' section and categories like Identity, Overview, Users, Groups, Devices, Applications, Protection, Identity Protection, Conditional Access, Authentication methods, Password reset, Custom security attributes, and Risky activities. The 'Password reset' category is currently selected. The main content area is titled 'Password reset | Registration' for the 'Contoso' tenant. It shows a 'Manage' section with 'Properties' and 'Authentication methods' options, and a 'Registration' section with 'Notifications', 'Customization', 'On-premises integration', and 'Administrator Policy'. A note at the bottom states: 'These settings only apply to end users in your organization. Admins are always enabled for self-service password reset and are required to use two authentication methods to reset their password. Click here to learn more about administrator password policies.' There are 'Save' and 'Discard' buttons at the top right of the form.

! Note

Users can dismiss the SSPR registration portal by selecting **cancel** or by closing the window. However, they're prompted to register each time they sign in until they complete their registration.

This interrupt to register for SSPR doesn't break the user's connection if they're already signed in.

Reconfirm authentication information

To make sure that authentication methods are correct when they're needed to reset or change their password, you can require users confirm their info registered information after a certain period of time. This option is only available if you enable the **Require users to register when signing in** option.

Valid values to prompt a user to confirm their registered methods are from *0* to *730* days. Setting this value to *0* means that users are never asked to confirm their authentication information. When using the combined registration experience users will be required to confirm their identity before reconfirming their information.

Authentication methods

When a user is enabled for SSPR, they must register at least one authentication method. We highly recommend that you choose two or more authentication methods so that your users have more flexibility in case they're unable to access one method when they need it. For more information, see [What are authentication methods?](#).

The following authentication methods are available for SSPR:

- Mobile app notification
- Mobile app code
- Email
- Mobile phone
- Office phone (available only for tenants with paid subscriptions)
- Security questions

Users can only reset their password if they register an authentication method that the administrator has enabled.

Warning

Accounts assigned Azure *administrator* roles are required to use methods as defined in the section [Administrator reset policy differences](#).

Method	Target	Enabled
FIDO2 security key	All users	Yes
Microsoft Authenticator	All users	Yes
SMS	All users	No
Temporary Access Pass	All users	Yes
Hardware OATH tokens (Preview)	All users	No
Third-party software OATH tokens	All users	Yes
Voice call	All users	Yes
Email OTP	All users	Yes
Certificate-based authentication	All users	No

Number of authentication methods required

You can configure the number of the available authentication methods a user must provide to reset or unlock their password. This value can be set to either *one* or *two*.

Users should register multiple authentication methods so they can sign-in another way if they're unable to access one method.

If a user doesn't register the minimum number of required methods, they see an error page when they try to use SSPR. They need to request that an administrator reset their password. For more information, see [Change authentication methods](#).

Mobile app and SSPR

When using a mobile app as a method for password reset, like Microsoft Authenticator, the following considerations apply if an organization hasn't [migrated to the centralized Authentication methods policy](#):

- When administrators require one method be used to reset a password, verification code is the only option available.
- When administrators require two methods be used to reset a password, users are able to use notification **OR** verification code in addition to any other enabled methods.

[+] Expand table

Number of methods required to reset	One	Two
Mobile app features available	Code	Code or Notification

Users can register their mobile app at <https://aka.ms/mfasetup>, or in the combined security info registration at <https://aka.ms/setupsecurityinfo>.

i Important

Authenticator can't be selected as the only authentication method when only one method is required. Similarly, Authenticator and only one additional method can't be selected if you require two methods.

When configuring SSPR policies that include the Authenticator app as a method, at least one additional method should be selected when one method is required, and at least two additional methods should be selected when configuring two methods are required.

Change authentication methods

If you start with a policy that has only one required authentication method for reset or unlock registered and you change that to two methods, what happens?

[] Expand table

Number of methods registered	Number of methods required	Result
1 or more	1	Able to reset or unlock
1	2	Unable to reset or unlock
2 or more	2	Able to reset or unlock

Changing the available authentication methods may also cause problems for users. If you change which authentication methods are available, users without the minimum amount of data available can't use SSPR.

Consider the following example scenario:

1. The original policy is configured with two authentication methods required. It uses only the office phone number and the security questions.
2. The administrator changes the policy to no longer use the security questions, but allows the use of a mobile phone and an alternate email.

3. Users without the mobile phone or alternate email fields populated now can't reset their passwords.

Notifications

To improve awareness of password events, SSPR lets you configure notifications for both the users and identity administrators.

Notify users on password resets

If this option is set to **Yes**, users resetting their password receive an email notifying them that their password has been changed. The email is sent via the SSPR portal to their primary and alternate email addresses that are stored in Microsoft Entra ID. If no primary or alternate email address is defined SSPR will attempt email notification via the users User Principal Name (UPN). No one else is notified of the reset event.

Notify all admins when other admins reset their passwords

If this option is set to **Yes**, then Global Administrators receive an email to their primary email address stored in Microsoft Entra ID. The email notifies them that another administrator has changed their password by using SSPR.

Note

Email notifications from the SSPR service will be sent from the following addresses based on the Azure cloud you are working with:

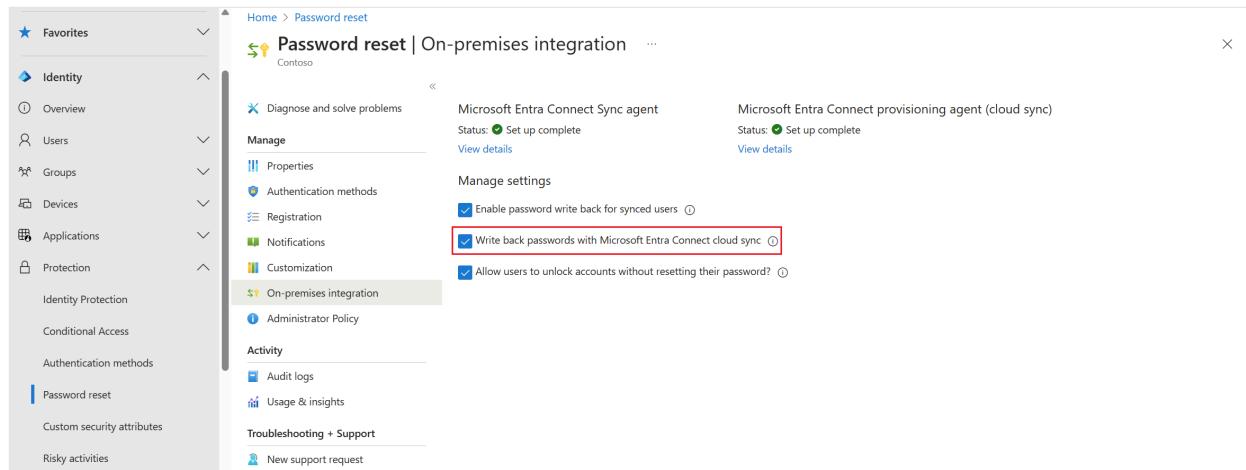
- Public: msonlineservicesteam@microsoft.com,
msonlineservicesteam@microsoftonline.com
- Microsoft Azure operated by 21Vianet (Azure in China):
msonlineservicesteam@oe.21vianet.com,
21Vianetonlineservicesteam@21vianet.com
- Azure for US Government: msonlineservicesteam@azureadnotifications.us,
msonlineservicesteam@microsoftonline.us

If you observe issues in receiving notifications, please check your spam settings.

If you want custom administrators to receive the notification emails, use SSPR customizations and [set up a custom helpdesk link or email](#).

On-premises integration

In a hybrid environment, you can configure Microsoft Entra Connect cloud sync to write password change events back from Microsoft Entra ID to an on-premises directory.



Microsoft Entra ID checks your current hybrid connectivity and provides messages in the Microsoft Entra admin center. For help with resolving possible errors, see [Troubleshoot Microsoft Entra Connect](#).

To get started with SSPR writeback, complete the following tutorial:

[Tutorial: Enable self-service password reset \(SSPR\) writeback](#)

Write back passwords to your on-premises directory

You can enable password writeback using the Microsoft Entra admin center. You can also temporarily disable password writeback without having to reconfigure Microsoft Entra Connect.

- If the option is set to **Yes**, then writeback is enabled. Federated, pass-through authentication, or password hash synchronized users are able to reset their passwords.
- If the option is set to **No**, then writeback is disabled. Federated, pass-through authentication, or password hash synchronized users aren't able to reset their passwords.

Allow users to unlock accounts without resetting their password

By default, Microsoft Entra ID unlocks accounts when it performs a password reset. To provide flexibility, you can choose to allow users to unlock their on-premises accounts without having to reset their password. Use this setting to separate those two operations.

- If set to **Yes**, users are given the option to reset their password and unlock the account, or to unlock their account without having to reset the password.
- If set to **No**, users are only be able to perform a combined password reset and account unlock operation.

On-premises Active Directory password filters

SSPR performs the equivalent of an admin-initiated password reset in Active Directory. If you use a third-party password filter to enforce custom password rules, and you require that this password filter is checked during Microsoft Entra self-service password reset, ensure that the third-party password filter solution is configured to apply in the admin password reset scenario. [Microsoft Entra password protection for Active Directory Domain Services](#) is supported by default.

Password reset for B2B users

Password reset and change are fully supported on all business-to-business (B2B) configurations. B2B user password reset is supported in the following three cases:

- **Users from a partner organization with an existing Microsoft Entra tenant:** If your partner has a Microsoft Entra tenant, we respect whatever password reset policies are enabled on that tenant. For password reset to work, the partner organization just needs to make sure that Microsoft Entra SSPR is enabled. There's no other charge for Microsoft 365 customers.
- **Users who sign up through self-service sign-up:** If your partner used the [self-service sign-up](#) feature to get into a tenant, we let them reset the password with the email they registered.
- **B2B users:** Any new B2B users created by using the new [Microsoft Entra B2B capabilities](#) can also reset their passwords with the email they registered during the invite process.

To test this scenario, go to <https://passwordreset.microsoftonline.com> with one of these partner users. If the user defined an alternate email or authentication email, password reset works as expected.

 **Note**

Microsoft accounts that are granted guest access to your Microsoft Entra tenant, such as those from Hotmail.com, Outlook.com, or other personal email addresses, can't use Microsoft Entra SSPR. For more information, see [When you can't sign in to your Microsoft account](#).

Next steps

To get started with SSPR, complete the following tutorial:

[Tutorial: Enable self-service password reset \(SSPR\)](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

How does self-service password reset writeback work in Microsoft Entra ID?

Article • 03/04/2025

Microsoft Entra self-service password reset (SSPR) lets users reset their passwords in the cloud, but most companies also have an on-premises Active Directory Domain Services (AD DS) environment for users. Password writeback allows password changes in the cloud to be written back to an on-premises directory in real time by using either [Microsoft Entra Connect](#) or [Microsoft Entra Connect cloud sync](#). When users change or reset their passwords using SSPR in the cloud, the updated passwords are also written back to the on-premises AD DS environment.

Important

This conceptual article explains to an administrator how self-service password reset writeback works. If you're an end user already registered for self-service password reset and need to get back into your account, go to <https://aka.ms/sspr>.

If your IT team hasn't enabled the ability to reset your own password, reach out to your helpdesk for additional assistance.

Password writeback is supported in environments that use the following hybrid identity models:

- [Password hash synchronization](#)
- [Pass-through authentication](#)
- [Active Directory Federation Services](#)

Password writeback provides the following features:

- **Enforcement of on-premises Active Directory Domain Services (AD DS) password policies:** When a user resets their password, it's checked to ensure it meets your on-premises AD DS policy before committing it to that directory. This review includes checking the history, complexity, age, password filters, and any other password restrictions that you define in AD DS.
- **Zero-delay feedback:** Password writeback is a synchronous operation. Users are notified immediately if their password doesn't meet the policy or can't be reset or changed for any reason.
- **Supports password changes from the access panel and Microsoft 365:** When federated or password hash synchronized users come to change their expired or

non-expired passwords, those passwords are written back to AD DS.

- **Supports password writeback when an admin resets them from the Microsoft Entra admin center:** When an admin resets a user's password in the [Microsoft Entra admin center](#), if that user is federated or password hash synchronized, the password is written back to on-premises. This functionality is currently not supported in the Office admin portal.
- **Doesn't require any inbound firewall rules:** Password writeback uses an Azure Service Bus relay as an underlying communication channel. All communication is outbound over port 443.
- **Supports side-by-side domain-level deployment** using [Microsoft Entra Connect](#) or [cloud sync](#) to target different sets of users depending on their needs, including users who are in disconnected domains.

Note

The on-premises service account that handles password write-back requests cannot change the passwords for users that belong to protected groups. Administrators can change their password in the cloud but they cannot use password write-back to reset a forgotten password for their on-premises user. For more information about protected groups, see [Protected accounts and groups in AD DS](#).

To get started with SSPR writeback, complete either one or both of the following tutorials:

- [Tutorial: Enable self-service password reset \(SSPR\) writeback](#)
- [Tutorial: Enable Microsoft Entra Connect cloud sync self-service password reset writeback to an on-premises environment \(Preview\)](#)

Microsoft Entra Connect and cloud sync side-by-side deployment

You can deploy Microsoft Entra Connect and cloud sync side-by-side in different domains to target different sets of users. This helps existing users continue to writeback password changes while adding the option in cases where users are in disconnected domains because of a company merger or split. Microsoft Entra Connect and cloud sync can be configured in different domains so users from one domain can use Microsoft Entra Connect while users in another domain use cloud sync. Cloud sync can also provide higher availability because it doesn't rely on a single instance of Microsoft Entra Connect. For a feature comparison between the two deployment options, see [Comparison between Microsoft Entra Connect and cloud sync](#).

How password writeback works

When a user account configured for federation, password hash synchronization (or, in the case of a Microsoft Entra Connect deployment, pass-through authentication) attempts to reset or change a password in the cloud, the following actions occur:

1. A check is performed to see what type of password the user has. If the password is managed on-premises:
 - A check is performed to see if the writeback service is up and running. If it is, the user can proceed.
 - If the writeback service is down, the user is informed that their password can't be reset right now.
2. Next, the user passes the appropriate authentication gates and reaches the **Reset password** page.
3. The user selects a new password and confirms it.
4. When the user selects **Submit**, the plaintext password is encrypted with a public key created during the writeback setup process.
5. The encrypted password is included in a payload that gets sent over an HTTPS channel to your tenant-specific service bus relay (that is set up for you during the writeback setup process). This relay is protected by a randomly generated password that only your on-premises installation knows.
6. After the message reaches the service bus, the password-reset endpoint automatically wakes up and sees that it has a reset request pending.
7. The service then looks for the user by using the cloud anchor attribute. For this lookup to succeed, the following conditions must be met:
 - The user object must exist in the AD DS connector space.
 - The user object must be linked to the corresponding metaverse (MV) object.
 - The user object must be linked to the corresponding Microsoft Entra connector object.
 - The link from the AD DS connector object to the MV must have the synchronization rule `Microsoft.InfromADUserAccountEnabled.xxx` on the link.

When the call comes in from the cloud, the synchronization engine uses the **cloudAnchor** attribute to look up the Microsoft Entra connector space object. It then follows the link back to the MV object, and then follows the link back to the AD DS object. Because there can be multiple AD DS objects (multi-forest) for the

same user, the sync engine relies on the `Microsoft.InfromADUserAccountEnabled.xxx` link to pick the correct one.

8. After the user account is found, an attempt to reset the password directly in the appropriate AD DS forest is made.
9. If the password set operation is successful, the user is told their password has been changed.

! Note

If the user's password hash is synchronized to Microsoft Entra ID by using password hash synchronization, there's a chance that the on-premises password policy is weaker than the cloud password policy. In this case, the on-premises policy is enforced. This policy ensures that your on-premises policy is enforced in the cloud, no matter if you use password hash synchronization or federation to provide single sign-on.

10. If the password set operation fails, an error prompts the user to try again. The operation might fail because of the following reasons:

- The service was down.
- The password they selected doesn't meet the organization's policies.
- Unable to find the user in local AD DS environment.

The error messages provide guidance to users so they can attempt to resolve without administrator intervention.

Password writeback security

Password writeback is a highly secure service. To ensure your information is protected, a four-tiered security model is enabled as follows:

- **Tenant-specific service-bus relay**
 - When you set up the service, a tenant-specific service bus relay is set up that's protected by a randomly generated strong password that Microsoft never has access to.
- **Locked down, cryptographically strong, password encryption key**
 - After the service bus relay is created, a strong symmetric key is created that is used to encrypt the password as it comes over the wire. This key only lives in your company's secret store in the cloud, which is heavily locked down and audited, just like any other password in the directory.

- **Industry standard Transport Layer Security (TLS)**
 1. When a password reset or change operation occurs in the cloud, the plaintext password is encrypted with your public key.
 2. The encrypted password is placed into an HTTPS message that's sent over an encrypted channel by using Microsoft TLS/SSL certs to your service bus relay.
 3. After the message arrives in the service bus, your on-premises agent wakes up and authenticates to the service bus by using the strong password that was previously generated.
 4. The on-premises agent picks up the encrypted message and decrypts it by using the private key.
 5. The on-premises agent attempts to set the password through the AD DS SetPassword API. This step is what allows enforcement of your AD DS on-premises password policy (such as the complexity, age, history, and filters) in the cloud.

- **Message expiration policies**

- If the message sits in service bus because your on-premises service is down, it times out and is removed after several minutes. The time-out and removal of the message increases security even further.

Password writeback encryption details

After a user submits a password reset, the reset request goes through several encryption steps before it arrives in your on-premises environment. These encryption steps ensure maximum service reliability and security. They are described as follows:

1. **Password encryption with 2048-bit RSA Key:** After a user submits a password to be written back to on-premises, the submitted password itself is encrypted with a 2048-bit RSA key.
2. **Package-level encryption with 256-bit AES-GCM:** The entire package, the password plus the required metadata, is encrypted by using AES-GCM (with a key size of 256 bits). This encryption prevents anyone with direct access to the underlying Service Bus channel from viewing or tampering with the contents.
3. **All communication occurs over TLS/SSL:** All the communication with Service Bus happens in an SSL/TLS channel. This encryption secures the contents from unauthorized third parties.
4. **Automatic key rollover every six months:** All keys roll over every six months, or every time password writeback is disabled and then re-enabled on Microsoft Entra Connect, to ensure maximum service security and safety.

Password writeback bandwidth usage

Password writeback is a low-bandwidth service that only sends requests back to the on-premises agent under the following circumstances:

- Two messages are sent when the feature is enabled or disabled through Microsoft Entra Connect.
- One message is sent once every five minutes as a service heartbeat for as long as the service is running.
- Two messages are sent each time a new password is submitted:
 - The first message is a request to perform the operation.
 - The second message contains the result of the operation, and is sent in the following circumstances:
 - Each time a new password is submitted during a user self-service password reset.
 - Each time a new password is submitted during a user password change operation.
 - Each time a new password is submitted during an admin-initiated user password reset (only from Entra admin portals).

Message size and bandwidth considerations

The size of each of the message described previously is typically under 1 KB. Even under extreme loads, the password writeback service itself is consuming a few kilobits per second of bandwidth. Because each message is sent in real time, only when required by a password update operation, and because the message size is so small, the bandwidth usage of the writeback capability is too small to have a measurable impact.

Supported writeback operations

Passwords are written back in all the following situations:

- **Supported end-user operations**
 - Any end-user self-service voluntary change password operation.
 - Any end-user self-service force change password operation, for example, password expiration.
 - Any end-user self-service password reset that originates from the [password reset portal ↗](#).
- **Supported administrator operations**
 - Any administrator self-service voluntary change password operation.

- Any administrator self-service force change password operation, for example, password expiration.
- Any administrator self-service password reset that originates from the [password reset portal](#).
- Any administrator-initiated end-user password reset from the Microsoft Entra admin center.
- Any administrator-initiated end-user password reset from the [Microsoft Graph API](#).

Unsupported writeback operations

Passwords aren't written back in any of the following situations:

- **Unsupported end-user operations**
 - Any end user resetting their own password by using PowerShell version 1, version 2, or the Microsoft Graph API.
- **Unsupported administrator operations**
 - Any administrator-initiated end-user password reset from PowerShell version 1, or version 2.
 - Any administrator-initiated end-user password reset from the [Microsoft 365 admin center](#).
 - Any administrator cannot use password reset tool to reset their own password for password writeback.

Note

If a user has the option "Password never expires" set in Active Directory (AD), the force password change flag will not be set in Active Directory (AD), so the user will not be prompted to change the password during the next sign-in even if the option to force the user to change their password on next logon option is selected during an administrator-initiated end-user password reset.

Next steps

To get started with SSPR writeback, complete the following tutorial:

[Tutorial: Enable self-service password reset \(SSPR\) writeback](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Password policies and account restrictions in Microsoft Entra ID

Article • 05/15/2025

In Microsoft Entra ID, there's a password policy that defines settings like the password complexity, length, or age. There's also a policy that defines acceptable characters and length for usernames.

When self-service password reset (SSPR) is used to change or reset a password in Microsoft Entra ID, the password policy is checked. If the password doesn't meet the policy requirements, the user is prompted to try again. Azure administrators have some restrictions on using SSPR that are different to regular user accounts, and there are minor exceptions for trial and free versions of Microsoft Entra ID.

This article describes the password policy settings and complexity requirements associated with user accounts. It also covers how to use PowerShell to check or set password expiration settings.

Username policies

Every account that signs in to Microsoft Entra ID must have a unique user principal name (UPN) attribute value associated with their account. In hybrid environments with an on-premises Active Directory Domain Services environment synchronized to Microsoft Entra ID using Microsoft Entra Connect, by default the Microsoft Entra ID UPN is set to the on-premises UPN.

The following table outlines the username policies that apply to both on-premises accounts that are synchronized to Microsoft Entra ID, and for cloud-only user accounts created directly in Microsoft Entra ID:

 Expand table

Property	UserPrincipalName requirements
Characters allowed	A-Z a-z 0-9 . - _ ! # ^ ~
Characters not allowed	Any "@" character that's not separating the username from the domain. Can't contain a period character "." immediately preceding the "@" symbol
Length constraints	The total length must not exceed 113 characters There can be up to 64 characters before the "@" symbol

Property	UserPrincipalName requirements
	There can be up to 48 characters after the "@" symbol

Microsoft Entra password policies

A password policy is applied to all user accounts that are created and managed directly in Microsoft Entra ID. Some of these password policy settings can't be modified, though you can [configure custom banned passwords for Microsoft Entra password protection](#) or account lockout parameters.

By default, an account is locked out after 10 unsuccessful sign-in attempts with the wrong password. The user is locked out for one minute. The lockout duration increases after further incorrect sign-in attempts. [Smart lockout](#) tracks the last three bad password hashes to avoid incrementing the lockout counter for the same password. If someone enters the same bad password multiple times, they aren't locked out. You can define the smart lockout threshold and duration.

The following Microsoft Entra password policy options are defined. Unless noted, you can't change these settings:

[] [Expand table](#)

Property	Requirements
Characters allowed	A-Z a-z 0-9 @ # \$ % ^ & * - _ ! + = [] { } \ : ' , . ? / ` ~ " () ; < > Blank space
Characters not allowed	Unicode characters
Password restrictions	A minimum of 8 characters and a maximum of 256 characters. Requires three out of four of the following types of characters: - Lowercase characters - Uppercase characters - Numbers (0-9) - Symbols (see the previous password restrictions)
Password expiry duration (Maximum password age)	Default value: No expiration . If the tenant was created before 2021, it has a 90 day expiration value by default. You can check current policy with Get-MgDomain . The value is configurable by using the Update-MgDomain cmdlet from the Microsoft Graph module for PowerShell.

Property	Requirements
Password expiry (Let passwords never expire)	Default value: false (indicates that passwords have an expiration date). The value can be configured for individual user accounts by using the Update-MgUser cmdlet.
Password change history	The last password <i>can't</i> be used again when the user changes a password.
Password reset history	The last password <i>can</i> be used again when the user resets a forgotten password.

If you enable [EnforceCloudPasswordPolicyForPasswordSyncedUsers](#), the Microsoft Entra password policy applies to user accounts synchronized from on-premises using Microsoft Entra Connect. In addition, if a user changes a password on-premises to include a unicode character, the password change may succeed on-premises but not in Microsoft Entra ID. If password hash synchronization is enabled with Microsoft Entra Connect, the user can still receive an access token for cloud resources. But if the tenant enables [User risk-based password change](#), the password change is reported as high risk.

The user is prompted to change their password again. But if the change still includes a unicode character, they could get locked out if [smart lockout](#) is also enabled.

Risk based password reset policy limitations

If you enable [EnforceCloudPasswordPolicyForPasswordSyncedUsers](#), a cloud password change is required once a high risk is identified. The user is prompted to change their password when they sign in to Microsoft Entra ID. The new password must comply with both the cloud and on-premises password policies.

If a password change meets on-premises requirements but fails to meet cloud requirements, the password change succeeds if password hash synchronization is enabled. For example, if the new password includes a Unicode character, the password change can be updated on-premises but not in the cloud.

If the password didn't comply with the cloud password requirements, it isn't updated in the cloud, and the account risk doesn't decrease. The user still receives an access token for cloud resources, but they're prompted to change their password again the next time they access cloud resources. The user doesn't see any error or notification that their chosen password failed to meet the cloud requirements.

Administrator reset policy differences

By default, administrator accounts are enabled for self-service password reset, and a strong default *two-gate* password reset policy is enforced. This policy may be different from the one you defined for your users, and this policy can't be changed. You should always test password reset functionality as a user without any Azure administrator roles assigned.

The two-gate policy requires two pieces of authentication data, such as an email address, authenticator app, or a phone number, and it prohibits security questions. Office and mobile voice calls are also prohibited for trial or free versions of Microsoft Entra ID.

The SSPR administrator policy doesn't depend upon the *Authentications method* policy. For example, if you disable third party software tokens in the *Authentication methods* policy, administrator accounts can still register third party software token applications and use them, but only for SSPR.

A two-gate policy applies in the following circumstances:

- All the following Azure administrator roles are affected:
 - Application Administrator
 - Authentication Administrator
 - Billing Administrator
 - Compliance Administrator
 - Cloud Device Administrator
 - Directory Synchronization Accounts (an admin role assigned to the Microsoft Entra Connect service)
 - Directory Writers
 - Dynamics 365 Administrator
 - Exchange Administrator
 - Global Administrator
 - Helpdesk Administrator
 - Intune Administrator
 - Microsoft Entra Joined Device Local Administrator
 - Partner Tier1 Support
 - Partner Tier2 Support
 - Password Administrator
 - Power Platform Administrator
 - Privileged Authentication Administrator
 - Privileged Role Administrator
 - Security Administrator
 - Service Support Administrator
 - SharePoint Administrator
 - Skype for Business Administrator
 - Teams Administrator

- Teams Communications Administrator
 - Teams Devices Administrator
 - User Administrator
- If 30 days elapsed in a trial subscription
 - Or-
 - A custom domain is configured for your Microsoft Entra tenant, such as *contoso.com*
 - Or-
 - Microsoft Entra Connect synchronizes identities from your on-premises directory

You can disable the use of SSPR for administrator accounts using the [Update-MgPolicyAuthorizationPolicy](#) PowerShell cmdlet. The `-AllowedToUseSspr:$true|$false` parameter enables/disables SSPR for administrators. Policy changes to enable or disable SSPR for administrator accounts can take up to 60 minutes to take effect.

Exceptions

A one-gate policy requires one piece of authentication data, such as an email address or phone number. A one-gate policy applies in the following circumstances:

- It's within the first 30 days of a trial subscription
 - Or-
- A custom domain isn't configured (the tenant is using the default **.onmicrosoft.com*, which isn't recommended for production use) and Microsoft Entra Connect isn't synchronizing identities.

Password expiration policies

User Administrators can use the [Microsoft Graph](#) to set user passwords not to expire.

You can also use PowerShell cmdlets to remove the never-expires configuration or to see which user passwords are set to never expire.

This guidance applies to other providers, such as Intune and Microsoft 365, which also rely on Microsoft Entra ID for identity and directory services. Password expiration is the only part of the policy that can be changed.

 Note

By default only passwords for user accounts that aren't synchronized through Microsoft Entra Connect can be configured to not expire. For more information about directory synchronization, see [Connect AD with Microsoft Entra ID](#).

Set or check the password policies by using PowerShell

To get started, [download and install the Microsoft Graph PowerShell module](#) and [connect it to your Microsoft Entra tenant](#).

After the module is installed, use the following steps to complete each task as needed.

Check the expiration policy for a password

1. Open a PowerShell prompt and [connect to your Microsoft Entra tenant](#) as at least a [User Administrator](#).
2. Run one of the following commands for either an individual user or for all users:
 - To see if a single user's password is set to never expire, run the following cmdlet. Replace <user ID> with the user ID of the user you want to check:

PowerShell

```
Get-MgUser -UserId <user ID> -Property UserPrincipalName,  
PasswordPolicies | Select-Object @{N="PasswordNeverExpires";E=  
{$_._.PasswordPolicies -contains "DisablePasswordExpiration"}}
```

- To see the **Password never expires** setting for all users, run the following cmdlet:

PowerShell

```
Get-MgUser -All -Property UserPrincipalName, PasswordPolicies | Select-  
Object UserPrincipalName, @{N="PasswordNeverExpires";E=  
{$_._.PasswordPolicies -contains "DisablePasswordExpiration"}}
```

Set a password to expire

1. Open a PowerShell prompt and [connect to your Microsoft Entra tenant](#) as at least a [User Administrator](#).
2. Run one of the following commands for either an individual user or for all users:

- To set the password of one user so that the password expires, run the following cmdlet. Replace <user ID> with the user ID of the user you want to check:

```
PowerShell
```

```
Update-MgUser -UserId <user ID> -PasswordPolicies None
```

- To set the passwords of all users in the organization so that they expire, use the following command:

```
PowerShell
```

```
Get-MgUser -All | foreach $_ { Update-MgUser -UserId $_.Id -  
PasswordPolicies None }
```

Set a password to never expire

- Open a PowerShell prompt and [connect to your Microsoft Entra tenant](#) as at least a [User Administrator](#).
- Run one of the following commands for either an individual user or for all users:

- To set the password of one user to never expire, run the following cmdlet. Replace <user ID> with the user ID of the user you want to check:

```
PowerShell
```

```
Update-MgUser -UserId <user ID> -PasswordPolicies  
DisablePasswordExpiration
```

- To set the passwords of all the users in an organization to never expire, run the following cmdlet:

```
PowerShell
```

```
Get-MgUser -All | foreach $_ { Update-MgUser -UserId $_.Id -  
PasswordPolicies DisablePasswordExpiration }
```

⚠ Warning

Passwords set to `-PasswordPolicies DisablePasswordExpiration` still age based on the `LastPasswordChangeDateTime` attribute. Based on the `LastPasswordChangeDateTime`

attribute, if you change the expiration to `-PasswordPolicies None`, all passwords that have a `LastPasswordChangeDateTime` older than 90 days require the user to change them the next time they sign in. This change can affect a large number of users.

Next steps

To get started with SSPR, see [Tutorial: Enable users to unlock their account or reset passwords using Microsoft Entra self-service password reset](#).

If you or users have problems with SSPR, see [Troubleshoot self-service password reset](#).

Licensing requirements for Microsoft Entra self-service password reset

Article • 03/04/2025

To reduce help desk calls and loss of productivity when a user can't sign in to their device or an application, user accounts in Microsoft Entra ID can be enabled for self-service password reset (SSPR). Features that make up SSPR include password change, reset, unlock, and writeback to an on-premises directory. Basic SSPR features are available in Microsoft 365 Business Standard or higher and all Microsoft Entra ID P1 or P2 SKUs at no cost.

This article details the different ways that self-service password reset can be licensed and used. For specific details about pricing and billing, see the [Microsoft Entra pricing page](#).

Although some unlicensed users may technically be able to access SSPR, a license is required for any user that you intend to benefit from the service.

! Note

Some tenant services are not currently capable of limiting benefits to specific users. Efforts should be taken to limit the service benefits to licensed users. This helps avoid potential service disruption to your organization once targeting capabilities are available.

Compare editions and features

The following table outlines the different SSPR scenarios for password change, reset, or on-premises writeback, and which SKUs provide the feature.

 Expand table

Feature	Microsoft Entra ID Free	Microsoft 365 Business Standard	Microsoft 365 Business Premium	Microsoft Entra ID P1 or P2
Cloud-only user password change When a user in Microsoft Entra ID	●	●	●	●

Feature	Microsoft Entra ID Free	Microsoft 365 Business Standard	Microsoft 365 Business Premium	Microsoft Entra ID P1 or P2
knows their password and wants to change it to something new.				
Cloud-only user password reset When a user in Microsoft Entra ID forgets their password and needs to reset it.		•	•	•
Hybrid user password change or reset with on-prem writeback When a user in Microsoft Entra synchronized from an on-premises directory using Microsoft Entra Connect wants to change or reset their password and also write the new password back to on-premises.			•	•

⚠️ Warning

Standalone Microsoft 365 Basic and Standard licensing plans don't support SSPR with on-premises writeback. The on-premises writeback feature requires Microsoft Entra ID P1, Premium P2, or Microsoft 365 Business Premium.

For additional licensing information, including costs, see the following pages:

- [Microsoft 365 licensing guidance for security & compliance](#)
- [Microsoft Entra pricing ↗](#)
- [Microsoft Entra features and capabilities ↗](#)
- [Enterprise Mobility + Security ↗](#)
- [Microsoft 365 Enterprise ↗](#)
- [Microsoft 365 Business](#)

Next steps

To get started with SSPR, complete the following tutorial:

[Tutorial: Enable self-service password reset \(SSPR\)](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Plan a Microsoft Entra self-service password reset deployment

Article • 03/04/2025

ⓘ Important

This deployment plan offers guidance and best practices for deploying Microsoft Entra self-service password reset (SSPR).

If you're an end user and need to get back into your account, go to <https://aka.ms/sspr>.

[Self-Service Password Reset \(SSPR\)](#) is a Microsoft Entra feature that enables users to reset their passwords without contacting IT staff for help. The users can quickly unblock themselves and continue working no matter where they're or time of day. By allowing the employees to unblock themselves, your organization can reduce the non-productive time and high support costs for most common password-related issues.

SSPR has the following key capabilities:

- Self-service allows end users to reset their expired or non-expired passwords without contacting an administrator or helpdesk for support.
- [Password Writeback](#) allows management of on-premises passwords and resolution of account lockout through the cloud.
- Password management activity reports give administrators insight into password reset and registration activity occurring in their organization.

This deployment guide shows you how to plan and then test an SSPR roll-out.

To quickly see SSPR in action and then come back to understand additional deployment considerations:

[Enable self-service password reset \(SSPR\)](#)

💡 Tip

As a companion to this article, we recommend using the [Plan your self-service password reset deployment guide](#) when signed in to the Microsoft 365 Admin Center. This guide customizes your experience based on your environment. To

review best practices without signing in and activating automated setup features, go to the [M365 Setup portal](#).

Learn about SSPR

Learn more about SSPR. See [How it works: Microsoft Entra self-service password reset](#).

Key benefits

The key benefits of enabling SSPR are:

- **Manage cost.** SSPR reduces IT support costs by enabling users to reset passwords on their own. It also reduces the cost of time lost due to lost passwords and lockouts.
- **Intuitive user experience.** It provides an intuitive one-time user registration process that allows users to reset passwords and unblock accounts on-demand from any device or location. SSPR allows users to get back to work faster and be more productive.
- **Flexibility and security.** SSPR enables enterprises to access the security and flexibility that a cloud platform provides. Administrators can change settings to accommodate new security requirements and roll these changes out to users without disrupting their sign-in.
- **Robust auditing and usage tracking.** An organization can ensure that the business systems remain secure while its users reset their own passwords. Robust audit logs include information of each step of the password reset process. These logs are available from an API and enable the user to import the data into a Security Incident and Event Monitoring (SIEM) system of choice.

Licensing

Microsoft Entra ID is licensed per-user meaning each user requires an appropriate license for the features they use. We recommend group-based licensing for SSPR.

To compare editions and features and enable group or user-based licensing, see [Licensing requirements for Microsoft Entra self-service password reset](#).

For more information about pricing, see [Microsoft Entra pricing](#).

Prerequisites

- A working Microsoft Entra tenant with at least a trial license enabled. If needed, [create one for free ↗](#).
- You must be assigned at least an [Authentication Policy Administrator](#) role.

Guided walkthrough

For a guided walkthrough of many of the recommendations in this article, see the [Plan your self-service password reset deployment ↗](#) guide when signed in to the Microsoft 365 Admin Center. To review best practices without signing in and activating automated setup features, go to the [M365 Setup portal ↗](#).

Training resources

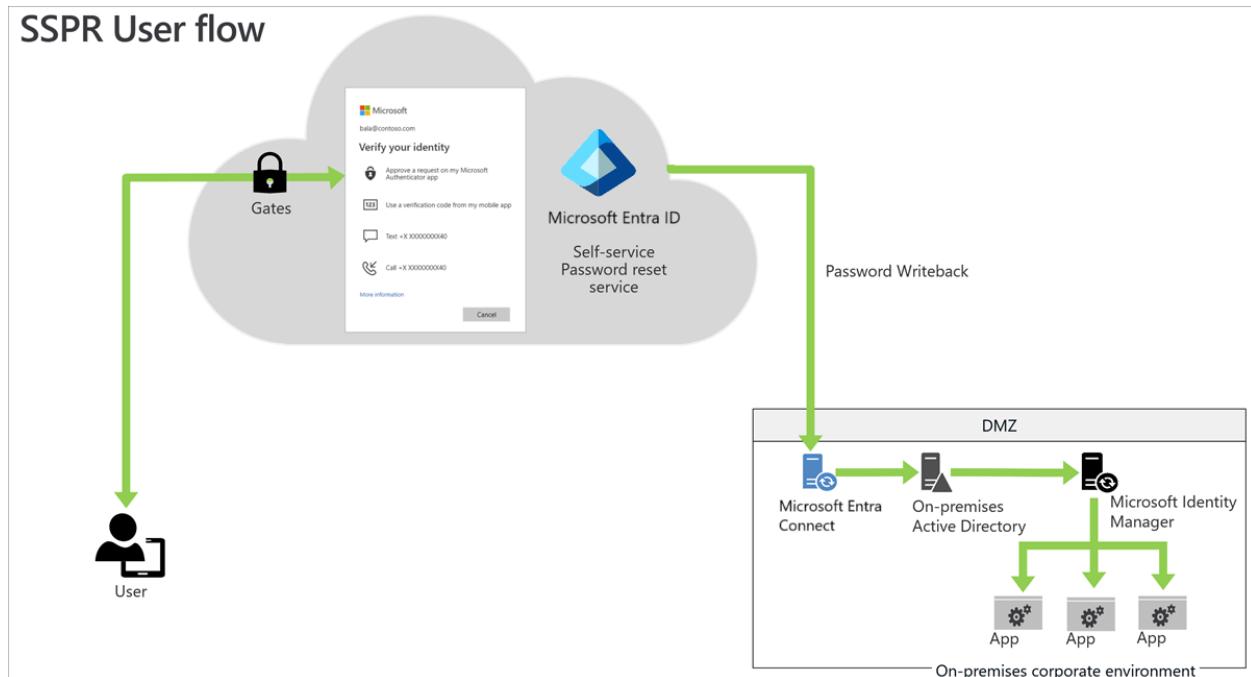
[+] [Expand table](#)

Resources	Link and Description
Videos	Empower your users with better IT scalability ↗ What is self-service password reset? ↗ Deploying self-service password reset ↗ How to enable and configure SSPR in Microsoft Entra ID ↗ How to configure self-service password reset for users in Microsoft Entra ID? ↗ How to [prepare users to] register [their] security information for Microsoft Entra ID ↗
Online courses	Managing Identities in Microsoft Entra ID ↗ Use SSPR to give your users a modern, protected experience. See especially the " Managing Microsoft Entra Users and Groups ↗ " module. Getting Started with the Microsoft Enterprise Mobility Suite ↗ Learn the best practices for extending on-premises assets to the cloud in a manner that allows for authentication, authorization, encryption, and a secured mobile experience. See especially the "Configuring Advanced Features of Microsoft Entra ID P1 or P2" module.
Tutorials	Complete a Microsoft Entra self-service password reset pilot roll out Enabling password writeback Microsoft Entra password reset from the login screen for Windows 10

Resources	Link and Description
FAQ	Password management frequently asked questions

Solution architecture

The following example describes the password reset solution architecture for common hybrid environments.



Description of workflow

To reset the password, users go to the [password reset portal](#). They must verify the previously registered authentication method or methods to prove their identity. If they successfully reset the password, they begin the reset process.

- For cloud-only users, SSPR stores the new password in Microsoft Entra ID.
- For hybrid users, SSPR writes back the password to the on-premises Active Directory via the Microsoft Entra Connect service.

ⓘ Note

For users who have [Password hash synchronization \(PHS\)](#) disabled, SSPR stores the passwords in the on-premises Active Directory only.

Best practices

You can help users register quickly by deploying SSPR alongside another popular application or service in the organization. This action generates a large volume of sign-ins and drives registration.

Before deploying SSPR, you may opt to determine the number and the average cost of each password reset call. You can use this data post deployment to show the value SSPR is bringing to the organization.

Combined registration for SSPR and Microsoft Entra multifactor authentication

SSPR allows users to reset their password in a secure way using the same methods they use for Microsoft Entra multifactor authentication. [Combined registration](#) is a single registration step for end users which enables registration of both MFA and SSPR methods at the same time. To make sure you understand the functionality and end-user experience, see the [Combined security information registration concepts](#).

It's critical to inform users about upcoming changes, registration requirements, and any necessary user actions. We provide [communication templates](#) and [user documentation](#) to prepare your users for the new experience and help to ensure a successful rollout. Send users to <https://myprofile.microsoft.com> to register by selecting the **Security Info** link on that page.

Plan the deployment project

Consider your organizational needs while you determine the strategy for this deployment in your environment.

Engage the right stakeholders

When technology projects fail, they typically do so due to mismatched expectations on impact, outcomes, and responsibilities. To avoid these pitfalls, [ensure that you are engaging the right stakeholders](#) and that stakeholder roles in the project are well understood by documenting the stakeholders and their project input and accountabilities.

Required administrator roles

[] [Expand table](#)

Business Role/Persona	Microsoft Entra role (if necessary)
Level 1 helpdesk	Password Administrator
Level 2 helpdesk	User Administrator
SSPR administrator	Authentication Administrator

Plan a pilot

We recommend that the initial configuration of SSPR is in a test environment. Start with a pilot group by enabling SSPR for a subset of users in your organization. See [Best practices for a pilot](#).

To create a group, see how to [create a group and add members in Microsoft Entra ID](#).

Plan configuration

The following settings are required to enable SSPR along with recommended values.

[\[+\] Expand table](#)

Area	Setting	Value
SSPR Properties	Self-service password reset enabled	Selected group for pilot / All for production
Authentication methods	Authentication methods required to register	Always 1 more than required for reset
	Authentication methods required to reset	One or two
Registration	Require users to register when signing in	Yes
	Number of days before users are asked to reconfirm their authentication information	90 – 180 days
Notifications	Notify users on password resets	Yes
	Notify all admins when other admins reset their password	Yes
Customization	Customize helpdesk link	Yes
	Custom helpdesk email or URL	Support site or email address

Area	Setting	Value
On-premises integration	Write back passwords to on-premises AD	Yes
	Allow users to unlock account without resetting password	Yes

SSPR properties

When enabling SSPR, choose an appropriate security group in the pilot environment.

- To enforce SSPR registration for everyone, we recommend using the **All** option.
- Otherwise, select the appropriate Microsoft Entra ID or AD security group.

Authentication methods

When SSPR is enabled, users can only reset their password if they have data present in the authentication methods that the administrator has enabled. Methods include phone, Authenticator app notification, security questions, and so on. For more information, see [What are authentication methods?](#).

We recommend the following authentication method settings:

- Set the **Authentication methods required to register** to at least one more than the number required to reset. Allowing multiple authentications gives users flexibility when they need to reset.
- Set **Number of methods required to reset** to a level appropriate to your organization. One requires the least friction, while two may increase your security posture.

Note: The user must have the authentication methods configured in the [Password policies and restrictions in Microsoft Entra ID](#).

Registration settings

Set **Require users to register when signing in** to **Yes**. This setting requires users to register when signing in, ensuring that all users are protected.

Set **Number of days before users is asked to reconfirm their authentication information** to between **90** and **180** days, unless your organization has a business need for a shorter time frame.

Notifications settings

Configure both the **Notify users on password resets** and the **Notify all admins when other admins reset their password to Yes**. Selecting **Yes** on both increases security by ensuring that users are aware when their password is reset. It also ensures that all admins are aware when an admin changes a password. If users or admins receive a notification and they haven't initiated the change, they can immediately report a potential security issue.

Note

Email notifications from the SSPR service are sent from the following addresses based on the Azure cloud you're working with:

- Public: msonlineserviceteam@microsoft.com
- China: msonlineserviceteam@oe.21vianet.com
- Government: msonlineserviceteam@azureadnotifications.us

If you observe issues in receiving notifications, check your spam settings.

Customization settings

It's critical to customize the helpdesk email or URL to ensure users who experience problems can get help immediately. Set this option to a common helpdesk email address or web page that your users are familiar with.

For more information, see [Customize the Microsoft Entra functionality for self-service password reset](#).

Password Writeback

Password Writeback is enabled with [Microsoft Entra Connect](#) and writes password resets in the cloud back to an existing on-premises directory in real time. For more information, see [What is Password Writeback?](#)

We recommend the following settings:

- Ensure that **Write back passwords to on-premises AD** is set to **Yes**.
- Set the **Allow users to unlock account without resetting password** to **Yes**.

By default, Microsoft Entra ID unlocks accounts when it performs a password reset.

Administrator password setting

Administrator accounts have elevated permissions. The on-premises enterprise or domain administrators can't reset their passwords through SSPR. On-premises admin accounts have the following restrictions:

- Can only change their password in their on-premises environment.
- Can never use the secret questions and answers as a method to reset their password.

We recommend that you don't sync your on-premises Active Directory admin accounts with Microsoft Entra ID.

Environments with multiple identity management systems

Some environments have multiple identity management systems. On-premises identity managers like Oracle IAM and SiteMinder, require synchronization with AD for passwords. You can do this using a tool like the Password Change Notification Service (PCNS) with Microsoft Identity Manager (MIM). To find information on this more complex scenario, see the article [Deploy the MIM Password Change Notification Service on a domain controller](#).

Plan Testing and Support

At each stage of your deployment from initial pilot groups through organization-wide, ensure that results are as expected.

Plan testing

To ensure that your deployment works as expected, plan a set of test cases to validate the implementation. To assess the test cases, you need a non-administrator test user with a password. If you need to create a user, see [Add new users to Microsoft Entra ID](#).

The following table includes useful test scenarios you can use to document your organization's expected results based on your policies.

[] [Expand table](#)

Business case	Expected results
SSPR portal is accessible from within the corporate network	Determined by your organization
SSPR portal is accessible from outside the corporate network	Determined by your organization
Reset user password from browser when user is not enabled for password reset	User is not able to access the password reset flow
Reset user password from browser when user has not registered for password reset	User is not able to access the password reset flow
User signs in when enforced to do password reset registration	Prompts the user to register security information
User signs in when password reset registration is complete	Prompts the user to register security information
SSPR portal is accessible when the user doesn't have a license	Is accessible
Reset user password from Windows 10 Microsoft Entra joined or Microsoft Entra hybrid joined device lock screen	User can reset password
SSPR registration and usage data are available to administrators in near real time	Is available via audit logs

You can also refer to [Complete out a Microsoft Entra self-service password reset pilot roll](#). In this tutorial, you enable a pilot roll out of SSPR in your organization and test using a non-administrator account.

Plan support

While SSPR doesn't typically create user issues, it's important to prepare support staff to deal with issues that may arise. To enable your support team's success, you can create an FAQ based on questions you receive from your users. Here are a few examples:

[] [Expand table](#)

Scenarios	Description
User doesn't have any registered authentication methods available	A user is trying to reset their password but doesn't have any of the authentication methods that they registered available (Example: they left their cell phone at home and can't access email)

Scenarios	Description
User isn't receiving a text or call on their office or cell phone	A user is trying to verify their identity via text or call but isn't receiving a text/call.
User can't access the password reset portal	A user wants to reset their password but isn't enabled for password reset and can't access the page to update passwords.
User can't set a new password	A user completes verification during the password reset flow but can't set a new password.
User doesn't see a Reset Password link on a Windows 10 device	A user is trying to reset password from the Windows 10 lock screen, but the device is either not joined to Microsoft Entra ID, or the Microsoft Intune device policy isn't enabled

Plan rollback

To roll back the deployment:

- For a single user, remove the user from the security group
- For a group, remove the group from SSPR configuration
- For everyone, disable SSPR for the Microsoft Entra tenant

Deploy SSPR

Before deploying, ensure that you have done the following:

1. Determined the appropriate [configuration settings](#).
2. Identified the users and groups for the [pilot](#) and production environments.
3. [Determined configuration settings](#) for registration and self-service.
4. [Configured password writeback](#) if you have a hybrid environment.

You're now ready to deploy SSPR!

See [Enable self-service password reset](#) for complete step-by-step directions on configuring the following areas.

1. [Authentication methods](#)
2. [Registration settings](#)

3. Notifications settings
4. Customization settings
5. On-premises integration

Enable SSPR in Windows

For machines running Windows 7, 8, 8.1, and 10 you can [enable users to reset their password at the Windows sign in screen](#)

Manage SSPR

Microsoft Entra ID can provide additional information on your SSPR performance through audits and reports.

Password management activity reports

You can use pre-built reports on Microsoft Entra admin center to measure the SSPR performance. If you're appropriately licensed, you can also create custom queries. For more information, see [Reporting options for Microsoft Entra password management](#).

 **Note**

You must opt-in for this data to be gathered for your organization. To opt in, you must visit the Reporting tab or the audit logs on the Microsoft Entra admin center at least once. Until then, the data doesn't collect for your organization.

Audit logs for registration and password reset are available for 30 days. If security auditing within your corporation requires longer retention, the logs need to be exported and consumed into a SIEM tool such as [Microsoft Sentinel](#), Splunk, or ArcSight.

The screenshot shows the Microsoft Azure portal interface. The left sidebar has a 'Manage' section with 'Properties', 'Authentication methods', 'Registration', 'Notifications', 'Customization', and 'On-premises integration'. Below that is an 'Activity' section with 'Audit logs' selected. Under 'TROUBLESHOOTING + SUPPORT', there are 'Troubleshoot' and 'New support request' options. The main content area is titled 'Activity Details: Audit log' and shows a table of audit logs. The table has columns for DATE, TARGET(S), and INITIATOR. Three rows are listed:

DATE	TARGET(S)	INITIATOR
10/24/2017 10:54:21 AM	User : Test@identityit	8130e
10/24/2017 10:53:41 AM	User : Test@identityit	8130e
10/24/2017 10:51:08 AM	User : Test@identityit	Test@

Below the table, there are sections for 'Activity', 'Activity Status', 'Initiated By (Actor)', 'Target(s)', 'Target', and 'Additional Details'. The 'Activity' section shows a date (10/24/2017 10:53:41 AM), name (User registered for self-service password reset), correlation ID, category (Self-service Password Management), and activity type (All). The 'Activity Status' section shows status (Success) and reason. The 'Initiated By (Actor)' section shows type (Other) and object ID. The 'Target(s)' section shows target type (User) and object ID. The 'Target' section shows target upn (Test@identityit). The 'Additional Details' section is empty.

Authentication methods - Usage and insights

Usage and insights enable you to understand how authentication methods for features like Microsoft Entra multifactor authentication and SSPR are working in your organization. This reporting capability provides your organization with the means to understand what methods register and how to use them.

Troubleshoot

- Refer to [Troubleshoot self-service password reset](#)
- Follow [Password management frequently asked questions](#)

Helpful documentation

- [What are authentication methods?](#)
- [How it works: Microsoft Entra self-service password reset?](#)
- [Customize the Microsoft Entra functionality for self-service password reset](#)
- [Password policies and restrictions in Microsoft Entra ID](#)
- [What is Password Writeback?](#)

Next steps

- To get started deploying SSPR, see [Enable Microsoft Entra self-service password reset](#)
 - Consider implementing Microsoft Entra password protection
 - Consider implementing Microsoft Entra Smart Lockout
-

Feedback

Was this page helpful?



Yes



No

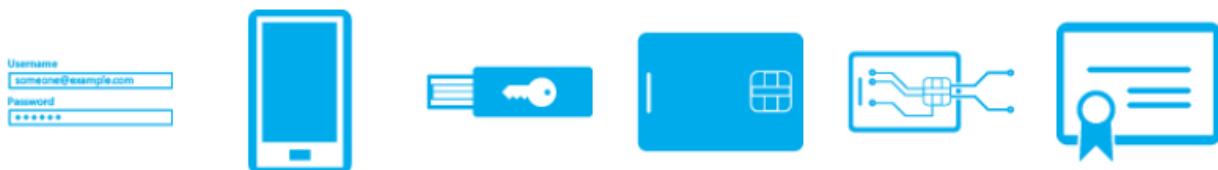
[Provide product feedback](#) ↗

How it works: Microsoft Entra multifactor authentication

Article • 03/04/2025

Multifactor authentication is a process in which users are prompted during the sign-in process for an additional form of identification, such as a code on their cellphone or a fingerprint scan.

If you only use a password to authenticate a user, it leaves an insecure vector for attack. If the password is weak or has been exposed elsewhere, an attacker could be using it to gain access. When you require a second form of authentication, security is increased because this additional factor isn't something that's easy for an attacker to obtain or duplicate.



Microsoft Entra multifactor authentication works by requiring two or more of the following authentication methods:

- Something you know, typically a password.
- Something you have, such as a trusted device that's not easily duplicated, like a phone or hardware key.
- Something you are - biometrics like a fingerprint or face scan.

Microsoft Entra multifactor authentication can also further secure password reset. When users register themselves for Microsoft Entra multifactor authentication, they can also register for self-service password reset in one step. Administrators can choose forms of secondary authentication and configure challenges for MFA based on configuration decisions.

You don't need to change apps and services to use Microsoft Entra multifactor authentication. The verification prompts are part of the Microsoft Entra sign-in, which automatically requests and processes the MFA challenge when needed.

ⓘ Note

The prompt language is determined by browser locale settings. If you use custom greetings but don't have one for the language identified in the browser locale,

English is used by default. Network Policy Server (NPS) will always use English by default, regardless of custom greetings. English is also used by default if the browser locale can't be identified.

The screenshot shows a Microsoft login page with a purple header. The main content area has a light gray background. At the top, it says "Keep your account secure" and "Your organization requires you to set up the following methods of proving who you are." Below this, a white box contains the "Microsoft Authenticator" setup instructions. It features a blue lock icon and the text "Start by getting the app". It explains that users need to install the Microsoft Authenticator app on their phone and choose "Next" after installation. There is also a link "I want to use a different authenticator app". A blue "Next" button is at the bottom right of the box. At the very bottom of the page, there is a link "I want to set up a different method".

Available verification methods

When users sign in to an application or service and receive an MFA prompt, they can choose from one of their registered forms of additional verification. Users can access [My Profile](#) to edit or add verification methods.

The following additional forms of verification can be used with Microsoft Entra multifactor authentication:

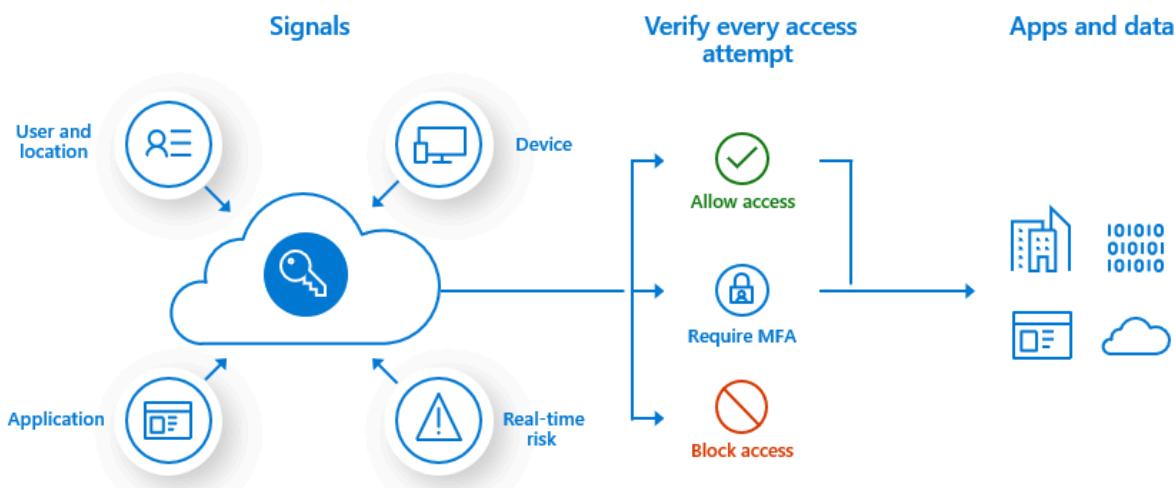
- Microsoft Authenticator
- Authenticator Lite (in Outlook)
- Windows Hello for Business
- Passkey (FIDO2)
- Passkey in Microsoft Authenticator (preview)
- Certificate-based authentication (when configured for multifactor authentication)
- External authentication methods (preview)
- Temporary Access Pass (TAP)
- OATH hardware token (preview)
- OATH software token
- SMS

- Voice call

How to enable and use Microsoft Entra multifactor authentication

You can use [security defaults](#) in Microsoft Entra tenants to quickly enable Microsoft Authenticator for all users. You can enable Microsoft Entra multifactor authentication to prompt users and groups for additional verification during sign-in.

For more granular controls, you can use [Conditional Access](#) policies to define events or applications that require MFA. These policies can allow regular sign-in when the user is on the corporate network or a registered device but prompt for additional verification factors when the user is remote or on a personal device.



Related content

To learn more about different authentication and validation methods, see [Authentication methods in Microsoft Entra ID](#).

To see MFA in action, enable Microsoft Entra multifactor authentication for a set of test users in the following tutorial:

[Enable Microsoft Entra multifactor authentication](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Protecting authentication methods in Microsoft Entra ID

Article • 04/29/2025

(!) Note

The Microsoft managed value for Authenticator Lite will move from disabled to enabled on June 26th, 2023. All tenants left in the default state **Microsoft managed** will be enabled for the feature on June 26th.

Microsoft Entra ID adds and improves security features to better protect customers against increasing attacks. As new attack vectors become known, Microsoft Entra ID can respond by enabling protection by default to help customers stay ahead of emerging security threats.

For example, in response to increasing MFA fatigue attacks, Microsoft recommended ways for customers to [defend users](#). One recommendation to prevent users from accidental multifactor authentication (MFA) approvals is to enable [number matching](#). As a result, default behavior for number matching will be explicitly **Enabled** for all Microsoft Authenticator users. You can learn more about new security features like number matching in our blog post [Advanced Microsoft Authenticator security features are now generally available!](#).

There are two ways for protection of a security feature to be enabled by default:

- After a security feature is released, customers can use the Microsoft Entra admin center or Graph API to test and roll out the change on their own schedule. To help defend against new attack vectors, Microsoft Entra ID can enable protection of a security feature by default for all tenants on a certain date, and there won't be an option to disable protection. Microsoft schedules default protection far in advance to give customers time to prepare for the change. Customers can't opt out if Microsoft schedules protection by default.
- Protection can be **Microsoft managed**, which means Microsoft Entra ID can enable or disable protection based upon the current landscape of security threats. Customers can choose whether to allow Microsoft to manage the protection. They can change from **Microsoft managed** to explicitly make the protection **Enabled** or **Disabled** at any time.

(!) Note

Only a critical security feature will have protection enabled by default.

Default protection enabled by Microsoft Entra ID

Number matching is a good example of protection for an authentication method that is currently optional for push notifications in Microsoft Authenticator in all tenants. Customers could choose to enable number matching for push notifications in Microsoft Authenticator for users and groups, or they could leave it disabled. Number matching is already the default behavior for passwordless notifications in Microsoft Authenticator, and users can't opt out.

As MFA fatigue attacks rise, number matching becomes more critical to sign-in security. As a result, Microsoft will change the default behavior for push notifications in Microsoft Authenticator.

Microsoft managed settings

In addition to configuring Authentication methods policy settings to be either **Enabled** or **Disabled**, IT admins can configure some settings in the Authentication methods policy to be **Microsoft managed**. A setting that is configured as **Microsoft managed** allows Microsoft Entra ID to enable or disable the setting.

The option to let Microsoft Entra ID manage the setting is a convenient way for an organization to allow Microsoft to enable or disable a feature by default. Organizations can more easily improve their security posture by trusting Microsoft to manage when a feature should be enabled by default. By configuring a setting as **Microsoft managed** (named *default* in Graph APIs), IT admins can trust Microsoft to enable a security feature they haven't explicitly disabled.

For example, an admin can enable [location and application name](#) in push notifications to give users more context when they approve MFA requests with Microsoft Authenticator. The additional context can also be explicitly disabled, or set as **Microsoft managed**. Today, the **Microsoft managed** configuration for location and application name is **Disabled**, which effectively disables the option for any environment where an admin chooses to let Microsoft Entra ID manage the setting.

As the security threat landscape changes over time, Microsoft can change the **Microsoft managed** configuration for location and application name to **Enabled**. For customers who want to rely upon Microsoft to improve their security posture, setting security features to **Microsoft managed** is an easy way stay ahead of security threats. They can trust Microsoft to determine the best way to configure security settings based on the current threat landscape.

The following table lists each setting that can be set to Microsoft managed and whether that setting is enabled or disabled by default.

Setting	Configuration
Registration campaign	Enabled for text message and voice call users
Location in Microsoft Authenticator notifications	Disabled
Application name in Microsoft Authenticator notifications	Disabled
System-preferred MFA	Enabled
Authenticator Lite	Enabled
Report suspicious activity	Disabled

As threat vectors change, Microsoft Entra ID can announce default protection for a **Microsoft managed** setting in [release notes](#) and on commonly read forums like [Tech Community](#).

For more information, see our blog post [It's Time to Hang Up on Phone Transports for Authentication](#) which discusses moving away from using text message and voice calls. This change leads to default enablement for the registration campaign to help users set up Authenticator for modern authentication.

Next steps

[Authentication methods in Microsoft Entra ID - Microsoft Authenticator](#)

System-preferred multifactor authentication - Authentication methods policy

Article • 03/19/2025

System-preferred multifactor authentication (MFA) prompts users to sign in by using the most secure method they registered. It's an important security enhancement for users who authenticate by using telecom transports. Administrators can enable system-preferred MFA to improve sign-in security and discourage less secure sign-in methods like Short Message Service (SMS).

For example, if a user registered both SMS and Microsoft Authenticator push notifications as methods for MFA, system-preferred MFA prompts the user to sign in by using the more secure push notification method. The user can still choose to sign in by using another method, but they're first prompted to try the most secure method they registered.

System-preferred MFA is a Microsoft managed setting, which is a [tristate policy](#). The **Microsoft managed** value of system-preferred MFA is **Enabled**. If you don't want to enable system-preferred MFA, change the state from **Microsoft managed** to **Disabled**, or exclude users and groups from the policy.

After system-preferred MFA is enabled, the authentication system does all the work. Users don't need to set any authentication method as their default because the system always determines and presents the most secure method they registered.

Enable system-preferred MFA in the Microsoft Entra admin center

By default, system-preferred MFA is Microsoft managed and disabled for all users.

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Settings**.
3. For **System-preferred multifactor authentication**, choose whether to explicitly enable or disable the feature, and include or exclude any users. Excluded groups take precedence over include groups.

For example, the following screenshot shows how to make system-preferred MFA explicitly enabled for only the Engineering group.

The screenshot shows a configuration interface for 'System-preferred multifactor authentication'. At the top, a descriptive text states: 'This setting designates whether the Microsoft-managed most secure multifactor authentication method is served to users.' Below this is a 'Learn more' link. The 'State' dropdown is set to 'Enabled'. Under the 'Include' tab, the 'Target' section is configured to target the 'Engineering' group, with the 'Select group' option selected. At the bottom are 'Save' and 'Discard' buttons.

4. After you finish making any changes, click **Save**.

Enable system-preferred MFA using Graph APIs

To enable system-preferred MFA in advance, you need to choose a single target group for the schema configuration, as shown in the [Request](#) example.

Authentication method feature configuration properties

By default, system-preferred MFA is [Microsoft managed](#) and enabled.

[] [Expand table](#)

Property	Type	Description
excludeTarget	featureTarget	A single entity that is excluded from this feature. You can only exclude one group from system-preferred MFA, which can be a dynamic or nested group.
includeTarget	featureTarget	A single entity that is included in this feature. You can only include one group for system-preferred MFA, which can be a dynamic or nested group.
State	advancedConfigState	Possible values are: enabled explicitly enables the feature for the selected group. disabled explicitly disables the feature for the selected group.

Property	Type	Description
		default allows Microsoft Entra ID to manage whether the feature is enabled or not for the selected group.

Feature target properties

System-preferred MFA can be enabled only for a single group, which can be a dynamic or nested group.

[Expand table](#)

Property	Type	Description
ID	String	ID of the entity targeted.
targetType	featureTargetType	The kind of entity targeted, such as group, role, or administrative unit. The possible values are: 'group', 'administrativeUnit', 'role', 'unknownFutureValue'.

Use the following API endpoint to enable **systemCredentialPreferences** and include or exclude groups:

```
https://graph.microsoft.com/v1.0/policies/authenticationMethodsPolicy
```

ⓘ Note

In Graph Explorer, you need to consent to the **Policy.ReadWrite.AuthenticationMethod** permission.

Request

The following example excludes a sample target group and includes all users. For more information, see [Update authenticationMethodsPolicy](#).

HTTP

```
PATCH https://graph.microsoft.com/v1.0/policies/authenticationMethodsPolicy
Content-Type: application/json

{
  "systemCredentialPreferences": {
```

```
        "state": "enabled",
        "excludeTargets": [
            {
                "id": "d1411007-6fcf-4b4c-8d70-1da1857ed33c",
                "targetType": "group"
            }
        ],
        "includeTargets": [
            {
                "id": "all_users",
                "targetType": "group"
            }
        ]
    }
}
```

FAQ

How does system-preferred MFA determine the most secure method?

When a user signs in, the authentication process checks which authentication methods are registered for the user. The user is prompted to sign-in with the most secure method according to the following order. The order of authentication methods is dynamic. It's updated as the security landscape changes, and as better authentication methods emerge. Due to known issues with certificate-based authentication (CBA) and system-preferred MFA, we moved CBA to the bottom of the list. Click the link for more information about each method.

1. [Temporary Access Pass](#)
2. [Passkey \(FIDO2\)](#)
3. [External authentication methods](#)
4. [Microsoft Authenticator notifications](#)
5. [Time-based one-time password \(TOTP\)¹](#)
6. [Telephony²](#)
7. [Certificate-based authentication](#)

¹Includes hardware or software TOTP from Microsoft Authenticator, Authenticator Lite, or third-party applications.

²Includes SMS and voice calls.

How does system-preferred MFA affect the NPS extension?

System-preferred MFA doesn't affect users who sign in by using the Network Policy Server (NPS) extension. Those users don't see any change to their sign-in experience.

What happens for users who aren't specified in the Authentication methods policy but enabled in the legacy MFA tenant-wide policy?

The system-preferred MFA also applies for users who are enabled for MFA in the legacy MFA policy.

verification options ([learn more](#))

Methods available to users:

Call to phone
 Text message to phone
 Notification through mobile app
 Verification code from mobile app or hardware token

Next steps

- [Authentication methods in Microsoft Entra ID](#)
- [How to run a registration campaign to set up Microsoft Authenticator](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Planning for mandatory multifactor authentication for Azure and other admin portals

Article • 04/25/2025

At Microsoft, we're committed to providing our customers with the highest level of security. One of the most effective security measures available to them is multifactor authentication (MFA). [Research by Microsoft](#) shows that MFA can block more than 99.2% of account compromise attacks.

That's why, starting in 2024, we'll enforce mandatory MFA for all Azure sign-in attempts. For more background about this requirement, see our [blog post](#). This topic covers which applications and accounts are affected, how enforcement gets rolled out to tenants, and other common questions and answers.

Important

If a user can't sign in to Azure and other admin portals after rollout of mandatory MFA, a Global Administrator can run a script to postpone the MFA requirement and allow users to sign in. For more information, see [How to postpone enforcement for a tenant where users are unable to sign in after rollout of mandatory multifactor authentication \(MFA\) requirement for the the Azure portal, Microsoft Entra admin center, or Microsoft Intune admin center](#).

There's no change for users if your organization already enforces MFA for them, or if they sign in with stronger methods like passwordless or passkey (FIDO2). To verify that MFA is enabled, see [How to verify that users are set up for mandatory MFA](#).

Scope of enforcement

The scope of enforcement includes which applications plan to enforce MFA, applications that are out of scope, when enforcement is planned to occur, and which accounts have a mandatory MFA requirement.

Applications

Note

The date of enforcement for Phase 2 has changed to July 1, 2025.

The following table lists affected apps, app IDs, and URLs for Azure.

[+] Expand table

Application Name	App ID	Enforcement starts
Azure portal	c44b4083-3bb0-49c1-b47d-974e53cbdf3c	Second half of 2024
Microsoft Entra admin center ↗	c44b4083-3bb0-49c1-b47d-974e53cbdf3c	Second half of 2024
Microsoft Intune admin center ↗	c44b4083-3bb0-49c1-b47d-974e53cbdf3c	Second half of 2024
Azure command-line interface (Azure CLI)	04b07795-8ddb-461a-bbee-02f9e1bf7b46	July 1, 2025
Azure PowerShell	1950a258-227b-4e31-a9cf-717495945fc2	July 1, 2025
Azure mobile app	0c1307d4-29d6-4389-a11c-5cbe7f65d7fa	July 1, 2025
Infrastructure as Code (IaC) tools	Use Azure CLI or Azure PowerShell IDs	July 1, 2025

The following table lists affected apps and URLs for Microsoft 365.

[+] Expand table

Application Name	URL	Enforcement starts
Microsoft 365 admin center	https://portal.office.com/adminportal/home	February 2025
Microsoft 365 admin center	https://admin.cloud.microsoft	February 2025
Microsoft 365 admin center	https://admin.microsoft.com	February 2025

Accounts

All users who sign into the [applications](#) listed earlier to perform any Create, Read, Update, or Delete (CRUD) operation must complete MFA when the enforcement begins. Users aren't required to use MFA if they access other applications, websites, or services hosted on Azure.

Each application, website, or service owner listed earlier controls the authentication requirements for users.

Break glass or emergency access accounts are also required to sign in with MFA once enforcement begins. We recommend that you update these accounts to use [passkey \(FIDO2\)](#) or configure [certificate-based authentication](#) for MFA. Both methods satisfy the MFA requirement.

Workload identities, such as managed identities and service principals, aren't impacted by [either phase](#) of this MFA enforcement. If user identities are used to sign in as a service account to run automation (including scripts or other automated tasks), those user identities need to sign in with MFA once enforcement begins. User identities aren't recommended for automation. You should migrate those user identities to [workload identities](#).

Client libraries

The OAuth 2.0 Resource Owner Password Credentials (ROPC) token grant flow is incompatible with MFA. After MFA is enabled in your Microsoft Entra tenant, ROPC-based APIs used in your applications throw exceptions. For more information about how to migrate from ROPC-based APIs in [Microsoft Authentication Libraries \(MSAL\)](#), see [How to migrate away from ROPC](#). For language-specific MSAL guidance, see the following tabs.

.NET

Changes are required if you use the [Microsoft.Identity.Client](#) package and one of the following APIs in your application:

- [IByUsernameAndPassword.AcquireTokenByUsernamePassword](#) (confidential client API)
- [PublicClientApplication.AcquireTokenByUsernamePassword](#) (public client API)

The same general MSAL guidance applies to the Azure Identity libraries. The `UsernamePasswordCredential` class provided in those libraries uses MSAL ROPC-based APIs. For language-specific guidance, see the following tabs.

.NET

Changes are required if you use the [Azure.Identity](#) package and do one of the following things in your application:

- Use `DefaultAzureCredential` or `EnvironmentCredential` with the following two environment variables set:
 - `AZURE_USERNAME`
 - `AZURE_PASSWORD`
- Using `UsernamePasswordCredential` ([deprecated as of the 1.14.0-beta.2 release ↗](#))

Migrate user-based service accounts to workload identities

We recommend customers discover user accounts that are used as service accounts begin to migrate them to workload identities. Migration often requires updating scripts and automation processes to use workload identities.

Review [How to verify that users are set up for mandatory MFA](#) to identify all user accounts, including user accounts being used as service accounts, that sign in to the [applications](#).

For more information about how to migrate from user-based service accounts to workload identities for authentication with these applications, see:

- [Sign into Azure with a managed identity using the Azure CLI](#)
- [Sign into Azure with a service principal using the Azure CLI](#)
- [Sign in to Azure PowerShell non-interactively for automation scenarios](#) includes guidance for both managed identity and service principal use cases

Some customers apply Conditional Access policies to user-based service accounts. You can reclaim the user-based license, and add a [workload identities](#) license to apply [Conditional Access for workload identities](#).

Implementation

This requirement for MFA at sign-in is implemented for admin portals and other [applications](#). Microsoft Entra ID [sign-in logs](#) shows it as the source of the MFA requirement.

Mandatory MFA isn't configurable. It's implemented separately from any access policies configured in the tenant.

For example, if your organization chose to retain Microsoft's [security defaults](#), and you currently have security defaults enabled, your users don't see any changes as MFA is already required for Azure management. If your tenant is using [Conditional Access](#) policies in Microsoft Entra and you already have a Conditional Access policy through which users sign into Azure with MFA, then your users don't see a change. Similarly, any restrictive Conditional Access

policies that target Azure and require stronger authentication, such as phishing-resistant MFA, continue to be enforced. Users don't see any changes.

Enforcement phases

 Note

The date of enforcement for Phase 2 has changed to July 1, 2025.

The enforcement of MFA rolls out in two phases:

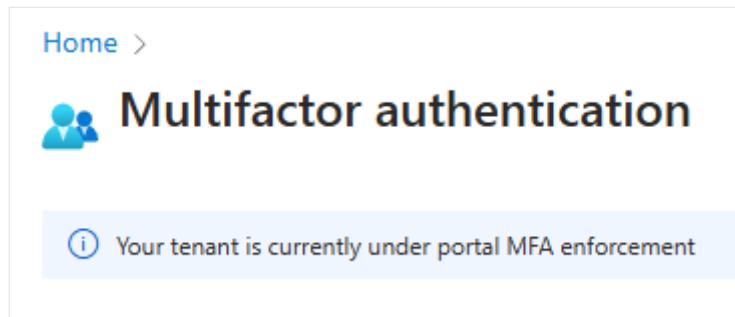
- **Phase 1:** Starting in October 2024, MFA is required to sign in to the Azure portal, Microsoft Entra admin center, and Microsoft Intune admin center. The enforcement will gradually roll out to all tenants worldwide. Starting in February 2025, MFA enforcement gradually begins for sign in to Microsoft 365 admin center. This phase won't impact other Azure clients such as Azure CLI, Azure PowerShell, Azure mobile app, or IaC tools.
- **Phase 2:** Starting July 1, 2025, MFA enforcement will gradually begin for Azure CLI, Azure PowerShell, Azure mobile app, IaC tools, and REST API endpoints. Some customers may use a user account in Microsoft Entra ID as a service account. It's recommended to migrate these user-based service accounts to [secure cloud based service accounts with workload identities](#).

Notification channels

Microsoft will notify all Microsoft Entra Global Administrators through the following channels:

- Email: Global Administrators who configured an email address will be informed by email of the upcoming MFA enforcement and the actions required to be prepared.
- Service health notification: Global Administrators receive a service health notification through the Azure portal, with the tracking ID of **4V20-VX0**. This notification contains the same information as the email. Global Administrators can also subscribe to receive service health notifications through email.
- Portal notification: A notification appears in the Azure portal, Microsoft Entra admin center, and Microsoft Intune admin center when they sign in. The portal notification links to this topic for more information about the mandatory MFA enforcement.
- Microsoft 365 message center: A message appears in the Microsoft 365 message center with message ID: **MC862873**. This message has the same information as the email and service health notification.

After enforcement, a banner appears in the [Azure portal](#):



The screenshot shows the Azure portal's Multifactor authentication page. At the top left is a 'Home' link. Below it is a section titled 'Multifactor authentication' with a blue icon of two people. A light blue banner at the bottom of this section contains the text 'Your tenant is currently under portal MFA enforcement' next to an information icon.

External authentication methods and identity providers

Support for external MFA solutions is in preview with [external authentication methods](#), and can be used to meet the MFA requirement. The legacy Conditional Access custom controls preview doesn't satisfy the MFA requirement. You should migrate to the external authentication methods preview to use an external solution with Microsoft Entra ID.

If you're using a federated Identity Provider (IdP), such as Active Directory Federation Services, and your MFA provider is integrated directly with this federated IdP, the federated IdP must be configured to send an MFA claim. For more information, see [Expected inbound assertions for Microsoft Entra MFA](#).

Request more time to prepare for enforcement

We understand that some customers may need more time to prepare for this MFA requirement. Microsoft is allowing customers with complex environments or technical barriers to postpone the enforcement for their tenants until September 30, 2025.

Global Administrators can go to the <https://aka.ms/managemfaforazure> to select the start date of enforcement for their tenant for admin portals in Phase 1. If you postponed the start date for Phase 1, Phase 2 enforcement *doesn't* begin before the start date you choose. Global Administrators must [elevate access](#) and use MFA before they can postpone the start date of MFA enforcement.

Global Administrators must perform this action for every tenant where they want to postpone the start date of enforcement.

By postponing the start date of enforcement, you take extra risk because accounts that access Microsoft services like the Azure portal are highly valuable targets for threat actors. We recommend all tenants set up MFA now to secure cloud resources.

FAQs

Question: If the tenant is only used for testing, is MFA required?

Answer: Yes, every Azure tenant will require MFA, with no exception for test environments.

Question: How does this requirement impact the Microsoft 365 admin center?

Answer: Mandatory MFA will roll out to the Microsoft 365 admin center starting in February 2025. Learn more about the mandatory MFA requirement for the Microsoft 365 admin center on the blog post [Announcing mandatory multifactor authentication for the Microsoft 365 admin center ↗](#).

Question: Is MFA mandatory for all users or only administrators?

Answer: All users who sign in to any of the [applications](#) listed previously are required to complete MFA, regardless of any administrator roles that are activated or eligible for them, or any [user exclusions](#) that are enabled for them.

Question: Do I need to complete MFA if I choose the option to **Stay signed in**?

Answer: Yes, even if you choose **Stay signed in**, you're required to complete MFA before you can sign in to these [applications](#).

Question: Does the enforcement apply to B2B guest accounts?

Answer: Yes, MFA has to be adhered either from the partner resource tenant, or the user's home tenant if it's set up properly to send MFA claims to the resource tenant by using cross-tenant access.

Question: Does the enforcement apply to Azure for US Government or Azure sovereign clouds?

Answer: Microsoft enforces mandatory MFA only in the public Azure cloud. Microsoft doesn't currently enforce MFA in Azure for US Government or other Azure sovereign clouds.

Question: How can we comply if we enforce MFA by using another identity provider or MFA solution, and we don't enforce by using Microsoft Entra MFA?

Answer: Third-party MFA can be integrated directly with Microsoft Entra ID. For more information, see [Microsoft Entra multifactor authentication external method provider reference](#). Microsoft Entra ID can be optionally configured with a federated Identity provider. If so, the identity provider solution needs to be configured properly to send the multipleauthn claim to Microsoft Entra ID. For more information, see [Satisfy Microsoft Entra ID multifactor authentication \(MFA\) controls with MFA claims from a federated IdP](#).

Question: Will mandatory MFA impact my ability to sync with Microsoft Entra Connect or Microsoft Entra Cloud Sync?

Answer: No. The synchronization service account isn't affected by the mandatory MFA requirement. Only [applications](#) listed earlier require MFA for sign in.

Question: Will I be able to opt out?

Answer: There's no way to opt out. This security motion is critical to all safety and security of the Azure platform and is being repeated across cloud vendors. For example, see [Secure by Design: AWS to enhance MFA requirements in 2024](#).

An option to postpone the enforcement start date is available for customers. Global Administrators can go to the [Azure portal](#) to postpone the start date of enforcement for their tenant. Global Administrators must have [elevated access](#) before they postpone the start date of MFA enforcement on this page. They must perform this action for each tenant that needs postponement.

Question: Can I test MFA before Azure enforces the policy to ensure nothing breaks?

Answer: Yes, you can [test their MFA](#) through the manual setup process for MFA. We encourage you to set this up and test. If you use Conditional Access to enforce MFA, you can use Conditional Access templates to test your policy. For more information, see [Require multifactor authentication for admins accessing Microsoft admin portals](#). If you run a free edition of Microsoft Entra ID, you can enable [security defaults](#).

Question: What if I already have MFA enabled, what happens next?

Answer: Customers that already require MFA for their users who access the applications listed earlier don't see any change. If you only require MFA for a subset of users, then any users not already using MFA will now need to use MFA when they sign in to the applications.

Question: How can I review MFA activity in Microsoft Entra ID?

Answer: To review details about when a user is prompted to sign in with MFA, use the Microsoft Entra sign-in logs. For more information, see [Sign-in event details for Microsoft Entra multifactor authentication](#).

Question: What if I have a "break glass" scenario?

Answer: We recommend updating these accounts to use [passkey \(FIDO2\)](#) or configure [certificate-based authentication](#) for MFA. Both methods satisfy the MFA requirement.

Question: What if I don't receive an email about enabling MFA before it was enforced, and then I get locked-out. How should I resolve it?

Answer: Users shouldn't be locked out, but they may get a message that prompts them to enable MFA once enforcement for their tenant has started. If the user is locked out, there may be other issues. For more information, see [Account has been locked](#).

Related content

Review the following topics to learn more about how to configure and deploy MFA:

- [How to postpone enforcement for a tenant where users are unable to sign in after rollout of mandatory multifactor authentication \(MFA\) requirement for the Azure portal, Microsoft Entra admin center, or Microsoft Intune admin center](#)
- [How to verify that users are set up for mandatory MFA](#)
- [Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication](#)
- [Secure sign-in events with Microsoft Entra multifactor](#)
- [Plan a Microsoft Entra multifactor authentication deployment](#)
- [Phishing-resistant MFA methods](#)
- [Microsoft Entra multifactor authentication](#)
- [Authentication methods](#)

Microsoft Entra multifactor authentication external method provider reference (Preview)

Article • 04/17/2025

This topic describes how an external authentication provider connects to Microsoft Entra multifactor authentication (MFA). An external authentication provider can integrate with Microsoft Entra ID tenants as an external authentication method (EAM). An EAM can satisfy the second factor of an MFA requirement for access to a resource or application. EAMs require at least a Microsoft Entra ID P1 license.

When a user signs in, that tenant policies are evaluated. The authentication requirements are determined based on the resource that the user tries to access.

Multiple policies may apply to the sign-in, depending on their parameters. Those parameters include users and groups, applications, platform, sign-in risk level, and more.

Based on the authentication requirements, the user may need to sign in with another factor to meet the MFA requirement. The second factor needs to complement the type of first factor.

EAMs are added to Microsoft Entra ID by the tenant admin. If a tenant requires an EAM for MFA, the sign-in is considered to meet the MFA requirement after Microsoft Entra ID validates both:

- The first factor completed with Microsoft Entra ID
- The second factor completed with the EAM

That validation meets the MFA requirement for two or more types of methods from:

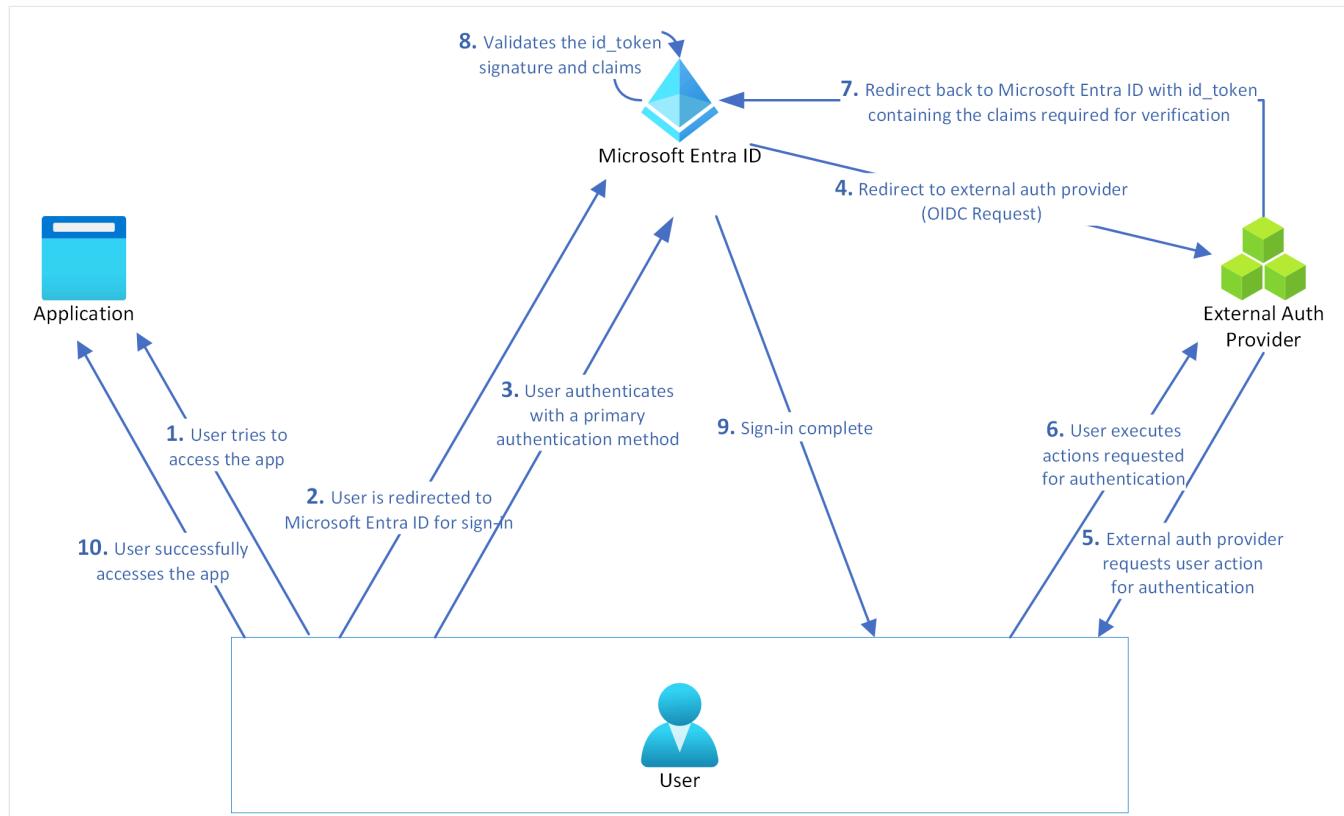
- Something you know (knowledge)
- Something you have (possession)
- Something you are (inherence)

EAMs are implemented on top of Open ID Connect (OIDC). This implementation requires at least three publicly facing endpoints:

- An OIDC Discovery endpoint, as described in [Discovery of provider metadata](#)
- A valid OIDC authentication endpoint
- A URL where the public certificates of the provider are published

Let's look closer at how sign-in works with an EAM:

1. A user tries to sign in with a first factor, like a password, to an application protected by Microsoft Entra ID.
2. Microsoft Entra ID determines that another factor needs to be satisfied. For example, a Conditional Access policy requires MFA.
3. The user chooses the EAM as a second factor.
4. Microsoft Entra ID redirects the user's browser session to the EAM URL:
 - a. This URL is discovered from the discovery URL provisioned by an admin when they created the EAM.
 - b. The application provides an expired or nearly expired token that contains information to identify the user and tenant.
5. The external authentication provider validates that the token came from Microsoft Entra ID, and checks the contents of the token.
6. The external authentication provider might optionally make a call to Microsoft Graph to fetch additional information about the user.
7. The external authentication provider performs any actions it deems necessary, such as authenticating the user with some credential.
8. The external authentication provider redirects the user back to Microsoft Entra ID with a valid token, including all required claims.
9. Microsoft Entra ID validates that the token's signature came from the configured external authentication provider, and then checks the contents of the token.
10. Microsoft Entra ID validates the token against the requirements.
11. The user satisfied the MFA requirement if the validation succeeds. The user might also have to meet other policy requirements.



Configure a new external authentication provider with Microsoft Entra ID

An application representing the integration is required for EAMs to issue the id_token_hint. The application can be created in two ways:

- Created in each tenant that uses the external provider.
- Created as one multitenant application. Privileged Role Administrators need to grant consent to enable the integration for their tenant.

A multitenant application reduces the chance of misconfiguration in each tenant. It also lets providers make changes to metadata like reply URLs in one place, rather than require each tenant to make the changes.

To configure a multitenant application, the provider admin must first:

1. Create an Microsoft Entra ID tenant if they don't have one yet.
2. Register an application in their tenant.
3. Set the Supported Account types of the application to: Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant).
4. Add the delegated permission `openid` and `profile` of Microsoft Graph to the application.
5. Don't publish any scopes in this application.
6. Add the external identity provider's valid authorization_endpoint URLs to that application as Reply URLs.

Note

The authorization_endpoint provided in the provider's discovery document should be added as a redirect url in the application registration. Otherwise, you get the following error: *ENTRA IDSTS50161: Failed to validate authorization url of external claims provider!*

The application registration process creates an application with several properties. These properties are required for our scenario.

 Expand table

Property	Description
Object ID	The provider can use the object ID with Microsoft Graph to query the application information. The provider can use the object ID to programmatically retrieve and edit the application information.
Application ID	The provider can use the application ID as the ClientId of their application.
Home page URL	The provider home page URL isn't used for anything, but is required as part of application registration.
Reply URLs	Valid redirect URLs for the provider. One should match the provider host URL that was set for the provider's tenant. One of the reply URLs registered must match the prefix of the authorization_endpoint that Microsoft Entra ID retrieves through OIDC discovery for the host url.

An application for each tenant is also a valid model to support the integration. If you use a single-tenant registration, the tenant admin needs to create an application registration with the properties in the preceding table for a single-tenant application.

 **Note**

Admin consent for the application is required in the tenant that uses the EAM. If consent isn't granted, the following error appears when an admin tries to use the EAM:
AADSTS900491: Service principal <your App ID> not found.

Configure optional claims

A provider can configure more claims by using [optional claims for id_token](#).

 **Note**

Regardless of how the application is created, the provider needs to configure optional claims for each cloud environment. If a multitenant application is used for global Azure and Azure for US Government, each cloud environment requires a different application and application ID.

Add an EAM to Microsoft Entra ID

External identity provider information is stored in the Authentication methods policy of each tenant. The provider information is stored as an authentication method of externalAuthenticationMethodConfiguration type.

Each provider has one entry in the list object of the policy. Each entry needs to state:

- If the method is enabled
- The included groups that can use the method
- The excluded groups that can't use the method

Conditional Access Administrators can create a policy with the Require MFA Grant to set the MFA requirement for user sign-in. External authentication methods aren't currently supported with authentication strengths.

For more information about how to add an external authentication method in the Microsoft Entra admin center, see [Manage an external authentication method in Microsoft Entra ID \(Preview\)](#).

Microsoft Entra ID interaction with provider

The next sections explain provider requirements and include examples for Microsoft Entra ID interaction with a provider.

Discovery of provider metadata

An external identity provider needs to provide an [OIDC Discovery endpoint](#). This endpoint is used to get more configuration data. The *full URL*, including `.well-known/oidc-configuration`, must be included in the Discovery URL configured when the EAM is created.

The endpoint returns a Provider Metadata [JSON document](#) hosted there. The endpoint must also return the valid content-length header.

The following table lists the data that should be present in the metadata of the provider. These values are required for this extensibility scenario. The JSON metadata document may contain more information.

For the OIDC document with the values for Provider Metadata, see [Provider Metadata](#).

 [Expand table](#)

Metadata value	Value	Comments
Issuer		This URL should match both the host URL used for discovery and the iss claim in the tokens issued by

Metadata value	Value	Comments
		the provider's service.
authorization_endpoint		The endpoint that Microsoft Entra ID communicates with for authorization. This endpoint must be present as one of the reply URLs for the allowed applications.
jwks_uri		Where Microsoft Entra ID can find the public keys needed to verify the signatures issued by the provider. [!NOTE] The JSON Web Key (JWK) <code>x5c</code> parameter must be present to provide X.509 representations of keys provided.
scopes_supported	openid	Other values may also be included but aren't required.
response_types_supported	id_token	Other values may also be included but aren't required.
subject_types_supported		
id_token_signing_alg_values_supported		Microsoft supports RS256
claim_types_supported	normal	This property is optional but if present, it should include the normal value; other values may also be included.

JSON

```
http://customcaserver.azurewebsites.net/v2.0/.well-known/openid-configuration
{
  "authorization_endpoint": "https://customcaserver.azurewebsites.net/api/Authorize",
  "claims_supported": [
    "email"
  ],
  "grant_types_supported": [
    "implicit"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "issuer": "https://customcaserver.azurewebsites.net",
  "jwks_uri": "http://customcaserver.azurewebsites.net/.well-known/jwks",
  "response_modes_supported": [
    "form_post"
  ],
  "response_types_supported": [
    "code"
  ]
}
```

```

    "id_token",
],
"scopes_supported": [
    "openid"
],
"SigningKeys": [],
"subject_types_supported": [
    "public"
]
}
}

http://customcaserver.azurewebsites.net/.well-known/jwks
{
    "keys": [
        {
            "kty": "RSA",
            "use": "sig",
            "kid": "A1bC2dE3fH4iJ5kL6mN7oP8qR9sT0u",
            "x5t": "A1bC2dE3fH4iJ5kL6mN7oP8qR9sT0u",
            "n": "jq277LRoE6WM0awT3b...vt8J6MZvmgboVB9S5CMQ",
            "e": "AQAB",
            "x5c": [
                "cZa3jz...Wo0rzA="
            ]
        }
    ]
}

```

! Note

The JWK `x5c` parameter must be present to provide X.509 representations of keys provided.

Provider metadata caching

To improve performance, Microsoft Entra ID caches metadata returned by the provider, including the keys. Provider metadata caching prevents a discovery call each time Microsoft Entra ID talks to an external identity provider.

This cache is refreshed every 24 hrs (one day). Here's how we suggest a provider rollover their keys:

1. Publish the **Existing Cert** and **New Cert** in the "`jwks_uri`".
2. Keep signing with **Existing Cert** until Microsoft Entra ID cache is refreshed, expired, or updated (every 2 days).
3. Switch to signing with **New Cert**.

We don't publish schedules for key rollovers. The dependent service must be prepared to handle both immediate and periodic rollovers. We suggest using a dedicated library built for this purpose, like [azure-active-directory-identitymodel-extensions-for-dotnet](#). For more information, see [Signing key rollover in Microsoft Entra ID](#).

Discovery of Microsoft Entra ID metadata

Providers also need to retrieve the public keys of Microsoft Entra ID to validate the tokens issued by Microsoft Entra ID.

Microsoft Entra ID metadata discovery endpoints:

- Global Azure: `https://login.microsoftonline.com/common/v2.0/.well-known/openid-configuration`
- Azure for US Government: `https://login.microsoftonline.us/common/v2.0/.well-known/openid-configuration`
- Microsoft Azure operated by 21Vianet:
`https://login.partner.microsoftonline.cn/common/v2.0/.well-known/openid-configuration`

Using the public key identifier from the token (the "kid" from [JSON Web Signature \(JWS\)](#)), one can determine which of the keys retrieved from the `jwks_uri` property should be used to validate the Microsoft Entra ID token signature.

Validating tokens issued by Microsoft Entra ID

For information about how to validate the tokens issued by Microsoft Entra ID, see [Validating and ID token](#). There are no special steps for the consumers of our discovery metadata.

Microsoft's [token validation library](#) has all the details on the specifics of token validation that are documented, or they can be ascertained from browsing the source code. For a sample, see [Azure Samples](#).

Once validation succeeds, you can work with the claims payload to get details of the user, and their tenant.

Note

It is important to validate the `id_token_hint` to ensure the `id_token_hint` is from a Microsoft tenant and represents your integration. The `id_token_hint` should be fully validated, particularly the signature, issuer, audience as well as the other claim values.

Microsoft Entra ID call to the external identity provider

Microsoft Entra ID uses the [OIDC implicit flow](#) to communicate with the external identity provider. Using this flow, communication with the provider is done exclusively by using the provider's authorization endpoint. To let the provider know the user for whom Microsoft Entra ID is making the request, Microsoft Entra ID passes a token in through the [id_token_hint](#) parameter.

This call is made through a POST request because the list of parameters passed to the provider is large. A large list prevents the use of browsers that limit the length of a GET request.

The Authentication request parameters are listed in the following table.

 **Note**

Unless they're listed in the following table, other parameters in the request should be ignored by the provider.

 [Expand table](#)

Authentication Query Parameter	Value	Description
scope	openid	
response_type	id_token	The value used for the implicit flow.
response_mode	form_post	We use form POST to avoid issues with large URLs. We expect all the parameters to be sent in the body of the request.
client_id		The client ID given to Microsoft Entra ID by the external identity provider, such as <i>ABCD</i> . For more information, see External authentication method description .
redirect_url		The redirection Uniform Resource Identifier (URI) to which the external identity provider sends the response (<code>id_token_hint</code>).
See an example after this table.		
nonce		A random string generated by Microsoft Entra ID. It can be the session ID. If provided, it needs to be returned in the response back to Microsoft Entra ID.
state		If passed in, the provider should return state in its response. Microsoft Entra ID uses state to keep context about the call.

Authentication Query Parameter	Value	Description
id_token_hint		A token issued by Microsoft Entra ID for the end user, and passed in for the benefit of the provider.
claims		A JSON blob that contains the claims requested. For details about the format of this parameter, see claims request parameter from the OIDC documentation, and an example after this table.
client-request-id	A GUID value	A provider can log this value to help troubleshoot problems.

Example of a redirection URI

The redirect Uniform Resource Identifiers (URIs) should be registered with the provider off-band. The redirect URIs that can be sent are:

- Global Azure:
`https://login.microsoftonline.com/common/federation/externalauthprovider`
- Azure for US Government:
`https://login.microsoftonline.us/common/federation/externalauthprovider`
- Microsoft Azure operated by 21Vianet:
`https://login.partner.microsoftonline.cn/common/federation/externalauthprovider`

Example of an EAM that satisfies MFA

Here's an example of an authentication where an EAM satisfies MFA. This example helps a provider know what claims Microsoft Entra ID expects.

The combination of the `acr` and `amr` values are used by Microsoft Entra ID to validate:

- The authentication method used for second factor satisfies the MFA requirement
- The authentication method differs in 'type' from the method used to complete the first factor for sign-in to Microsoft Entra ID

JSON

```
{
  "id_token": {
    "acr": {
      "essential": true,
      "values": ["possessionorinherence"]
    },
    "amr": [
      "essential": true,
      "values": ["password"]
    ]
  }
}
```

```

        "values": [ "face", "fido", "fpt", "hwk", "iris", "otp", "pop", "retina",
"sc", "sms", "swk", "tel", "vbm"]
    }
}

```

Default Id_token_hint claims

This section describes the required content of the token passed as id_token_hint in the request made to the provider. The token may contain more claims than in the following table.

[Expand table](#)

Claim	Value	Description
iss		Identifies the security token service (STS) that constructs and returns the token, and the Microsoft Entra ID tenant in which the user authenticated. Your app should use the GUID portion of the claim to restrict the set of tenants that can sign in to the app, if applicable. Issuer should match the issuer URL from the OIDC discovery JSON metadata for the tenant where the user signed in.
aud		The audience should be set to the external identity provider's client ID for Microsoft Entra ID.
exp		The expiration time is set to expire a short time after the issuing time, sufficient to avoid time skew issues. Because this token isn't meant for authentication, there's no reason for its validity to outlast the request by much.
iat		Set issuing time as usual.
tid		The tenant ID is for advertising the tenant to the provider. It represents the Microsoft Entra ID tenant that the user is from.
oid		The immutable identifier for an object in the Microsoft identity platform. In this case, it's a user account. It can also be used to perform authorization checks safely, and as a key in database tables. This ID uniquely identifies the user across applications. Two different applications that sign in the same user receive the same value in the oid claim. Thus, oid can be used in queries to Microsoft online services, such as Microsoft Graph.
preferred_username		Provides a human readable value that identifies the subject of the token. This value isn't guaranteed to be unique within a tenant, and is meant only for display purposes.
sub		Subject identifier for the end user at the Issuer. The principal about which the token asserts information, such as the user of an application. This value is immutable and can't be reassigned or reused. It can be used to perform

Claim	Value	Description
		<p>authorization checks safely, such as when the token is used to access a resource, and can be used as a key in database tables. Because the subject is always present in the tokens that Microsoft Entra ID issues, we recommend using this value in a general-purpose authorization system. The subject is, however, a pairwise identifier; it's unique to a particular application ID. <i>Therefore, if a single user signs in to two different applications using two different client IDs, those applications receive two different values for the subject claim.</i> This result may or may not be desired, depending on your architecture and privacy requirements. See also the oid claim (which does remain the same across apps within a tenant).</p>

To prevent it from being used for anything other than a hint, the token is issued as expired. The token is signed, and can be verified using the published Microsoft Entra ID discovery metadata.

Optional claims from Microsoft Entra ID

If a provider needs optional claims from Microsoft Entra ID, then you can configure the following optional claims for id_token: `given_name`, `family_name`, `preferred_username`, `upn`. For more information, see [Optional claims](#).

Recommended use of claims

Microsoft recommends associating accounts on the provider side with the account in Azure AD by using the `oid` and `tid` claims. These two claims are guaranteed to be unique for the account in the tenant.

Example of an `id_token_hint`

Here's an example of an `id_token_hint` for a directory member:

JSON
<pre>{ "typ": "JWT", "alg": "RS256", "kid": "C2dE3fH4iJ5kL6mN7oP8qR9sT0uV1w" }.{ "ver": "2.0", "iss": "https://login.microsoftonline.com/aaaabbbb-0000-cccc-1111- dddd2222eeee/v2.0", "sub": "mBfcvuhSHkDWVgV72x2ruIYdSsPSvcj2R0qfc6mGEAA", "aud": "00001111-aaaa-2222-bbbb-3333cccc4444", "exp": 1536093790, "iat": 1536093791,</pre>

```
"nbf": 1536093791,  
"name": "Test User 2",  
"preferred_username": "testuser2@contoso.com"  
"oid": "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb",  
"tid": "aaaabbbb-0000-cccc-1111-dddd2222eeee"  
}.
```

Here's an example of the id_token hint for a guest user in the tenant:

JSON

```
{  
  "typ": "JWT",  
  "alg": "RS256",  
  "kid": "C2dE3fH4iJ5kL6mN7oP8qR9sT0uV1w"  
}.{  
  "ver": "2.0",  
  "iss": "https://login.microsoftonline.com/9122040d-6c67-4c5b-b112-  
36a304b66dad/v2.0",  
  "sub": "mBfcvuhSHkDWgV72x2ruIYdSsPSvcj2R0qfc6mGEAA",  
  "aud": "00001111-aaaa-2222-bbbb-3333cccc4444",  
  "exp": 1536093790,  
  "iat": 1536093791,  
  "nbf": 1536093791,  
  "name": "External Test User (Hotmail)",  
  "preferred_username": "externaltestuser@hotmail.com",  
  "oid": "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb",  
  "tid": "aaaabbbb-0000-cccc-1111-dddd2222eeee"  
}.
```

Suggested actions for external identity providers

We suggest that external identity providers complete these steps. The list isn't exhaustive, and providers should complete other validation steps as they see fit.

1. From the request:

- Ensure that the redirect_uri is published provided in [Microsoft Entra ID call to the external identity provider](#).
- Ensure that the client_id has a value assigned to Microsoft Entra ID, such as ABCD.
- The provider should first [validate](#) the id_token_hint that is presented to it by Microsoft Entra ID.

2. From the claims in the id_token_hint:

- They can optionally make a call to [Microsoft Graph](#) to fetch other details about this user. The **oid** and **tid** claims in the `id_token_hint` is useful in this regard. For details about the claims provided in the `id_token_hint`, see [Default `id_token_hint` claims](#).

3. Then carry out any other authentication activity that the provider's product is built to do.
4. Depending upon the result of user's actions and other factors, the provider would then construct and send a response back to Microsoft Entra ID, as explained in the next section.

Microsoft Entra ID processing of the provider response

The provider needs to POST a response back to the `redirect_uri`. The following parameters should be provided on a successful response:

[Expand table](#)

Parameter	Value	Description
<code>id_token</code>		The token issued by the external identity provider.
<code>state</code>		The same state that was passed in the request, if any. Otherwise, this value shouldn't be present.

On success, the provider would then issue an `id_token` for the user. Microsoft Entra ID uses the published OIDC metadata to verify that the token contains the expected claims, and does any other validation of the token that OIDC requires.

[Expand table](#)

Claim	Value	Description
<code>iss</code>		Issuer – must match the issuer from the provider's discovery metadata.
<code>aud</code>		Audience – the Microsoft Entra ID client ID. See <code>client_id</code> in Microsoft Entra ID call to the external identity provider .
<code>exp</code>		Expiration time – set as usual.
<code>iat</code>		Issuing time – set as usual.
<code>sub</code>		Subject – must match the <code>sub</code> from the <code>id_token_hint</code> sent to initiate this request.
<code>nonce</code>		The same nonce that was passed in the request.
<code>acr</code>		The <code>acr</code> claims for the authentication request. This value should match one of the values from the request sent to initiate this request. Only one <code>acr</code> claim should be returned. For

Claim	Value	Description
		the list of claims, see Supported acr claims .
amr		The amr claims for the authentication method used in authentication. This value should be returned as an array, and only one method claim should be returned. For the list of claims, see Supported amr claims .

Supported acr claims

[+] [Expand table](#)

Claim	Notes
possessionorinherence	Authentication must take place with a possession or inherence based factor.
knowledgeorpossession	Authentication must take place with a knowledge or possession based factor.
knowledgeorinherence	Authentication must take place with a knowledge or inherence based factor.
knowledgeorpossessionorinherence	Authentication must take place with a knowledge or possession or inherence based factor.
knowledge	Authentication must take place with knowledge based factor.
possession	Authentication must take place with possession based factor.
inherence	Authentication must take place with inherence based factor.

Supported amr claims

[+] [Expand table](#)

Claim	Notes
face	Biometric with facial recognition
fido	FIDO2 was used
fpt	Biometric with fingerprint
hwk	Proof of possession of hardware-secured key
iris	Biometric with iris scan

Claim	Notes
otp	One time password
pop	Proof of possession
retina	Biometric of retina scan
sc	Smart card
sms	Confirmation by text to registered number
swk	Confirmation of presence of a software-secured key
tel	Confirmation by telephone
vbm	Biometric with voiceprint

Microsoft Entra ID requires MFA to be satisfied to issue a token with MFA claims. As a result, only methods with a different type can satisfy the second factor requirement. As mentioned earlier, the different method types that can be used to satisfy the second factor are knowledge, possession, and inherence.

Microsoft Entra ID validates the type mapping based on the following table.

[Expand table](#)

Claim Method	Type	Notes
face	Inherence	Biometric with facial recognition
fido	Possession	FIDO2 was used. Some implementations may also require biometric, but possession method type is mapped because it's the primary security attribute.
fpt	Inherence	Biometric with fingerprint
hwk	Possession	Proof of possession of hardware-secured key
iris	Inherence	Biometric with iris scan
otp	Possession	One-time password
pop	Possession	Proof of possession
retina	Inherence	Biometric of retina scan
sc	Possession	Smart card
sms	Possession	Confirmation by text to registered number

Claim Method	Type	Notes
swk	Possession	Proof of presence of a software-secured key
tel	Possession	Confirmation by telephone
vbm	Inherence	Biometric with voiceprint

If no issues are found with the token, then Microsoft Entra ID considers MFA to be satisfied, and issues a token to the end user. Otherwise, the end user's request fails.

Failure is indicated by issuing error response parameters.

[] [Expand table](#)

Parameter	Value	Description
Error	An ASCII error code, such as access_denied or temporarily_unavailable.	

Microsoft Entra ID considers the request successful if the id_token parameter is present in the response, and if the token is valid. Otherwise, the request is considered unsuccessful. Microsoft Entra ID fails the original authentication attempt due to requirement of the Conditional Access policy.

Microsoft Entra ID abandons the state of the authentication attempt on its end about 5 minutes after the redirection to the provider.

Microsoft Entra ID error response handling

Microsoft Azure services use a correlationId to correlate calls across various internal and external systems. It serves as a common identifier of the whole operation or flow that potentially involves multiple HTTP calls. When an error occurs during any of the operations, the response contains a field named Correlation ID.

When you reach out to Microsoft support or a similar service, provide the value of this Correlation ID as it helps to access the telemetry and logs faster.

For example:

ENTRA IDSTS70002: Error validating credentials. ENTRA IDSTS50012: External ID token from issuer 'https://sts.XXXXXXXXXX.com/auth/realms/XXXXXXXXXmfa' failed signature verification. KeyID of token is 'A1bC2dE3fH4iJ5kL6mN7oP8qR9sT0u' Trace ID: 0000aaaa-

Custom controls and EAMs

In Microsoft Entra ID, EAMs and Conditional Access custom controls can operate in parallel while customers prepare for and migrate to EAMs.

Customers who currently use an integration with an external provider by using custom controls can continue to use them, and any Conditional Access policies they configured to manage access. Admins are recommended to create a parallel set of Conditional Access policies during the migration period:

- The policies should use the **Require multifactor authentication** grant control instead of the custom control grant.

 **Note**

Grant controls based on authentication strengths, including the built-in MFA strength, aren't satisfied by the EAM. Policies should only be configured with **Require multifactor authentication**. Support for EAMs with authentication strengths will come later.

- The new policy can be tested first with a subset of users. The test group would be excluded from the policy that requires the custom controls, and included in the policy that requires MFA. Once the admin is comfortable that the policy that requires MFA is satisfied by the EAM, the admin can include all required users in the policy with the MFA grant, and the policy configured for custom controls can be moved to **Off**.

Integration support

If you have any issues when you build EAM integration with Microsoft Entra ID, the Microsoft Customer Experience Engineering (CxE) Independent Solution Vendor (ISV) may be able to assist. To engage with the CxE ISV team, submit a [request for assistance](#).

References

- [OAuth2.0 and OIDC specification](#)

Glossary

 Expand table

Term	Description
MFA	Multifactor authentication.
EAM	An external authentication method is an authentication method from a provider other than Microsoft Entra ID that is used as part of authenticating a user.
OIDC	Open ID Connect is an authentication protocol based on OAuth 2.0.
00001111-aaaa-2222-bbbb-3333cccc4444	An example of an appid integrated for an external authentication method.

Next steps

For more information about how to configure an EAM in [Microsoft Entra admin center](#), see [Manage an external authentication method in Microsoft \(Preview\)](#).

Reauthentication prompts and session lifetime for Microsoft Entra multifactor authentication

Article • 03/04/2025

Microsoft Entra ID has multiple settings that determine how often users need to reauthenticate. This reauthentication might involve only a first factor, such as password, Fast IDentity Online (FIDO), or passwordless Microsoft Authenticator. Or it might require multifactor authentication (MFA). You can configure these reauthentication settings as needed for your own environment and the user experience that you want.

The Microsoft Entra ID default configuration for user sign-in frequency is a rolling window of 90 days. Asking users for credentials often seems like a sensible thing to do, but it can backfire. If users are trained to enter their credentials without thinking, they can unintentionally supply them to a malicious credential prompt.

It might sound alarming to not ask for a user to sign back in. However, any violation of IT policies revokes the session. Some examples include a password change, an incompliant device, or an operation to disable an account. You can also explicitly [revoke users' sessions by using Microsoft Graph PowerShell](#).

This article details recommended configurations and how various settings work and interact with each other.

Recommended settings

To give your users the right balance of security and ease of use by asking them to sign in at the right frequency, we recommend the following configurations:

- If you have Microsoft Entra ID P1 or P2:
 - Enable single sign-on (SSO) across applications by using [managed devices](#) or [seamless SSO](#).
 - If reauthentication is required, use a Microsoft Entra Conditional Access [Sign-in frequency](#) policy.
 - For users who sign in from unmanaged devices or for mobile device scenarios, persistent browser sessions might not be preferable. Or you might use Conditional Access to enable persistent browser sessions with the [Sign-in frequency](#) policy. Limit the duration to an appropriate time based on the sign-in risk, where a user with less risk has a longer session duration.

- If you have a Microsoft 365 Apps license or a Microsoft Entra ID Free license:
 - Enable SSO across applications by using [managed devices](#) or [seamless SSO](#).
 - Keep the **Show option to remain signed in** option enabled and guide your users to accept **Stay signed in?** at sign-in.
- For mobile device scenarios, make sure your users use the Microsoft Authenticator app. This app is a broker to other Microsoft Entra ID federated apps, and it reduces authentication prompts on the device.

Our research shows that these settings are right for most tenants. Some combinations of these settings, such as **Remember multifactor authentication** and **Show option to remain signed in**, can result in prompts for your users to authenticate too often. Regular reauthentication prompts are bad for user productivity and can make users more vulnerable to attacks.

Configure settings for Microsoft Entra session lifetime

To optimize the frequency of authentication prompts for your users, you can configure settings for the Microsoft Entra session lifetime. Understand the needs of your business and users, and configure settings that provide the best balance for your environment.

Session lifetime policies

Without any session lifetime settings, the browser session has no persistent cookies. Every time users close and open the browser, they get a prompt for reauthentication. In Office clients, the default time period is a rolling window of 90 days. With this default Office configuration, if the user resets the password or the session is inactive for more than 90 days, the user must reauthenticate with the required first and second factors.

A user might see multiple MFA prompts on a device that doesn't have an identity in Microsoft Entra ID. Multiple prompts result when each application has its own OAuth Refresh Token that isn't shared with other client apps. In this scenario, MFA prompts multiple times as each application requests an OAuth Refresh Token to be validated with MFA.

In Microsoft Entra ID, the most restrictive policy for session lifetime determines when the user needs to reauthenticate. Consider a scenario in which you enable both of these settings:

- **Show option to remain signed in**, which uses a persistent browser cookie
- **Remember multifactor authentication** with a value of 14 days

In this example, the user needs to reauthenticate every 14 days. This behavior follows the most restrictive policy, even though **Show option to remain signed in** by itself wouldn't require the user to reauthenticate on the browser.

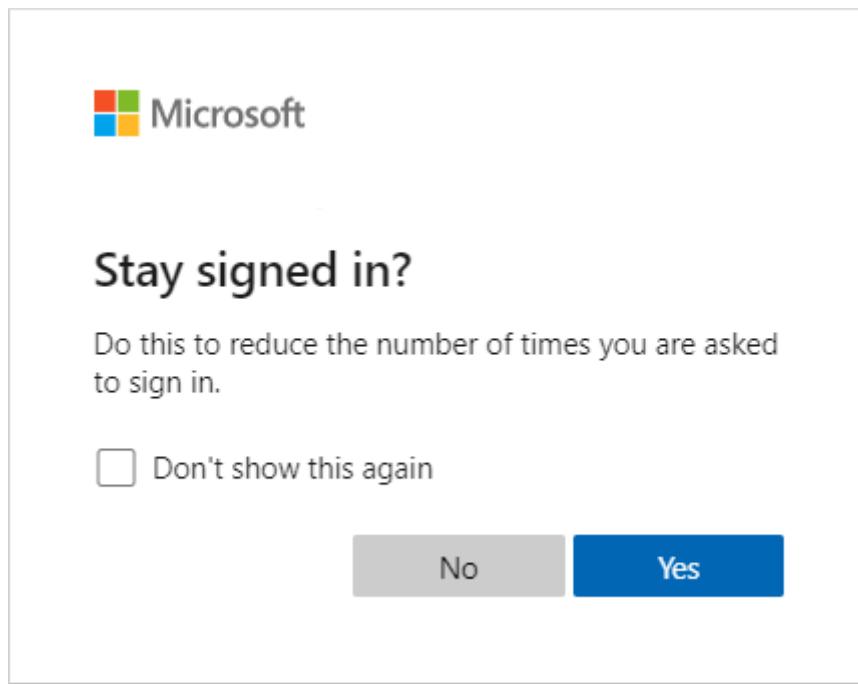
Managed devices

Devices joined to Microsoft Entra ID through Microsoft Entra join or Microsoft Entra hybrid join receive a [Primary Refresh Token \(PRT\)](#) to use SSO across applications.

This PRT lets a user sign in once on the device and allows IT staff to make sure that the device meets standards for security and compliance. If you need to ask a user to sign in more frequently on a joined device for some apps or scenarios, you can use the [Conditional Access Sign-in frequency](#) policy.

Option to remain signed in

When a user selects **Yes** on the **Stay signed in?** prompt option during sign-in, the selection sets a persistent cookie on the browser. This persistent cookie remembers both first and second factors, and it applies only for authentication requests in the browser.

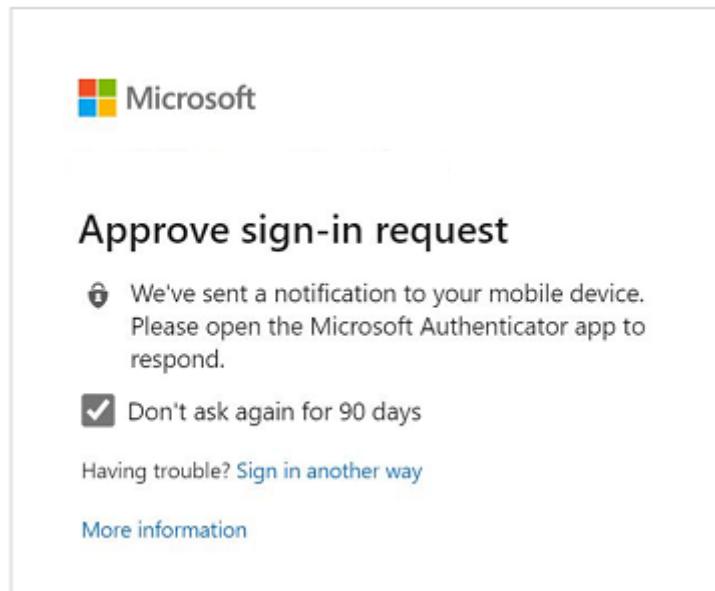


If you have a Microsoft Entra ID P1 or P2 license, we recommend using a Conditional Access policy for **Persistent browser session**. This policy overwrites the **Show option to remain signed in** setting and provides an improved user experience. If you don't have a Microsoft Entra ID P1 or P2 license, we recommend enabling the **Show option to remain signed in** setting for your users.

For more information on configuring the option to let users remain signed in, see [Manage the 'Stay signed in?' prompt](#).

Option to remember multifactor authentication

The **Remember multifactor authentication** setting lets you configure a value of 1 to 365 days. It sets a persistent cookie on the browser when a user selects the **Don't ask again for X days** option at sign-in.



Although this setting reduces the number of authentications on web apps, it increases the number of authentications for modern authentication clients, such as Office clients. These clients normally prompt only after password reset or inactivity of 90 days. However, setting this value to less than 90 days shortens the default MFA prompts for Office clients, and it increases reauthentication frequency. When you use this setting in combination with **Show option to remain signed in** or Conditional Access policies, it might increase the number of authentication requests.

If you use **Remember multifactor authentication** and have a Microsoft Entra ID P1 or P2 license, consider migrating these settings to Conditional Access **Sign-in frequency**. Otherwise, consider using **Show option to remain signed in** instead.

More information, see [Remember multifactor authentication](#).

Authentication session management with Conditional Access

The administrator can use the **Sign-in frequency** policy to choose a sign-in frequency that applies for both first and second factor in both client and browser. We recommend using these settings, along with using managed devices, in scenarios where you need to

restrict authentication sessions. For example, you might need to restrict an authentication session for critical business applications.

Persistent browser session allows users to remain signed in after closing and reopening their browser window. Like the **Show option to remain signed in** setting, it sets a persistent cookie on the browser. But because the admin configures it, it doesn't require the user to select **Yes** in the **Stay signed-in?** option. In that way, it provides a better user experience. If you use the **Show option to remain signed in** option, we recommend that you enable the **Persistent browser session** policy instead.

For more information, see [Configure adaptive session lifetime policies](#).

Configurable token lifetimes

The **Configurable token lifetimes** setting allows configuration of a lifetime for a token that Microsoft Entra ID issues. **Authentication session management with Conditional Access** replaces this policy. If you're using **Configurable token lifetimes** now, we recommend starting the migration to the Conditional Access policies.

Review your tenant configuration

Now that you understand how various settings work and the recommended configuration, it's time to check your tenants. You can start by looking at the sign-in logs to understand which session lifetime policies were applied during sign-in.

Under each sign-in log, go to the **Authentication Details** tab and explore **Session Lifetime Policies Applied**. For more information, see [Learn about the sign-in log activity details](#).

Activity Details: Sign-ins						
Basic info	Location	Device info	Authentication Details	Conditional Access	Report-only	Additional Details
Authentication Policies Applied				Session Lifetime Policies Applied		
Conditional Access		Remember MFA				
Date	Authentication met...	Authentication met...	Succeeded	Result detail	Requirements	
10/20/2021, 1:27:42 PM	FIDO2 security key	My FIDO2 Key	true			
10/20/2021, 1:27:42 PM	Previously satisfied		true	MFA requirement skip...		

To configure or review the **Show option to remain signed in** option:

Important

Microsoft recommends that you use roles with the fewest permissions. This practice helps improve security for your organization. Global Administrator is a highly privileged role that should be limited to emergency scenarios or when you can't use an existing role.

1. Sign in to the [Microsoft Entra admin center](#) as a **Global Administrator**.
2. Browse to **Identity > Company Branding**. Then, for each locale, select **Show option to remain signed in**.
3. Select **Yes**, and then select **Save**.

To remember multifactor authentication settings on trusted devices:

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Policy Administrator**.
2. Browse to **Protection > Multifactor authentication**.
3. Under **Configure**, select **Additional cloud-based MFA settings**.
4. On the **Multifactor authentication service settings** pane, scroll to **Remember multifactor authentication settings** and select the checkbox.

To configure Conditional Access policies for sign-in frequency and persistent browser sessions:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Conditional Access Administrator**.
2. Browse to **Protection > Conditional Access**.
3. Configure a policy by using the options for session management that this article recommends.

To review token lifetimes, [use Microsoft Graph PowerShell to query any Microsoft Entra policies](#). Disable any policies that you have in place.

If more than one setting is enabled in your tenant, we recommend that you update your settings based on the licensing available for you. For example, if you have a Microsoft Entra ID P1 or P2 license, you should use only the Conditional Access policies of **Sign-in frequency** and **Persistent browser session**. If you have a Microsoft 365 Apps license or a Microsoft Entra ID Free license, you should use the **Show option to remain signed in** configuration.

If you enabled configurable token lifetimes, keep in mind that this capability will be removed soon. Plan a migration to a Conditional Access policy.

The following table summarizes the recommendations based on licenses:

Category	Microsoft 365 Apps or Microsoft Entra ID Free	Microsoft Entra ID P1 or P2
SSO	Microsoft Entra join or Microsoft Entra hybrid join, or seamless SSO for unmanaged devices	Microsoft Entra join or Microsoft Entra hybrid join
Reauthentication settings	Show option to remain signed in	Conditional Access policies for sign-in frequency and persistent browser sessions

Related content

- [Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication](#)
- [Tutorial: Use risk detections for user sign-ins to trigger Microsoft Entra multifactor authentication or password changes](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Understanding telephony fraud

Article • 03/04/2025

In today's digital landscape, telecommunication services seamlessly integrate into our daily lives. But technological progress also brings the risk of fraudulent activities like International Revenue Share Fraud (IRSF), which poses financial consequences and service disruptions. IRSF involves exploiting telecommunication billing systems by unauthorized actors. They divert telephony traffic and generate profits through a technique called *traffic pumping*. Traffic pumping targets multifactor authentication systems, and causes inflated charges, service unreliability, and system errors.

To counter this risk, a thorough understanding of IRSF is crucial for implementing preventive measures like regional restrictions and phone number verification, while our system aims to minimize disruptions and safeguard both our business, users, and your business we prioritize your security and as such we may sometimes take proactive measures.

How we help fight telephony fraud

To protect our customers and vigilantly defend against bad actors who attempt fraud, we may engage in proactive remediation in the event of a fraud attack. Telephony fraud is a very dynamic space where even seconds can result in massive financial impact. To limit that impact, we may proactively engage temporary throttling when we detect excessive authentication requests from a particular region, phone, or user. These throttles normally clear after a few hours to a few days.

How you can help fight telephony fraud

To help fight telephony fraud, B2C customers can take steps to improve security of authentication activities such as sign-in, MFA, password reset, and forgot username:

- Use the recommended versions of user flows
- Remove region codes that aren't relevant to your organization
- Use CAPTCHA to help distinguish between human users and automated bots
- Review your telecom usage to make sure it matches the expected behavior from your users

For more information, see [Securing phone-based MFA in B2C](#).

In addition, you may sometimes encounter throttles because you're requesting traffic from a region that requires an opt-in. For more information, see [Regions that need to opt in for MFA telephony verification](#).

Next steps

- [Authentication methods in Microsoft Entra ID](#)
 - [Securing phone-based MFA in B2C](#)
 - [Regions that need to opt in for MFA telephony verification](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Regions that need to opt in for MFA telephony verification

Article • 04/17/2025

As a protection for our customers, Microsoft doesn't automatically support telephony verification for certain region codes. If you want to receive traffic from phone numbers with these region codes, your administrator must submit a support ticket and request to opt in.

Why this protection is needed

In today's digital world, telecommunication services have become ingrained into our lives. But, advancements come with a risk of fraudulent activities. International Revenue Share Fraud (IRSF) is a threat with severe financial implications that also makes using services more difficult. Let's look at IRSF fraud more in-depth.

IRSF is a type of telephony fraud where criminals exploit the billing system of telecommunication services providers to make profit for themselves. Bad actors gain unauthorized access to a telecommunication network and divert traffic to those networks to skim profit for every transaction that is sent to that network. To divert traffic, bad actors steal existing usernames and passwords, create new usernames and passwords, or try a host of other things to send text messages and voice calls through their telecommunication network. Bad actors take advantage of multifactor authentication screens, which require a text message or voice call before a user can access their account. This activity causes exorbitant charges and makes services unreliable for our customers, causing downtime, and system errors.

Here's how an IRSF attack may happen:

1. A bad actor first gets premium rate phone numbers and registers them.
2. A bad actor uses automated scripts to request voice calls or text messages. The bad actor is colluding with number providers and the telecommunication network to drive more traffic to those services. The bad actor skims some of the profits of the increased traffic.
3. A bad actor will hop around different region codes to continue to drive traffic and make it hard for them to get caught.

The most common way to conduct IRSF is through an end-user experience that requires a two-factor authentication code. Bad actors add those premium rate phone numbers and pump traffic to them by requesting two-factor authentication codes. This activity results in revenue-skimming, and can lead to billions of dollars in loss.

IRSF poses a significant threat to online businesses and can cause reputational damage. By understanding IRSF, you can be more aware of the problem and can engage in implementing

preventive measures such as regional restrictions, rate limiting, and phone number verification.

SMS verification

For SMS verification, the following region codes require an opt-in. This means that if you'd like to use telecom in these regions, you'll have to reach out to support.

 [Expand table](#)

Region Code	Region Name
7	Russia
20	Egypt
53	Cuba
58	Venezuela
62	Indonesia
63	Philippines
84	Vietnam
92	Pakistan
93	Afghanistan
94	Sri Lanka
95	Myanmar
98	Iran
211	South Sudan
212	Morocco
213	Algeria
216	Tunisia
218	Libya
220	Gambia
221	Senegal
222	Mauritania

Region Code	Region Name
223	Mali
224	Guinea
225	Cote d'Ivoire
226	Burkina Faso
227	Niger
228	Togo
229	Benin
231	Liberia
232	Sierra Leone
233	Ghana
234	Nigeria
235	Chad
236	Central African Republic
237	Cameroon
238	Cabo Verde
239	São Tomé and Príncipe
240	Equatorial Guinea
241	Gabon
242	Congo
243	Congo
244	Angola
245	Guinea-Bissau
246	British Indian Ocean Territory
247	Ascension Island
248	Seychelles
249	Sudan

Region Code	Region Name
250	Rwanda
251	Ethiopia
252	Somalia
253	Djibouti
254	Kenya
255	Tanzania
256	Uganda
257	Burundi
258	Mozambique
260	Zambia
261	Madagascar
262	Mayotte
263	Zimbabwe
265	Malawi
266	Lesotho
267	Botswana
268	Antigua and Barbuda
269	Comoros
290	Saint Helena, Ascension, and Tristan da Cunha
291	Eritrea
297	Aruba
299	Greenland
350	Gibraltar
355	Albania
356	Malta
359	Bulgaria

Region Code	Region Name
370	Lithuania
371	Latvia
372	Estonia
373	Moldova
374	Armenia
375	Belarus
376	Andorra
377	Monaco
380	Ukraine
381	Serbia
382	Montenegro
383	Kosovo
385	Croatia
386	Slovenia
387	Bosnia and Herzegovina
389	North Macedonia
500	Falkland Islands
501	Belize
502	Guatemala
503	El Salvador
504	Honduras
505	Nicaragua
507	Panama
508	Saint Pierre and Miquelon
509	Haiti
591	Bolivia

Region Code	Region Name
592	Guyana
593	Ecuador
594	French Guiana
597	Suriname
598	Uruguay
670	Timor-Leste
672	Antarctica
674	Nauru
675	Papua New Guinea
676	Tonga
677	Solomon Islands
678	Vanuatu
681	Wallis and Futuna
682	Cook Islands
683	Niue
685	Samoa
686	Kiribati
687	New Caledonia
689	French Polynesia
690	Tokelau
691	Micronesia
692	Marshall Islands
856	Laos
960	Maldives
961	Lebanon
962	Jordan

Region Code	Region Name
963	Syria
964	Iraq
966	Saudi Arabia
967	Yemen
968	Oman
971	United Arab Emirates
972	Israel
973	Bahrain
974	Qatar
975	Bhutan
976	Mongolia
977	Nepal
992	Tajikistan
993	Turkmenistan
994	Azerbaijan
995	Georgia
996	Kyrgyzstan
998	Uzbekistan
1242	Bahamas
1264	Anguilla
1268	Antigua and Barbuda
1284	British Virgin Islands
1345	Cayman Islands
1473	Grenada
1649	Turks and Caicos Islands
1664	Montserrat

Region Code	Region Name
1721	Sint Maarten
1758	Saint Lucia
1809	Dominica
1829	Dominican Republic
1849	Dominican Republic
1869	Saint Kitts and Nevis
1876	Jamaica
1767	Dominica
970	Palestinian Authority
880	Bangladesh
1868	Trinidad and Tobago
1441	Bermuda
423	Liechtenstein
965	Kuwait
855	Cambodia
1658	Jamaica
32	Belgium
1671	Guam
91	India
680	Palau
595	Paraguay
51	Peru
1670	Northern Mariana Islands
378	San Marino
27	South Africa
688	Tuvalu

Region Code	Region Name
54	Argentina
1246	Barbados
679	Fiji
49	Germany
590	Saint Barthelemy, Saint Martin, Guadeloupe
60	Malaysia
596	Martinique
230	Mauritius
31	Netherlands
599	Curacao, Netherlands Antilles
850	North Korea
1787	Puerto Rico
1939	Puerto Rico
1784	Saint Vincent and the Grenadines

Voice verification

For voice verification, the following region codes require an opt-in.

[Expand table](#)

Region Code	Region Name
53	Cuba
58	Venezuela
93	Afghanistan
94	Sri Lanka
95	Myanmar (Burma)
98	Iran

Region Code	Region Name
211	South Sudan
212	Morocco
213	Algeria
216	Tunisia
218	Libya
220	Gambia
221	Senegal
222	Mauritania
223	Mali
224	Guinea
225	Cote d'Ivoire
226	Burkina Faso
227	Niger
228	Togo
229	Benin
231	Liberia
232	Sierra Leone
233	Ghana
235	Chad
236	Central African Republic
237	Cameroon
238	Cabo Verde
239	São Tomé and Príncipe
240	Equatorial Guinea
241	Gabon
242	Congo

Region Code	Region Name
243	Congo
244	Angola
245	Guinea-Bissau
246	British Indian Ocean Territory
247	Ascension Island
248	Seychelles
249	Sudan
250	Rwanda
251	Ethiopia
252	Somalia
253	Djibouti
254	Kenya
255	Tanzania
256	Uganda
257	Burundi
258	Mozambique
260	Zambia
261	Madagascar
262	Mayotte
263	Zimbabwe
265	Malawi
266	Lesotho
267	Botswana
268	Antigua and Barbuda
269	Comoros
290	Saint Helena, Ascension, and Tristan da Cunha

Region Code	Region Name
291	Eritrea
297	Aruba
299	Greenland
350	Gibraltar
355	Albania
356	Malta
359	Bulgaria
370	Lithuania
371	Latvia
372	Estonia
373	Moldova
374	Armenia
375	Belarus
376	Andorra
377	Monaco
381	Serbia
382	Montenegro
383	Kosovo
385	Croatia
386	Slovenia
387	Bosnia and Herzegovina
389	North Macedonia
500	Falkland Islands
501	Belize
502	Guatemala
503	El Salvador

Region Code	Region Name
504	Honduras
505	Nicaragua
507	Panama
508	Saint Pierre and Miquelon
509	Haiti
591	Bolivia
592	Guyana
593	Ecuador
594	French Guiana
597	Suriname
598	Uruguay
670	Timor-Leste
672	Antarctica
674	Nauru
675	Papua New Guinea
676	Tonga
677	Solomon Islands
678	Vanuatu
681	Wallis and Futuna
682	Cook Islands
683	Niue
685	Samoa
686	Kiribati
687	New Caledonia
689	French Polynesia
690	Tokelau

Region Code	Region Name
691	Micronesia
692	Marshall Islands
856	Laos
960	Maldives
961	Lebanon
962	Jordan
963	Syria
964	Iraq
967	Yemen
968	Oman
973	Bahrain
974	Qatar
975	Bhutan
976	Mongolia
977	Nepal
992	Tajikistan
993	Turkmenistan
994	Azerbaijan
995	Georgia
996	Kyrgyzstan
998	Uzbekistan
1242	Bahamas
1264	Anguilla
1268	Antigua and Barbuda
1284	British Virgin Islands
1345	Cayman Islands

Region Code	Region Name
1473	Grenada
1649	Turks and Caicos Islands
1664	Montserrat
1721	Sint Maarten
1758	Saint Lucia
1809	Dominica
1829	Dominican Republic
1849	Dominican Republic
1869	Saint Kitts and Nevis
1876	Jamaica
1658	Jamaica
32	Belgium
1671	Guam
91	India
680	Palau
595	Paraguay
51	Peru
1670	Northern Mariana Islands
378	San Marino
27	South Africa
688	Tuvalu
54	Argentina
1246	Barbados
679	Fiji
49	Germany
590	Saint Barthelemy, Saint Martin, Guadeloupe

Region Code	Region Name
60	Malaysia
596	Martinique
230	Mauritius
31	Netherlands
599	Curacao, Netherlands Antilles
850	North Korea
1787	Puerto Rico
1939	Puerto Rico
1784	Saint Vincent and the Grenadines

Next steps

- [Understanding telephony fraud](#)
- [Authentication methods in Microsoft Entra ID](#)

Data residency and customer data for Microsoft Entra multifactor authentication

Article • 03/04/2025

Microsoft Entra ID stores customer data in a geographical location based on the address an organization provides when subscribing to a Microsoft online service such as Microsoft 365 or Azure. For information on where your customer data is stored, see [Where your data is located](#) in the Microsoft Trust Center.

Cloud-based Microsoft Entra multifactor authentication and MFA Server process and store personal data and organizational data. This article outlines what and where data is stored.

The Microsoft Entra multifactor authentication service has datacenters in the United States, Europe, and Asia Pacific. The following activities originate from the regional datacenters except where noted:

- Multifactor authentication SMS and phone calls originate from datacenters in the customer's region and are routed by global providers. Phone calls using custom greetings always originate from data centers in the United States.
- General purpose user authentication requests from other regions are currently processed based on the user's location.
- Push notifications that use the Microsoft Authenticator app are currently processed in regional datacenters based on the user's location. Vendor-specific device services, such as Apple Push Notification Service or Google Firebase Cloud Messaging, might be outside the user's location.

Personal data stored by Microsoft Entra multifactor authentication

Personal data is user-level information that's associated with a specific person. The following data stores contain personal information:

- Blocked users
- Bypassed users
- Microsoft Authenticator device token change requests
- Multifactor authentication activity reports—store multifactor authentication activity from the multifactor authentication on-premises components: NPS Extension, AD

FS adapter and MFA server.

- Microsoft Authenticator activations

This information is retained for 90 days.

Microsoft Entra multifactor authentication doesn't log personal data such as usernames, phone numbers, or IP addresses. However, *UserObjectId* identifies authentication attempts to users. Log data is stored for 30 days.

Data stored by Microsoft Entra multifactor authentication

For Azure public clouds, excluding Azure AD B2C authentication, the NPS Extension, and the Windows Server 2016 or 2019 Active Directory Federation Services (AD FS) adapter, the following personal data is stored:

[\[+\] Expand table](#)

Event type	Data store type
OATH token	Multifactor authentication logs
One-way SMS	Multifactor authentication logs
Voice call	Multifactor authentication logs Multifactor authentication activity report data store Blocked users (if fraud was reported)
Microsoft Authenticator notification	Multifactor authentication logs Multifactor authentication activity report data store Blocked users (if fraud was reported) Change requests when the Microsoft Authenticator device token changes

For Microsoft Azure Government, Microsoft Azure operated by 21Vianet, Azure AD B2C authentication, the NPS extension, and the Windows Server 2016 or 2019 AD FS adapter, the following personal data is stored:

[\[+\] Expand table](#)

Event type	Data store type
OATH token	Multifactor authentication logs Multifactor authentication activity report data store
One-way SMS	Multifactor authentication logs Multifactor authentication activity report data store

Event type	Data store type
Voice call	Multifactor authentication logs Multifactor authentication activity report data store Blocked users (if fraud was reported)
Microsoft Authenticator notification	Multifactor authentication logs Multifactor authentication activity report data store Blocked users (if fraud was reported) Change requests when the Microsoft Authenticator device token changes

Data stored by MFA Server

If you use MFA Server, the following personal data is stored.

ⓘ Important

In September 2022, Microsoft announced deprecation of Azure Multifactor authentication Server. Beginning September 30, 2024, Azure Multifactor authentication Server deployments will no longer service multifactor authentication requests, which could cause authentications to fail for your organization. To ensure uninterrupted authentication services and to remain in a supported state, organizations should [migrate their users' authentication data](#) to the cloud-based Azure MFA service by using the latest Migration Utility included in the most recent [Azure MFA Server update](#). For more information, see [Azure MFA Server Migration](#).

[] [Expand table](#)

Event type	Data store type
OATH token	Multifactor authentication logs Multifactor authentication activity report data store
One-way SMS	Multifactor authentication logs Multifactor authentication activity report data store
Voice call	Multifactor authentication logs Multifactor authentication activity report data store Blocked users (if fraud was reported)
Microsoft Authenticator notification	Multifactor authentication logs Multifactor authentication activity report data store Blocked users (if fraud was reported)

Event type	Data store type
	Change requests when Microsoft Authenticator device token changes

Organizational data stored by Microsoft Entra multifactor authentication

Organizational data is tenant-level information that can expose configuration or environment setup. Tenant settings from the multifactor authentication pages might store organizational data such as lockout thresholds or caller ID information for incoming phone authentication requests:

- Account lockout
- Fraud alert
- Notifications
- Phone call settings

For MFA Server, the following pages might contain organizational data:

- Server settings
- One-time bypass
- Caching rules
- Multi-Factor Authentication Server status

Multifactor authentication activity reports for public cloud

Multifactor authentication activity reports store activity from on-premises components: NPS Extension, AD FS adapter, and MFA server. The multifactor authentication service logs are used to operate the service. The following sections show where activity reports and services logs are stored for specific authentication methods for each component in different customer regions. Standard voice calls may failover to a different region.

Note

The multifactor authentication activity reports contain personal data such as User Principal Name (UPN) and complete phone number.

MFA server and cloud-based MFA

[Expand table](#)

Component	Authentication method	Customer region	Activity report location	Service log location
MFA server	All methods	Any	United States	MFA backend in United States
Cloud MFA	All methods	Any	Microsoft Entra sign-in logs in region	Cloud in-region

Multifactor authentication activity reports for sovereign clouds

The following table shows the location for service logs for sovereign clouds.

[Expand table](#)

Sovereign cloud	Sign-in logs	Multifactor authentication activity report	Multifactor authentication service logs
Microsoft Azure operated by 21Vianet	China	United States	United States
Microsoft Government Cloud	United States	United States	United States

Next steps

For more information about what user information is collected by cloud-based Microsoft Entra multifactor authentication and MFA Server, see [Microsoft Entra multifactor authentication user data collection](#).

Feedback

Was this page helpful?

 [Yes](#) [No](#)

[Provide product feedback ↗](#)

Features and licenses for Microsoft Entra multifactor authentication

Article • 03/04/2025

To protect user accounts in your organization, multifactor authentication should be used. This feature is especially important for accounts that have privileged access to resources. Basic multifactor authentication features are available to Microsoft 365 and Microsoft Entra ID users and administrators for no extra cost. If you want to upgrade the features for your admins or extend multifactor authentication to the rest of your users with more authentication methods and greater control, you can enable Microsoft Entra multifactor authentication by using Conditional Access. For more information, see [Common Conditional Access policy: Require MFA for all users](#).

Important

This article details the different ways that Microsoft Entra multifactor authentication can be licensed and used. For specific details about pricing and billing, see the [Microsoft Entra pricing page](#).

Available versions of Microsoft Entra multifactor authentication

Microsoft Entra multifactor authentication can be used, and licensed, in a few different ways depending on your organization's needs. All tenants are entitled to basic multifactor authentication features by using security defaults. You may already be entitled to use advanced Microsoft Entra multifactor authentication depending on the license you currently have. For example, the first 50,000 monthly active users in Microsoft Entra External ID can use MFA and other Premium P1 or P2 features for free. For more information, see [Azure Active Directory B2C pricing](#).

The following table details the different ways to get Microsoft Entra multifactor authentication and some of the features and use cases for each.

 Expand table

If you're a user of	Capabilities and use cases
Microsoft 365 Business Premium	EMS E3, Microsoft 365 E3, and Microsoft 365 Business Premium includes Microsoft Entra ID P1. EMS E5 or Microsoft 365 E5 includes Microsoft

If you're a user of	Capabilities and use cases
and EMS or Microsoft 365 E3 and E5	Entra ID P2. You can use the same Conditional Access features noted in the following sections to provide multifactor authentication to users.
Microsoft Entra ID P1	You can use Microsoft Entra Conditional Access to prompt users for multifactor authentication during certain scenarios or events to fit your business requirements.
Microsoft Entra ID P2	Provides the strongest security position and improved user experience. Adds risk-based Conditional Access to the Microsoft Entra ID P1 features that adapts to user's patterns and minimizes multifactor authentication prompts.
All Microsoft 365 plans	Microsoft Entra multifactor authentication can be enabled for all users using security defaults . Management of Microsoft Entra multifactor authentication is through the Microsoft 365 portal. For an improved user experience, upgrade to Microsoft Entra ID P1 or P2 and use Conditional Access. For more information, see secure Microsoft 365 resources with multifactor authentication .
Office 365 free Microsoft Entra ID Free	You can use security defaults to prompt users for multifactor authentication as needed but you don't have granular control of enabled users or scenarios, but it does provide that additional security step.

Feature comparison based on licenses

The following table provides a list of the features that are available in the various versions of Microsoft Entra ID for multifactor authentication. Plan out your needs for securing user authentication, then determine which approach meets those requirements. For example, although Microsoft Entra ID Free provides security defaults that provide Microsoft Entra multifactor authentication where only the mobile authenticator app can be used for the authentication prompt. This approach may be a limitation if you can't ensure the mobile authentication app is installed on a user's personal device. See [Microsoft Entra ID Free tier](#) later in this topic for more details.

[+] [Expand table](#)

Feature	Microsoft Entra ID Free - Security defaults (enabled for all users)	Microsoft Entra ID Free - Global Administrators only	Office 365	Microsoft Entra ID P1	Microsoft Entra ID P2
Protect Microsoft Entra tenant admin accounts with MFA	•	• (<i>Microsoft Entra Global Administrator accounts only</i>)	•	•	•
Mobile app as a second factor	•	•	•	•	•
Phone call as a second factor			•	•	•
Text message as a second factor		•	•	•	•
Admin control over verification methods		•	•	•	•
Fraud alert				•	•
MFA Reports				•	•
Custom greetings for phone calls				•	•
Custom caller ID for phone calls				•	•
Trusted IPs				•	•
Remember MFA for trusted devices		•	•	•	•
MFA for on-premises applications				•	•
Conditional Access				•	•
Risk-based Conditional					•

Feature	Microsoft Entra ID Free - Security defaults (enabled for all users)	Microsoft Entra ID Free - Global Administrators only	Office 365	Microsoft Entra ID P1	Microsoft Entra ID P2
Access					

Compare multifactor authentication policies

Our recommended approach to enforce MFA is using [Conditional Access](#). Review the following table to determine what capabilities are included in your licenses.

[\[+\] Expand table](#)

Policy	Security defaults	Conditional Access	Per-user MFA
Management			
Standard set of security rules to keep your company safe	●		
One-click on/off	●		
Included in Office 365 licensing (See license considerations)	●		●
Pre-configured templates in Microsoft 365 Admin Center wizard	●	●	
Configuration flexibility		●	
Functionality			
Exempt users from the policy		●	●
Authenticate by phone call or text message	●	●	●
Authenticate by Microsoft Authenticator and Software tokens	●	●	●
Authenticate by FIDO2, Windows Hello for Business, and Hardware tokens		●	●
Blocks legacy authentication protocols	●	●	●
New employees are automatically protected	●	●	

Policy	Security defaults	Conditional Access	Per-user MFA
Dynamic MFA triggers based on risk events	•		
Authentication and authorization policies	•		
Configurable based on location and device state	•		
Support for "report only" mode	•		
Ability to completely block users/services	•		

Microsoft Entra ID Free tier

All users in a Microsoft Entra ID Free tenant can use Microsoft Entra multifactor authentication by using security defaults. The mobile authentication app can be used for Microsoft Entra multifactor authentication when using Microsoft Entra ID Free security defaults.

- [Learn more about Microsoft Entra security defaults](#)
- [Enable security defaults for users in Microsoft Entra ID Free](#)

You enable Microsoft Entra multifactor authentication in one of the following ways, depending on the type of account you use:

- If you use a Microsoft Account, [register for multifactor authentication ↗](#).
- If you aren't using a Microsoft Account, [turn on multifactor authentication for a user or group in Microsoft Entra ID](#).

Next steps

- For more information on costs, see [Microsoft Entra pricing ↗](#).
- [What is Conditional Access](#)
- [What is Microsoft Entra ID Protection?](#)
- MFA can also be [enabled on a per-user basis](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Set up multifactor authentication for Microsoft 365

Article • 10/02/2024

Check out all of our small business content on [Small business help & learning](#).

Check out [Microsoft 365 small business help](#) on YouTube.

Multifactor authentication means you and your employees must provide more than one way to sign in to Microsoft 365 is one of the easiest ways to secure your business. Based on your understanding of [multifactor authentication \(MFA\) and its support in Microsoft 365](#), it's time to set it up and roll it out to your organization.

Multifactor authentication (MFA) is an important first step in securing your organization. Microsoft 365 for business gives you the option to use security defaults or Conditional Access policies to turn on MFA for your admins and user accounts. For most organizations, **Security defaults** offer a good level of sign-in security. But if your organization must meet more stringent requirements, you can use [Conditional Access policies](#).

Tip

If you need help with the steps in this topic, consider [working with a Microsoft small business specialist](#). With Business Assist, you and your employees get around-the-clock access to small business specialists as you grow your business, from onboarding to everyday use.

Before you begin

- You must be a Global admin to manage MFA. For more information, see [About admin roles](#).
- If you have legacy per-user MFA turned on, [Turn off legacy per-user MFA](#).
- Advanced: If you have third-party directory services with Active Directory Federation Services (AD FS), set up the Azure MFA Server. See [advanced scenarios with Microsoft Entra multifactor authentication and third-party VPN solutions](#) for more information.

Watch: Turn on multifactor authentication

<https://learn-video.azurefd.net/vod/player?id=eb0acd2a-edf5-4c1d-8e72-e3254bc7dc12&locale=en-us&embedUrl=%2Fmicrosoft-365%2Fadmin%2Fsecurity-and-compliance%2Fset-up-multi-factor-authentication>

Steps: Turn on multifactor authentication

If you purchased your subscription or trial after October 21, 2019, and you're prompted for MFA when you sign in, **security defaults** have been automatically enabled for your subscription. If you purchased your subscription before October 2019, follow these steps to turn on **security default MFA**.

1. Sign in to the [Microsoft Entra admin center](#) as least a **Security Administrator**.
2. Browse to **Identity > Overview > Properties**.
3. Select **Manage security defaults**.
4. Set **Security defaults** to **Enabled**.
5. Select **Save**.

For more information, see [What are security defaults?](#)

Turn off per-user MFA

If you've previously turned on per-user MFA, you must turn it off before enabling Security defaults. You should also turn off per-user MFA after you've configure your policies and settings in Conditional Access.

1. In the Microsoft 365 admin center, in the left nav choose **Users > Active users**.
2. On the **Active users** page, choose **multifactor authentication**.
3. On the multifactor authentication page, select each user and set their multifactor authentication status to **Disabled**.

Turn Security default MFA off

Important

It's not recommended to turn off MFA.

1. Sign in to the [Microsoft Entra admin center](#) as at least a **security administrator** or **conditional access administrator**.
2. Browse to **Identity > Overview > Properties**.
3. Select **Manage security defaults**.
4. Set **Security defaults** to **Disabled (not recommended)**.
5. Select **Save**.

Use Conditional Access policies

If your organization has more granular sign-in security needs, [Conditional Access policies](#) can offer you more control. Conditional Access lets you create and define policies that react to sign in events and request additional actions before a user is granted access to an application or service. You can also get started by using [conditional access templates](#).

Important

Do not forget to disable per-user MFA after you have enabled Conditional Access policies. This is important as it will result in inconsistent user experience.

Conditional Access is available for customers who bought Microsoft Entra ID P1, or licenses that include this, such as Microsoft 365 Business Premium, and Microsoft 365 E3. For more information, see [create a Conditional Access policy](#).

Risk-based conditional access is available through Microsoft Entra ID P2 license, or licenses that include risk based conditional access, like Microsoft 365 E5. For more information, see [risk-based Conditional Access](#).

For more information about the Microsoft Entra ID P1 and P2, see [Microsoft Entra pricing](#).

Next steps - Send to your users

- [What is multifactor authentication](#)
- [Sign-in after registration](#)
- [Change additional verification method](#)
- [Register for additional verification method](#)

Related content

[Set up multifactor authentication](#) (video)

[Turn on multifactor authentication for your phone](#) (article)

[Security defaults and multifactor authentication](#) (article)

Frequently asked questions about Microsoft Entra multifactor authentication

FAQ

This FAQ answers common questions about Microsoft Entra multifactor authentication and using the multifactor authentication service. It's broken down into questions about the service in general, billing models, user experiences, and troubleshooting.

Important

In September 2022, Microsoft announced deprecation of Multifactor Authentication Server. Beginning September 30, 2024, Multifactor Authentication Server deployments will no longer service multifactor authentication requests, which could cause authentications to fail for your organization. To ensure uninterrupted authentication services and to remain in a supported state, organizations should [migrate their users' authentication data](#) to the cloud-based Microsoft Entra multifactor authentication service by using the latest Migration Utility included in the most recent [MFA Server update](#). For more information, see [MFA Server Migration](#).

General

How does Azure Multifactor Authentication Server handle user data?

With Multifactor Authentication Server, user data is only stored on the on-premises servers. No persistent user data is stored in the cloud. When the user performs two-step verification, Multifactor Authentication Server sends data to the Microsoft Entra multifactor authentication cloud service for authentication. Communication between Multifactor Authentication Server and the multifactor authentication cloud service uses Secure Sockets Layer (SSL) or Transport Layer Security (TLS) over port 443 outbound.

When authentication requests are sent to the cloud service, data is collected for authentication and usage reports. The following data fields are included in two-step verification logs:

- **Unique ID** (either user name or on-premises Multifactor Authentication Server ID)
- **First and Last Name** (optional)
- **Email Address** (optional)
- **Phone Number** (when using a voice call or text message authentication)
- **Device Token** (when using mobile app authentication)
- **Authentication Mode**
- **Authentication Result**
- **Multifactor Authentication Server Name**
- **Multifactor Authentication Server IP**
- **Client IP** (if available)

The optional fields can be configured in Multifactor Authentication Server.

The verification result (success or denial), and the reason if it was denied, is stored with the authentication data. This data is available in authentication and usage reports.

For more information, see [Data residency and customer data for Microsoft Entra multifactor authentication](#).

What short codes are used for sending text messages to my users?

In the United States, we use the following short codes:

- 97671
- 69829
- 51789
- 99399

In Canada, we use the following short codes:

- 759731
- 673801

There's no guarantee of consistent text message or voice-based multifactor authentication prompt delivery by the same number. In the interest of our users, we may add or remove short codes at any time as we make route adjustments to improve text message deliverability.

We don't support short codes for countries or regions besides the United States and Canada.

Does Microsoft Entra multifactor authentication throttle user sign-ins?

Yes, in certain cases that typically involve repeated authentication requests in a short time window, Microsoft Entra multifactor authentication throttles user sign-in attempts to protect telecommunication networks, mitigate MFA fatigue-style attacks and protect its own systems for the benefit of all customers.

Although we don't share specific throttling limits, they're based around reasonable usage.

Is my organization charged for sending the phone calls and text messages that are used for authentication?

No, you're not charged for individual phone calls placed or text messages sent to users through Microsoft Entra multifactor authentication. If you use a per-authentication MFA provider, you're billed for each authentication, but not for the method used.

Your users might be charged for the phone calls or text messages they receive, according to their personal phone service.

Does the per-user billing model charge me for all enabled users, or just the ones that performed two-step verification?

Billing is based on the number of users configured to use multifactor authentication, regardless of whether they performed two-step verification that month.

How does multifactor authentication billing work?

When you create a per-user or per-authentication MFA provider, your organization's Azure subscription is billed monthly based on usage. This billing model is similar to how Azure bills for usage of virtual machines and Web Apps.

When you purchase a subscription for Microsoft Entra multifactor authentication, your organization only pays the annual license fee for each user. MFA licenses and Microsoft

365, Microsoft Entra ID P1 or P2, or Enterprise Mobility + Security bundles are billed this way.

For more information, see [How to get Microsoft Entra multifactor authentication](#).

Is there a free version of Microsoft Entra multifactor authentication?

Security defaults can be enabled in the Microsoft Entra ID Free tier. With security defaults, all users are enabled for multifactor authentication using the Microsoft Authenticator app. There's no ability to use text message or phone verification with security defaults, just the Microsoft Authenticator app.

For more information, see [What are security defaults?](#)

Can my organization switch between per-user and per-authentication consumption billing models at any time?

If your organization purchases MFA as a standalone service with consumption-based billing, you choose a billing model when you create an MFA provider. You can't change the billing model after an MFA provider is created.

If your MFA provider is *not* linked to a Microsoft Entra tenant, or you link the new MFA provider to a different Microsoft Entra tenant, user settings, and configuration options aren't transferred. Also, existing MFA Servers need to be reactivated using activation credentials generated through the new MFA Provider. Reactivating the MFA Servers to link them to the new MFA Provider doesn't impact phone call and text message authentication, but mobile app notifications stop working for all users until they reactivate the mobile app.

Learn more about MFA providers in [Getting started with an Azure multifactor authentication provider](#).

Can my organization switch between consumption-based billing and subscriptions (a license-based model) at any time?

In some instances, yes.

If your directory has a *per-user* Microsoft Entra multifactor authentication provider, you can add MFA licenses. Users with licenses aren't counted in the per-user consumption-based billing. Users without licenses can still be enabled for MFA through the MFA provider. If you purchase and assign licenses for all your users configured to use multifactor authentication, you can delete the Microsoft Entra multifactor authentication provider. You can always create another per-user MFA provider if you have more users than licenses in the future.

If your directory has a *per-authentication* Microsoft Entra multifactor authentication provider, you're always billed for each authentication, as long as the MFA provider is linked to your subscription. You can assign MFA licenses to users, but you'll still be billed for every two-step verification request, whether it comes from someone with an MFA license assigned or not.

Does my organization have to use and synchronize identities to use Microsoft Entra multifactor authentication?

If your organization uses a consumption-based billing model, Microsoft Entra ID is optional, but not required. If your MFA provider isn't linked to a Microsoft Entra tenant, you can only deploy Azure Multifactor Authentication Server on-premises.

Microsoft Entra ID is required for the license model because licenses are added to the Microsoft Entra tenant when you purchase and assign them to users in the directory.

Manage and support user accounts

What should I tell my users to do if they don't receive a response on their phone?

Have your users attempt up to five times in 5 minutes to get a phone call or text message for authentication. Microsoft uses multiple providers for delivering calls and text messages. If this approach doesn't work, open a support case to troubleshoot further.

Third-party security apps may also block the verification code text message or phone call. If using a third-party security app, try disabling the protection, then request another MFA verification code be sent.

If the prior steps don't work, check if users are configured for more than one verification method. Try signing in again, but select a different verification method on the sign-in page.

For more information, see the [end-user troubleshooting guide](#).

What should I do if one of my users can't get in to their account?

You can reset the user's account by making them go through the registration process again. Learn more about [managing user and device settings with Microsoft Entra multifactor authentication in the cloud](#).

What should I do if one of my users loses a phone that is using app passwords?

To prevent unauthorized access, delete all the user's app passwords. After the user has a replacement device, they can recreate the passwords. Learn more about [managing user and device settings with Microsoft Entra multifactor authentication in the cloud](#).

What if a user can't sign in to nonbrowser apps?

If your organization still uses legacy clients, and you [allowed the use of app passwords](#), then your users can't sign in to these legacy clients with their username and password. Instead, they need to [set up app passwords](#). Your users must clear (delete) their sign-in information, restart the app, and then sign in with their username and *app password* instead of their regular password.

If your organization doesn't have legacy clients, you shouldn't allow your users to create app passwords.

Note

Modern authentication for Office 2013 clients

App passwords are only necessary for apps that don't support modern authentication. Office 2013 clients support modern authentication protocols, but need to be configured. Modern authentication is available to any customer running the March 2015 or later update for Office 2013. For more information, see the blog post [Updated Office 365 modern authentication](#).

My users say that sometimes they don't receive the text message or the verification times out.

Delivery of text messages isn't guaranteed because uncontrollable factors might affect the reliability of the service. These factors include the destination country or region, the mobile phone carrier, and the signal strength.

Third-party security apps may also block the verification code text message or phone call. If using a third-party security app, try disabling the protection, then request another MFA verification code be sent.

If your users often have problems with reliably receiving text messages, tell them to use the Microsoft Authenticator app or phone call method instead. The Microsoft Authenticator can receive notifications both over cellular and Wi-Fi connections. In addition, the mobile app can generate verification codes even when the device has no signal at all. The Microsoft Authenticator app is available for [Android ↗](#), [iOS ↗](#), and [Windows Phone ↗](#).

Can I change the amount of time my users have to enter the verification code from a text message before the system times out?

In some cases, yes.

For one-way SMS with MFA Server v7.0 or higher, you can configure the timeout setting by setting a registry key. After the MFA cloud service sends the text message, the verification code (or one-time passcode) is returned to the MFA Server. The MFA Server stores the code in memory for 300 seconds by default. If the user doesn't enter the code before the 300 seconds have passed, their authentication is denied. Use these steps to change the default timeout setting:

1. Go to `HKLM\Software\Wow6432Node\Positive Networks\PhoneFactor`.
2. Create a **DWORD** registry key called `pfsvc_pendingSmsTimeoutSeconds` and set the time in seconds that you want the MFA Server to store one-time passcodes.

💡 Tip

If you have multiple MFA Servers, only the one that processed the original authentication request knows the verification code that was sent to the user. When the user enters the code, the authentication request to validate it must be sent to

the same server. If the code validation is sent to a different server, the authentication is denied.

If users don't respond to the SMS within the defined timeout period, their authentication is denied.

For one-way SMS with Microsoft Entra multifactor authentication in the cloud (including the AD FS adapter or the Network Policy Server extension), you can't configure the timeout setting. Microsoft Entra ID stores the verification code for 180 seconds.

Can I use hardware tokens with Multifactor Authentication Server?

If you're using Multifactor Authentication Server, you can import third-party Open Authentication (OATH) time-based, one-time password (TOTP) tokens, and then use them for two-step verification.

You can use ActiveIdentity tokens that are OATH TOTP tokens if you put the secret key in a CSV file and import to Multifactor Authentication Server. You can use OATH tokens with Active Directory Federation Services (ADFS), Internet Information Server (IIS) forms-based authentication, and Remote Authentication Dial-In User Service (RADIUS) as long as the client system can accept the user input.

You can import third-party OATH TOTP tokens with the following formats:

- Portable Symmetric Key Container (PSKC)
- CSV if the file contains a serial number, a secret key in Base 32 format, and a time interval

Can I use Multifactor Authentication Server to secure Terminal Services?

Yes, but if you're using Windows Server 2012 R2 or later, you can only secure Terminal Services by using Remote Desktop Gateway (RD Gateway).

Security changes in Windows Server 2012 R2 changed how Multifactor Authentication Server connects to the Local Security Authority (LSA) security package in Windows Server 2012 and earlier versions. For versions of Terminal Services in Windows Server 2012 or earlier, you can [secure an application with Windows Authentication](#). If you're using Windows Server 2012 R2, you need RD Gateway.

I configured Caller ID in MFA Server, but my users still receive multifactor authentication calls from an anonymous caller.

When multifactor authentication calls are placed through the public telephone network, sometimes they're routed through a carrier that doesn't support caller ID. Because of this carrier behavior, caller ID isn't guaranteed, even though the multifactor authentication system always sends it.

Why are my users being prompted to register their security information?

There are several reasons that users could be prompted to register their security information:

- The user has been enabled for MFA by their administrator in Microsoft Entra ID, but doesn't have security information registered for their account yet.
- The user has been enabled for self-service password reset in Microsoft Entra ID. The security information will help them reset their password in the future if they ever forget it.
- The user accessed an application that has a Conditional Access policy to require MFA and hasn't previously registered for MFA.
- The user is registering a device with Microsoft Entra ID (including Microsoft Entra join), and your organization requires MFA for device registration, but the user hasn't previously registered for MFA.
- The user is generating Windows Hello for Business in Windows 10 (which requires MFA) and hasn't previously registered for MFA.
- The organization has created and enabled an MFA Registration policy that has been applied to the user.
- The user previously registered for MFA, but chose a verification method that an administrator has since disabled. The user must therefore go through MFA registration again to select a new default verification method.

Errors

What should users do if they see an "Authentication request isn't for an activated

account" error message when using mobile app notifications?

Ask the user to complete the following procedure to remove their account from the Microsoft Authenticator, then add it again:

1. Go to [their account profile](#) and sign in with an organizational account.
2. Select **Additional Security Verification**.
3. Remove the existing account from the Microsoft Authenticator app.
4. Select **Configure**, and then follow the instructions to reconfigure the Microsoft Authenticator.

What should users do if they see a 0x800434D4L error message when signing in to a nonbrowser application?

The *0x800434D4L* error occurs when you try to sign in to a nonbrowser application, installed on a local computer, that doesn't work with accounts that require two-step verification.

A workaround for this error is to have separate user accounts for admin-related and nonadmin operations. Later, you can link mailboxes between your admin account and nonadmin account so that you can sign in to Outlook by using your nonadmin account. For more details about this solution, learn how to [give an administrator the ability to open and view the contents of a user's mailbox](#).

What are the possible reasons why a user fails, with the error code "LsaLogonUser failed with NTSTATUS -1073741715 for MFA Server"?

Error 1073741715 = Status Logon Failure -> The attempted logon is invalid. This is due to either a bad username or authentication.

A plausible reason for this error: If the primary credentials entered are correct, there might be a mismatch between the supported NTLM version on the MFA server and the domain controller. MFA Server supports only NTLMv1 (LmCompatibilityLevel=1 through 4) and not NTLMv2 (LmCompatibilityLevel=5).

Next steps

If your question isn't answered here, the following support options are available:

- Search the [Microsoft Support Knowledge Base](#) for solutions to common technical issues.
 - Search for and browse technical questions and answers from the community, or ask your own question in the [Microsoft Entra Q&A](#).
 - Contact Microsoft professional through [Multifactor Authentication Server support](#). When contacting us, it's helpful if you can include as much information about your issue as possible. Information you can supply includes the page where you saw the error, the specific error code, the specific session ID, and the ID of the user who saw the error.
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Combined password policy and check for weak passwords in Microsoft Entra ID

Article • 03/04/2025

Beginning in October 2021, Microsoft Entra validation for compliance with password policies also includes a check for [known weak passwords](#) and their variants. This article explains details about the password policy criteria checked by Microsoft Entra ID.

Microsoft Entra password policies

A password policy is applied to all user and admin accounts that are created and managed directly in Microsoft Entra ID. You can [ban weak passwords](#) and define parameters to [lock out an account](#) after repeated bad password attempts. Other password policy settings can't be modified.

The Microsoft Entra password policy doesn't apply to user accounts synchronized from an on-premises AD DS environment using Microsoft Entra Connect unless you enable `EnforceCloudPasswordPolicyForPasswordSyncedUsers`. If `EnforceCloudPasswordPolicyForPasswordSyncedUsers` and password writeback are enabled, Microsoft Entra password expiration policy applies, but the on-premises password policy takes precedence for length, complexity, and so on.

The following Microsoft Entra password policy requirements apply for all passwords that are created, changed, or reset in Microsoft Entra ID. Requirements are applied during user provisioning, password change, and password reset flows. You can't change these settings except as noted.

[] [Expand table](#)

Property	Requirements
Characters allowed	Uppercase characters (A - Z) Lowercase characters (a - z) Numbers (0 - 9) Symbols: - @ # \$ % ^ & * - _ ! + = [] { } \ : ' . ? / ` ~ " () ; < > - blank space
Characters not allowed	Unicode characters

Property	Requirements
Password length	<p>Passwords require</p> <ul style="list-style-type: none"> - A minimum of eight characters - A maximum of 256 characters
Password complexity	<p>Passwords require three out of four of the following categories:</p> <ul style="list-style-type: none"> - Uppercase characters - Lowercase characters - Numbers - Symbols <p>Note: Password complexity check isn't required for Education tenants.</p>
Password not recently used	When a user changes their password, the new password shouldn't be the same as the current password.
Password isn't banned by Microsoft Entra Password Protection	The password can't be on the global list of banned passwords for Microsoft Entra Password Protection, or on the customizable list of banned passwords specific to your organization.

Password expiration policies

Password expiration policies are unchanged but they're included in this article for completeness. Those assigned at least the [User Administrator](#) role can use the [Microsoft Graph PowerShell cmdlets](#) to set user passwords not to expire.

Note

By default, only passwords for user accounts that aren't synchronized through Microsoft Entra Connect can be configured to not expire. For more information about directory synchronization, see [Connect AD with Microsoft Entra ID](#).

You can also use PowerShell to remove the never-expires configuration, or to see user passwords that are set to never expire.

The following expiration requirements apply to other providers that use Microsoft Entra ID for identity and directory services, such as Microsoft Intune and Microsoft 365.

 Expand table

Property	Requirements
Password expiry duration (Maximum password age)	<p>Default value: 90 days.</p> <p>The value is configurable by using the Update-MgDomain</p>

Property	Requirements
	cmdlet from the Microsoft Graph PowerShell module.
Password expiry (Let passwords never expire)	Default value: false (indicates that password's have an expiration date). The value can be configured for individual user accounts by using the Update-MgUser cmdlet.

Next steps

- [Password policies and account restrictions in Microsoft Entra ID](#)
 - [Eliminate bad passwords using Microsoft Entra Password Protection](#)
-

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Eliminate bad passwords using Microsoft Entra Password Protection

Article • 03/04/2025

As a general rule, security guidance recommends that you don't use the same password in multiple places, to make it complex, and to avoid simple passwords like *Password123*. You can provide your users with [guidance on how to choose passwords](#), but weak or insecure passwords are often still used. Microsoft Entra Password Protection detects and blocks known weak passwords and their variants, and can also block other weak terms that are specific to your organization.

With Microsoft Entra Password Protection, default global banned password lists are automatically applied to all users in a Microsoft Entra tenant. To support your own business and security needs, you can define entries in a custom banned password list. When users change or reset their passwords, these banned password lists are checked to enforce the use of strong passwords.

You should use other features like [Microsoft Entra multifactor authentication](#), not just rely on strong passwords enforced by Microsoft Entra Password Protection. For more information on using multiple layers of security for your sign-in events, see [Your Pa\\$\\$word doesn't matter](#).

Important

This conceptual article explains to an administrator how Microsoft Entra Password Protection works. If you're an end user already registered for self-service password reset and need to get back into your account, go to <https://aka.ms/sspr>.

If your IT team hasn't enabled the ability to reset your own password, reach out to your helpdesk.

Global banned password list

The Microsoft Entra ID Protection team constantly analyzes Microsoft Entra security telemetry data looking for commonly used weak or compromised passwords. Specifically, the analysis looks for base terms that often are used as the basis for weak passwords. When weak terms are found, they're added to the *global banned password list*. The contents of the global banned password list aren't based on any external data source, but on the results of Microsoft Entra security telemetry and analysis.

When a password is changed or reset for any user in a Microsoft Entra tenant, the current version of the global banned password list is used to validate the strength of the password. This validation check results in stronger passwords for all Microsoft Entra customers.

The global banned password list is automatically applied to all users in a Microsoft Entra tenant. There's nothing to enable or configure, and can't be disabled. This global banned password list is applied to users when they change or reset their own password through Microsoft Entra ID.

ⓘ Note

Cyber-criminals also use similar strategies in their attacks to identify common weak passwords and variations. To improve security, Microsoft doesn't publish the contents of the global banned password list.

Custom banned password list

Some organizations want to improve security and add their own customizations on top of the global banned password list. To add your own entries, you can use the *custom banned password list*. Terms added to the custom banned password list should be focused on organizational-specific terms such as the following examples:

- Brand names
- Product names
- Locations, such as company headquarters
- Company-specific internal terms
- Abbreviations that have specific company meaning

When terms are added to the custom banned password list, they're combined with the terms in the global banned password list. Password change or reset events are then validated against the combined set of these banned password lists.

ⓘ Note

The custom banned password list is limited to a maximum of 1,000 terms. It isn't designed for blocking extremely large lists of passwords.

To fully apply the benefits of the custom banned password list, first understand [how are passwords evaluated](#) before you add terms to the custom banned list.

This approach lets you efficiently detect and block large numbers of weak passwords and their variants.

The screenshot shows the 'Authentication methods | Password protection' page in the Microsoft Entra ID Security portal. The left sidebar has sections for 'Manage' (Policies, Password protection, Registration campaign, Authentication strengths, Settings) and 'Monitoring' (Activity, User registration details, Registration and reset events, Bulk operation results). The main area shows 'Custom smart lockout' settings (Lockout threshold: 10, Lockout duration in seconds: 60) and a 'Custom banned passwords' section. The 'Custom banned password list' section is highlighted with a red box. It contains a table with columns 'Enforce custom list' (Yes) and 'Custom banned password list' (contoso, fabrikam, tailwind, michigan, wolverine, harbaugh, howard). Below this is a 'Password protection for Windows Server Active Directory' section with an 'Enable password protection on Windows Server Active Directory' switch set to 'Yes'. At the bottom, there's a 'Mode' switch between 'Enforced' and 'Audit'.

Let's consider a customer named *Contoso*. The company is based in London and makes a product named *Widget*. For this example customer, it would be wasteful and less secure to try to block specific variations of these terms:

- "Contoso!1"
- "Contoso@London"
- "ContosoWidget"
- "!Contoso"
- "LondonHQ"

Instead, it's much more efficient and secure to block only the key base terms, such as the following examples:

- "Contoso"
- "London"
- "Widget"

The password validation algorithm then automatically blocks weak variants and combinations.

To get started with using a custom banned password list, complete the following tutorial:

[Tutorial: Configure custom banned passwords](#)

Password spray attacks and third-party compromised password lists

Microsoft Entra Password Protection helps you defend against password spray attacks. Most password spray attacks don't attempt to attack any given individual account more than a few times. This behavior would increase the likelihood of detection, either via account lockout or other means.

Instead, most password spray attacks submit only a few of the known weakest passwords against each of the accounts in an enterprise. This technique allows the attacker to quickly search for an easily compromised account and avoid potential detection thresholds.

Microsoft Entra Password Protection efficiently blocks all known weak passwords likely to be used in password spray attacks. This protection is based on real-world security telemetry data from Microsoft Entra ID to build the global banned password list.

There are some third-party websites that enumerate millions of passwords that have been compromised in previous publicly known security breaches. It's common for third-party password validation products to be based on brute-force comparison against those millions of passwords. However, those techniques aren't the best way to improve overall password strength given the typical strategies used by password spray attackers.

ⓘ Note

The global banned password list isn't based on any third-party data sources, including compromised password lists.

Although the global banned list is small in comparison to some third-party bulk lists, it's sourced from real-world security telemetry on actual password spray attacks. This approach improves the overall security and effectiveness, and the password validation algorithm also uses smart fuzzy-matching techniques. As a result, Microsoft Entra Password Protection efficiently detects and blocks millions of the most common weak passwords from being used in your enterprise.

On-premises hybrid scenarios

Many organizations have a hybrid identity model that includes on-premises Active Directory Domain Services (AD DS) environments. To extend the security benefits of Microsoft Entra Password Protection into your AD DS environment, you can install components on your on-premises servers. These agents require password change

events in the on-premises AD DS environment to comply with the same password policy as in Microsoft Entra ID.

For more information, see [Enforce Microsoft Entra Password Protection for AD DS](#).

How are passwords evaluated

When a user changes or resets their password, the new password is checked for strength and complexity by validating it against the combined list of terms from the global and custom banned password lists.

Even if a user's password contains a banned password, the password may be accepted if the overall password is otherwise strong enough. A newly configured password goes through the following steps to assess its overall strength to determine if it should be accepted or rejected.

 **Note**

Password protection in Microsoft Entra ID doesn't correlate with password protection for on-premises users. The validations in password protection aren't consistent for users across the two services. Ensure the users in your tenant meet the required password parameters for their respective service when initially setting their password or completing SSPR.

Step 1: Normalization

A new password first goes through a normalization process. This technique allows for a small set of banned passwords to be mapped to a much larger set of potentially weak passwords.

Normalization has the following two parts:

- All uppercase letters are changed to lower case.
- Then, common character substitutions are performed, such as in the following example:

 [Expand table](#)

Original letter	Substituted letter
0	o

Original letter	Substituted letter
1	I
\$	s
@	a

Consider the following example:

- The password "blank" is banned.
- A user tries to change their password to "Bl@nK".
- Even though "Bl@nk" isn't banned, the normalization process converts this password to "blank".
- This password would be rejected.

Step 2: Check if password is considered banned

A password is then examined for other matching behavior, and a score is generated. This final score determines if the password change request is accepted or rejected.

Fuzzy matching behavior

Fuzzy matching is used on the normalized password to identify if it contains a password found on either the global or the custom banned password lists. The matching process is based on an edit distance of one (1) comparison.

Consider the following example:

- The password "abcdef" is banned.
- A user tries to change their password to one of the following:
 - 'abcdeg' - *last character changed from 'f' to 'g'*
 - 'abcdefg' - *'g' appended to end*
 - 'abcde' - *trailing 'f' was deleted from end*
- Each of the above passwords doesn't specifically match the banned password "abcdef".

However, since each example is within an edit distance of 1 of the banned term 'abcdef', they're all considered as a match to "abcdef".

- These passwords would be rejected.

Substring matching (on specific terms)

Substring matching is used on the normalized password to check for the user's first and last name, and the tenant name. Tenant name matching isn't done when validating passwords on an AD DS domain controller for on-premises hybrid scenarios.

Important

Substring matching is only enforced for names, and other terms, that are at least four characters long.

Consider the following example:

- A user named Poll who wants to reset their password to "p0LL23fb".
- After normalization, this password would become "poll23fb".
- Substring matching finds that the password contains the user's first name "Poll".
- Even though "poll23fb" wasn't specifically on either banned password list, substring matching found "Poll" in the password.
- This password would be rejected.

Score Calculation

The next step is to identify all instances of banned passwords in the user's normalized new password. Points are assigned based on the following criteria:

1. Each banned password found in a user's password is given one point.
2. Each remaining character that isn't part of a banned password is given one point.
3. A password must be at least five (5) points to be accepted.

For the next two example scenarios, Contoso is using Microsoft Entra Password Protection and has "contoso" on their custom banned password list. Let's also assume that "blank" is on the global list.

In the following example scenario, a user changes their password to "C0ntos0Blank12":

- After normalization, this password becomes "contosoblank12".
- The matching process finds that this password contains two banned passwords: "contoso" and "blank".
- This password is then given the following score:

$$[\text{contoso}] + [\text{blank}] + [1] + [2] = 4 \text{ points}$$

- As this password is under five (5) points, it's rejected.

Let's look a slightly different example to show how more complexity in a password can build the required number of points to be accepted. In the following example scenario, a user changes their password to "ContoS0Bl@nkf9!":

- After normalization, this password becomes "contosoblankf9!".
- The matching process finds that this password contains two banned passwords: "contoso" and "blank".
- This password is then given the following score:

$$[contoso] + [blank] + [f] + [9] + [!] = 5 \text{ points}$$

- As this password is at least five (5) points, it's accepted.

Important

The banned password algorithm, along with the global banned password list, can and do change at any time in Azure based on ongoing security analysis and research.

For the on-premises DC agent service in hybrid scenarios, updated algorithms only take effect after the DC agent software is upgraded.

What do users see

When a user attempts to reset or change a password to something that would be banned, one of the following error messages are displayed:

"Unfortunately, your password contains a word, phrase, or pattern that makes your password easily guessable. Please try again with a different password."

"We've seen that password too many times before. Choose something harder to guess."

"Choose a password that's harder for people to guess."

License requirements

 Expand table

Users	Microsoft Entra Password Protection with global banned password list	Microsoft Entra Password Protection with custom banned password list
Cloud-only users	Microsoft Entra ID Free	Microsoft Entra ID P1 or P2
Users synchronized from on-premises AD DS	Microsoft Entra ID P1 or P2	Microsoft Entra ID P1 or P2

 **Note**

On-premises AD DS users that aren't synchronized to Microsoft Entra ID also benefit from Microsoft Entra Password Protection based on existing licensing for synchronized users.

For more information about licensing, see [Microsoft Entra pricing site](#).

Next steps

To get started with using a custom banned password list, complete the following tutorial:

[Tutorial: Configure custom banned passwords](#)

You can also then [enable on-premises Microsoft Entra Password Protection](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Enforce on-premises Microsoft Entra Password Protection for Active Directory Domain Services

Article • 03/04/2025

Microsoft Entra Password Protection detects and blocks known weak passwords and their variants, and can also block additional weak terms that are specific to your organization. On-premises deployment of Microsoft Entra Password Protection uses the same global and custom banned password lists that are stored in Microsoft Entra ID, and does the same checks for on-premises password changes as Microsoft Entra ID does for cloud-based changes. These checks are performed during password changes and password reset events against on-premises Active Directory Domain Services (AD DS) domain controllers.

Design principles

Microsoft Entra Password Protection is designed with the following principles in mind:

- Domain controllers (DCs) never have to communicate directly with the internet.
- No new network ports are opened on DCs.
- No AD DS schema changes are required. The software uses the existing AD DS *container* and *serviceConnectionPoint* schema objects.
- Any supported AD DS domain or forest functional level can be used.
- The software doesn't create or require accounts in the AD DS domains that it protects.
- User clear-text passwords never leave the domain controller, either during password validation operations or at any other time.
- The software isn't dependent on other Microsoft Entra features. For example, Microsoft Entra password hash sync (PHS) isn't related or required for Microsoft Entra Password Protection.
- Incremental deployment is supported, however the password policy is only enforced where the Domain Controller Agent (DC Agent) is installed.

Incremental deployment

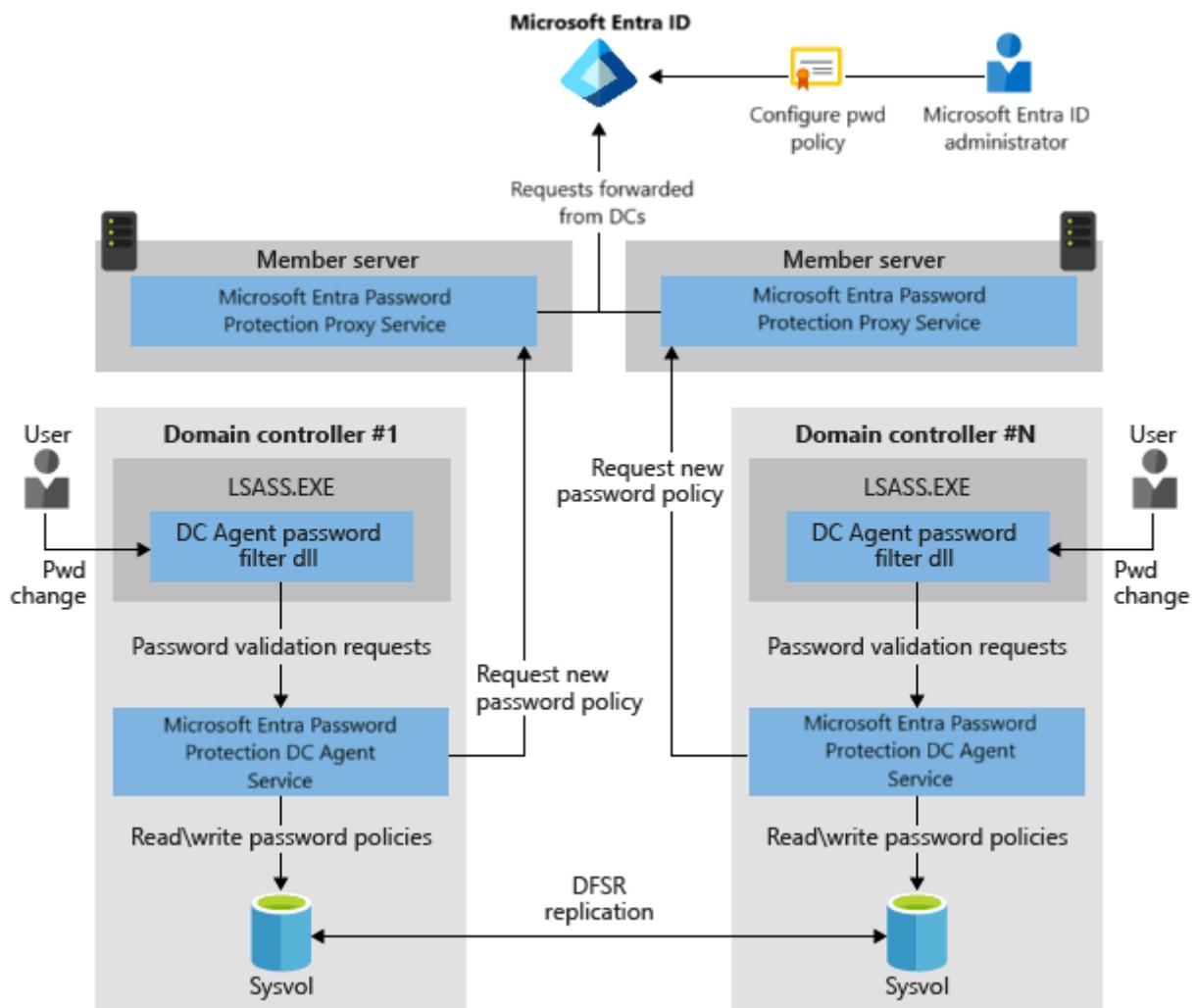
Microsoft Entra Password Protection supports incremental deployment across DCs in an AD DS domain. It's important to understand what this really means and what the tradeoffs are.

The Microsoft Entra Password Protection DC agent software can only validate passwords when it's installed on a DC, and only for password changes that are sent to that DC. It's not possible to control which DCs are chosen by Windows client machines for processing user password changes. To guarantee consistent behavior and universal Microsoft Entra Password Protection security enforcement, the DC agent software must be installed on all DCs in a domain.

Many organizations want to carefully test Microsoft Entra Password Protection on a subset of their DCs prior to a full deployment. To support this scenario, Microsoft Entra Password Protection supports partial deployment. The DC agent software on a given DC actively validates passwords even when other DCs in the domain don't have the DC agent software installed. Partial deployments of this type aren't secure and aren't recommended other than for testing purposes.

Architectural diagram

It's important to understand the underlying design and function concepts before you deploy Microsoft Entra Password Protection in an on-premises AD DS environment. The following diagram shows how the components of Microsoft Entra Password Protection work together:



- The Microsoft Entra Password Protection Proxy service runs on any domain-joined machine in the current AD DS forest. The service's primary purpose is to forward password policy download requests from DCs to Microsoft Entra ID and then return the responses from Microsoft Entra ID to the DC.
- The password filter DLL of the DC Agent receives user password-validation requests from the operating system. The filter forwards them to the DC Agent service that's running locally on the DC.
- The DC Agent service of Microsoft Entra Password Protection receives password-validation requests from the password filter DLL of the DC Agent. The DC Agent service processes them by using the current (locally available) password policy and returns the result of *pass* or *fail*.

How Microsoft Entra Password Protection works

The on-premises Microsoft Entra Password Protection components work as follows:

1. Each Microsoft Entra Password Protection Proxy service instance advertises itself to the DCs in the forest by creating a *serviceConnectionPoint* object in Active Directory.

Each DC Agent service for Microsoft Entra Password Protection also creates a *serviceConnectionPoint* object in Active Directory. This object is used primarily for reporting and diagnostics.
2. The DC Agent service is responsible for initiating the download of a new password policy from Microsoft Entra ID. The first step is to locate a Microsoft Entra Password Protection Proxy service by querying the forest for proxy *serviceConnectionPoint* objects.
3. When an available proxy service is found, the DC Agent sends a password policy download request to the proxy service. The proxy service in turn sends the request to Microsoft Entra ID, then returns the response to the DC Agent service.
4. After the DC Agent service receives a new password policy from Microsoft Entra ID, the service stores the policy in a dedicated folder at the root of its domain *sysvol* folder share. The DC Agent service also monitors this folder in case newer policies replicate in from other DC Agent services in the domain.
5. The DC Agent service always requests a new policy at service startup. After the DC Agent service is started, it checks the age of the current locally available policy hourly. If the policy is older than one hour, the DC Agent requests a new policy.

from Microsoft Entra ID via the proxy service, as described previously. If the current policy isn't older than one hour, the DC Agent continues to use that policy.

6. When password change events are received by a DC, the cached policy is used to determine if the new password is accepted or rejected.

Key considerations and features

- Whenever a Microsoft Entra Password Protection password policy is downloaded, that policy is specific to a tenant. In other words, password policies are always a combination of the Microsoft global banned-password list and the per-tenant custom banned-password list.
- The DC Agent communicates with the proxy service via RPC over TCP. The proxy service listens for these calls on a dynamic or static RPC port, depending on the configuration.
- The DC Agent never listens on a network-available port.
- The proxy service never calls the DC Agent service.
- The proxy service is stateless. It never caches policies or any other state downloaded from Azure.
- Proxy registration works by adding credentials to the AADPasswordProtectionProxy Service Principal. Don't be alarmed by any events in the audit logs when this occurs.
- The DC Agent service always uses the most recent locally available password policy to evaluate a user's password. If no password policy is available on the local DC, the password is automatically accepted. When that happens, an event message is logged to warn the administrator.
- Microsoft Entra Password Protection isn't a real-time policy application engine. There can be a delay between when a password policy configuration change is made in Microsoft Entra ID and when that change reaches and is enforced on all DCs.
- Microsoft Entra Password Protection acts as a supplement to the existing AD DS password policies, not a replacement. This includes any other 3rd-party password filter dlls that may be installed. AD DS always requires that all password validation components agree before accepting a password.

Forest / tenant binding for Microsoft Entra Password Protection

Deployment of Microsoft Entra Password Protection in an AD DS forest requires registration of that forest with Microsoft Entra ID. Each proxy service that's deployed

must also be registered with Microsoft Entra ID. These forest and proxy registrations are associated with a specific Microsoft Entra tenant, which is identified implicitly by the credentials that are used during registration.

The AD DS forest and all deployed proxy services within a forest must be registered with the same tenant. It's not supported to have an AD DS forest or any proxy services in that forest being registered to different Microsoft Entra tenants. Symptoms of such a mis-configured deployment include the inability to download password policies.

 **Note**

Customers that have multiple Microsoft Entra tenants must therefore choose one distinguished tenant to register each forest for Microsoft Entra Password Protection purposes.

Download

The two required agent installers for Microsoft Entra Password Protection are available from the [Microsoft Download Center](#).

Next steps

To get started with using on-premises Microsoft Entra Password Protection, complete the following how-to:

[Deploy on-premises Microsoft Entra Password Protection](#)

Feedback

Was this page helpful?

 Yes

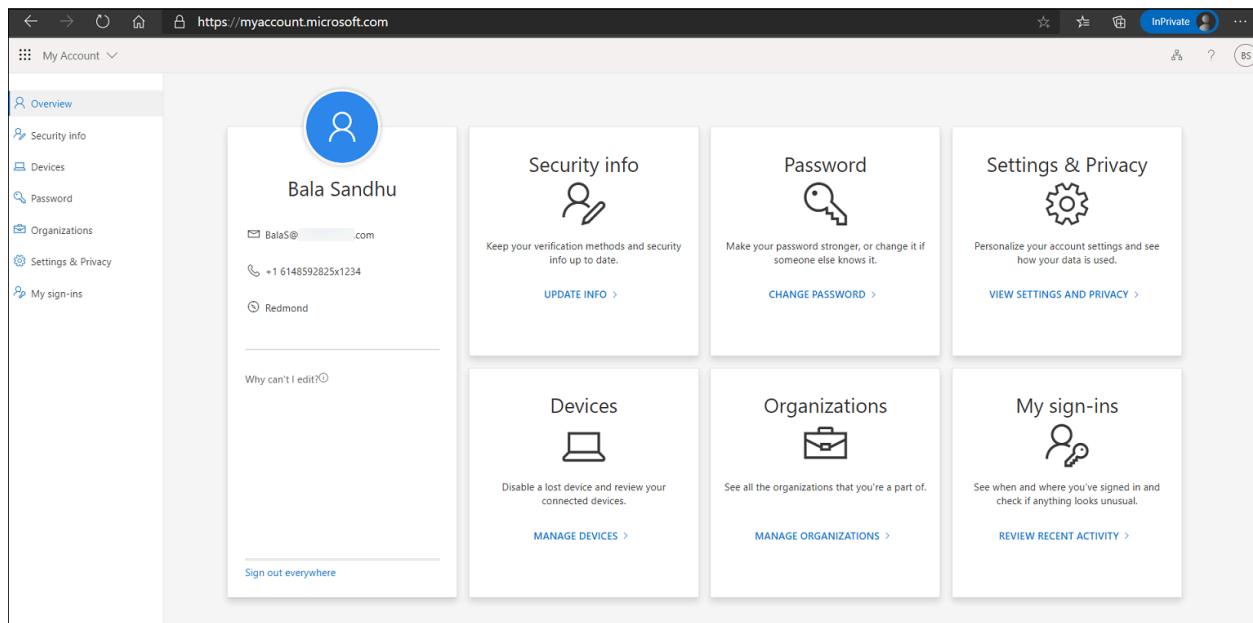
 No

[Provide product feedback](#)

Combined security information registration for Microsoft Entra overview

Article • 03/04/2025

Before combined registration, users registered authentication methods for Microsoft Entra multifactor authentication and self-service password reset (SSPR) separately. People were confused that similar methods were used for multifactor authentication and SSPR but they had to register for both features. Now, with combined registration, users can register once and get the benefits of both multifactor authentication and SSPR. We recommend this video on [How to enable and configure SSPR in Microsoft Entra ID](#).

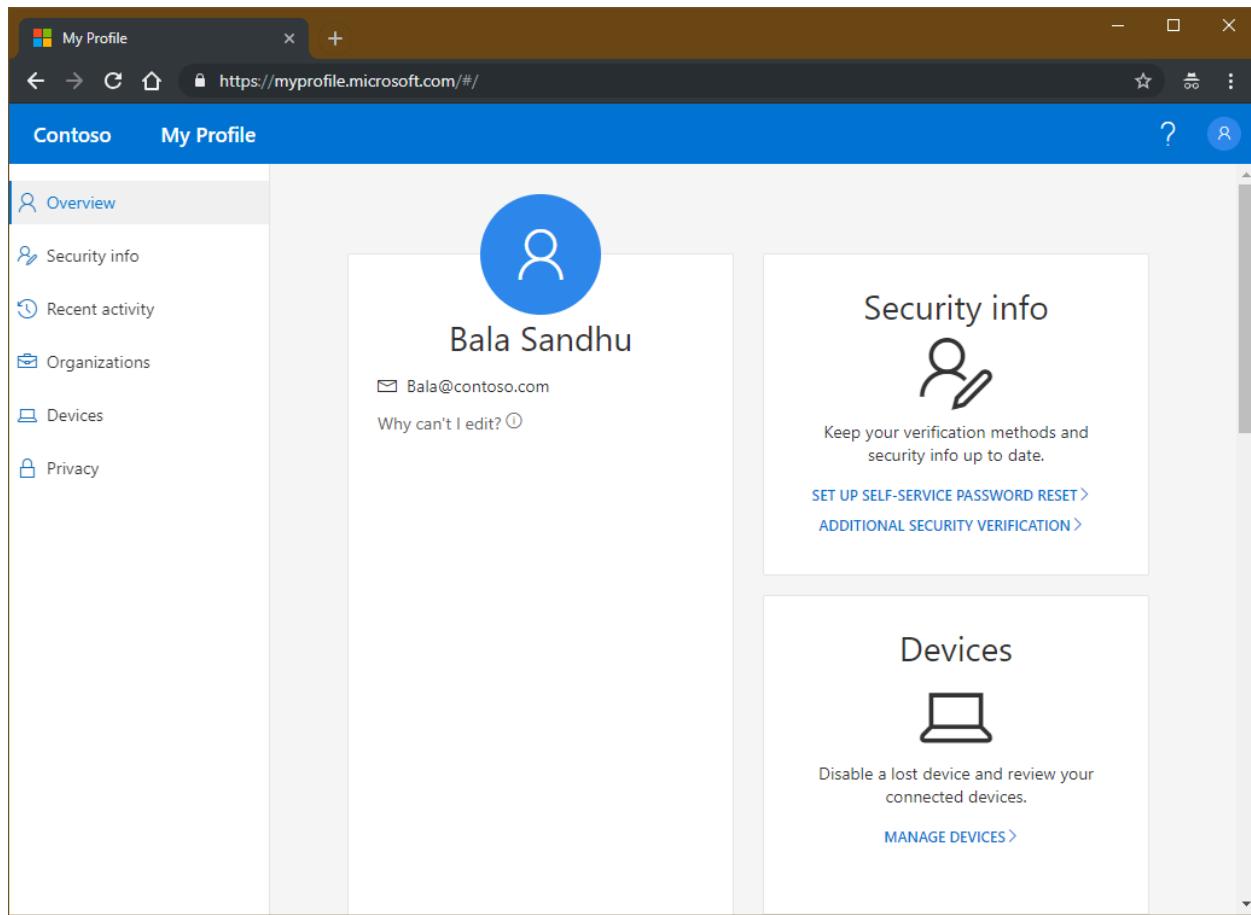


Before enabling the new experience, review this administrator-focused documentation and the user-focused documentation to ensure you understand the functionality and effect of this feature. Base your training on the [user documentation](#) to prepare your users for the new experience and help to ensure a successful rollout.

Combined registration is rolled out to all customers in Azure and Azure for US Government. The portal control that allows you to switch from legacy to combined registration experience is removed after your tenant migrates to the combined registration.

My Account pages are localized based on the language settings of the computer accessing the page. Microsoft stores the most recent language used in the browser cache, so subsequent attempts to access the pages continue to render in the last language used. If you clear the cache, the pages re-render.

If you want to force a specific language, you can add `?l=ng=<language>` to the end of the URL, where `<language>` is the code of the language you want to render.

A screenshot of a web browser window titled "My Profile". The URL in the address bar is https://myprofile.microsoft.com/#/. The main content area shows a profile card for "Bala Sandhu" with a blue circular icon, an email address "Bala@contoso.com", and a link "Why can't I edit? ⓘ". To the right of the profile card is a "Security info" section with a subtitle "Keep your verification methods and security info up to date.", a "SET UP SELF-SERVICE PASSWORD RESET >" link, and an "ADDITIONAL SECURITY VERIFICATION >" link. Below these sections is a "Devices" section with a laptop icon, a subtitle "Disable a lost device and review your connected devices.", and a "MANAGE DEVICES >" link. On the left side of the main content area is a sidebar with navigation links: "Overview" (selected), "Security info", "Recent activity", "Organizations", "Devices", and "Privacy".

Methods available in combined registration

Combined registration supports the authentication methods and actions in the following table.

[] Expand table

Method	Register	Change	Delete
Microsoft Authenticator	Yes (maximum of 5)	No	Yes
Other authenticator app	Yes (maximum of 5)	No	Yes
Hardware token	No	No	Yes
Phone	Yes (maximum of 2)	Yes	Yes
Alternate phone	Yes	Yes	Yes
Office phone*	Yes	Yes	Yes
Email	Yes	Yes	Yes

Method	Register	Change	Delete
Security questions	Yes	No	Yes
Passwords	No	Yes	No
App passwords*	Yes	No	Yes
Passkey (FIDO2)*	Yes (maximum of 10)	No	Yes

① Note

If you enable Microsoft Authenticator for passwordless authentication mode in the Authentication methods policy, users need to also enable passwordless sign-in in the Authenticator app.

Alternate phone can only be registered in *manage mode* on [Security info](#) and requires Voice calls to be enabled in the Authentication methods policy.

Office phone can only be registered in *Interrupt mode* if the users *Business phone* property is set. Office phone can be added by users in *Manage mode* from [Security info](#) without this requirement.

App passwords are available only to users who are enforced for per-user MFA. App passwords aren't available to users who are enabled for Microsoft Entra multifactor authentication by a Conditional Access policy.

Passkeys (FIDO2) can also be provisioned by using a custom client or partner integration with Microsoft Graph. For more information, see our [APIs](#).

Users can set one of the following options as the default multifactor authentication method.

- Microsoft Authenticator – push notification or passwordless
- Authenticator app or hardware token – code
- Phone call
- Text message

① Note

Virtual phone numbers aren't supported for Voice calls or SMS messages.

Third party authenticator apps don't provide push notification. As we continue to add more authentication methods to Microsoft Entra ID, those methods become available in

combined registration.

Combined registration modes

There are two modes of combined registration:

- **Interrupt mode** is a wizard-like experience, presented to users when they register or refresh their security info at sign-in.
- **Manage mode** is part of the user profile and allows users to manage their security info.

For both modes, users who have previously registered a method that can be used for Microsoft Entra multifactor authentication need to perform multifactor authentication before they can access their security info. Users must confirm their information before continuing to use their previously registered methods.

Interrupt mode

Combined registration adheres to both multifactor authentication and SSPR policies, if both are enabled for your tenant. These policies control whether a user is interrupted for registration during sign-in and which methods are available for registration. If only an SSPR policy is enabled, then users are be able to skip (indefinitely) the registration interruption and complete it at a later time.

The following are sample scenarios where users might be prompted to register or refresh their security info:

- *multifactor authentication registration enforced through Microsoft Entra ID Protection:* Users are asked to register during sign-in. They register multifactor authentication methods and SSPR methods (if the user is enabled for SSPR).
- *multifactor authentication registration enforced through per-user multifactor authentication:* Users are asked to register during sign-in. They register multifactor authentication methods and SSPR methods (if the user is enabled for SSPR).
- *multifactor authentication registration enforced through Conditional Access or other policies:* Users are asked to register when they use a resource that requires multifactor authentication. They register multifactor authentication methods and SSPR methods (if the user is enabled for SSPR).
- *SSPR registration enforced:* Users are asked to register during sign-in. They register only SSPR methods.
- *SSPR refresh enforced:* Users are required to review their security info at an interval set by the admin. Users are shown their info and can confirm the current info or make changes if needed.

When registration is enforced, users are shown the minimum number of methods needed to be compliant with both multifactor authentication and SSPR policies, from most to least secure. Users going through combined registration where both MFA and SSPR registration are enforced, and the SSPR policy requires two methods, are first required to register an MFA method as the first method and can select another MFA or SSPR specific method as the second registered method (such as email, security questions, and so on)

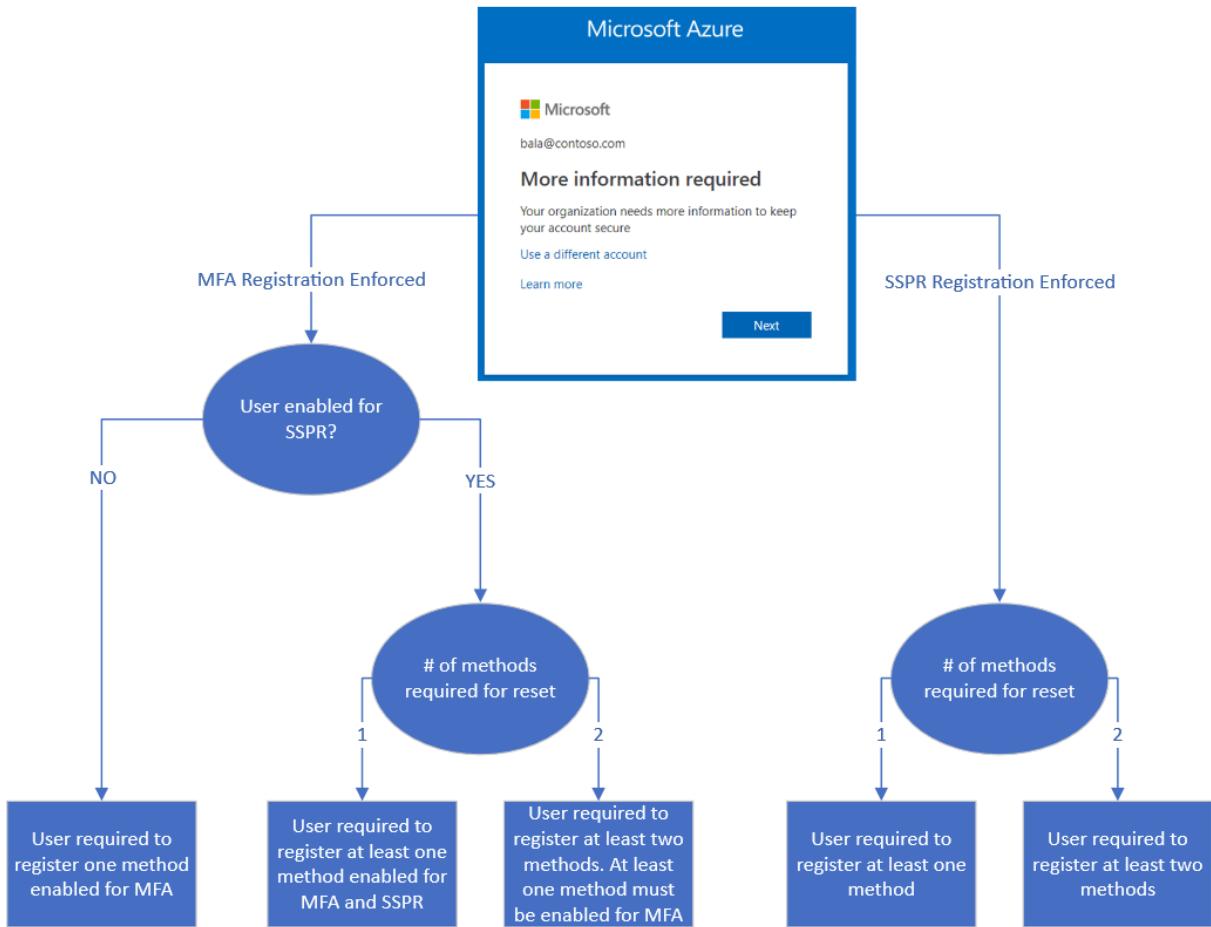
Consider the following example scenario:

- A user is enabled for SSPR. The SSPR policy requires two methods to reset and is enabled Microsoft Authenticator app, email, and phone.
- When the user chooses to register, two methods are required:
 - The user is shown Microsoft Authenticator app and phone by default.
 - The user can choose to register email instead of Authenticator app or phone.

When they set up Microsoft Authenticator, the user can select **I want to setup a different method** to register other authentication methods. The list of available methods is determined by the Authentication methods policy for the tenant.

The screenshot shows a setup wizard for Microsoft Authenticator. At the top, it says "Keep your account secure" and "Your organisation requires you to set up the following methods of proving who you are." Below this, it says "Microsoft Authenticator" and "Start by getting the app". It provides instructions to download the app and choose "Next". There is a link "I want to use a different authenticator app". At the bottom, there are two buttons: "Next" and "Skip setup". The link "I want to use a different authenticator app" is highlighted with a red border.

The following flowchart describes which methods are shown to a user when interrupted to register during sign-in:



If you have both multifactor authentication and SSPR enabled, we recommend that you enforce multifactor authentication registration.

If the SSPR policy requires users to review their security info at regular intervals, users are interrupted during sign-in and shown all their registered methods. They can confirm the current info if it's up to date, or they can make changes if they need to. Users must perform multifactor authentication to access this page.

Manage mode

Users can go to [Security info](#), or they can select **Security info** from My Account. From there, users can add methods, delete or change existing methods, change the default method, and more.

Session controls for Combined Registration

By default Combined registration enforces all MFA capable users to strongly authenticate prior to registering or managing their security info. If a user is currently signed in, and previously completed MFA as part of a valid session, no additional MFA is required by default, unless a user is attempting to add or modify a passkey (FIDO2) method. Adding or modifying a passkey (FIDO2) method requires users to have strongly

authenticated within the past 5 minutes. If MFA hasn't been completed in the past 5 minutes, the user is asked to sign-in and complete fresh MFA. Organizations can modify the authentication requirements by defining [Conditional Access policies for securing security info registration..](#)

Combined registration sessions are only valid for 15 minutes. If a user's registration or management actions take longer than this time period, the session expires and the user is asked to sign back in to continue.

Key usage scenarios

Change a password in MySignIns

A user navigates to [Security info](#). After signing in, the user can change their password. If the user authenticates with a password and a multifactor authentication method, they're able to use the enhanced user experience to change their password without entering their existing password. When finished, the user has the new password updated on the Security info page. Authentication methods such as Temporary Access Pass (TAP) aren't supported for password change unless the user knows their existing password.

 Note

If you have any links that point to the legacy change password experience, update them to the following forward link to direct users to the new **My Sign Ins Change Password** experience: <https://go.microsoft.com/fwlink/?linkid=2224198>.

Protect Security info registration with Conditional Access

To secure when and how users register for Microsoft Entra multifactor authentication and self-service password reset, you can use user actions in Conditional Access policy. This functionality may be enabled in organizations that want users to register for Microsoft Entra multifactor authentication and SSPR from a central location, such as a trusted network location during HR onboarding. Learn more on how to configure [common Conditional Access policies for securing security info registration.](#)

Set up security info during sign-in

An admin enforced registration.

A user hasn't set up all required security info and goes to the Microsoft Entra admin center. After the user enters the user name and password, the user is prompted to set up security info. The user then follows the steps shown in the wizard to set up the required security info. If your settings allow it, the user can choose to set up methods other than those shown by default. After users complete the wizard, they review the methods they set up and their default method for multifactor authentication. To complete the setup process, the user confirms the info and continues to the Microsoft Entra admin center.

Set up security info from My Account

An admin hasn't enforced registration.

A user who hasn't yet set up all required security info goes to <https://myaccount.microsoft.com>. The user selects **Security info** in the left pane. From there, the user chooses to add a method, selects any of the methods available, and follows the steps to set up that method. When finished, the user sees the method that was set up on the Security info page.

Set up other methods after partial registration

If a user partially satisfied MFA or SSPR registration due to existing authentication method registrations performed by the user or admin, users are only be asked to register additional information allowed by the Authentication methods policy settings when registration is required. If more than one other authentication method is available for the user to choose and register, an option on the registration experience titled **I want to set up another method** is shown and allows the user to set up their desired authentication method.

Keep your account secure

Your organization requires you to set up the following methods of proving who you are.

Method 1 of 2: App



App



Phone

Microsoft Authenticator



Start by getting the app

On your phone, install the Microsoft Authenticator app. [Download now](#)

After you install the Microsoft Authenticator app on your device, choose "Next".

[I want to use a different authenticator app](#)

[Next](#)

[Skip setup](#)

Delete security info from My Account

A user who has previously set up at least one method navigates to [Security info](#). The user chooses to delete one of the previously registered methods. When finished, the user no longer sees that method on the Security info page.

Change the default method from My Account

A user who has previously set up at least one method that can be used for multifactor authentication navigates to [Security info](#). The user changes the current default method to a different default method. When finished, the user sees the new default method on the Security info page.

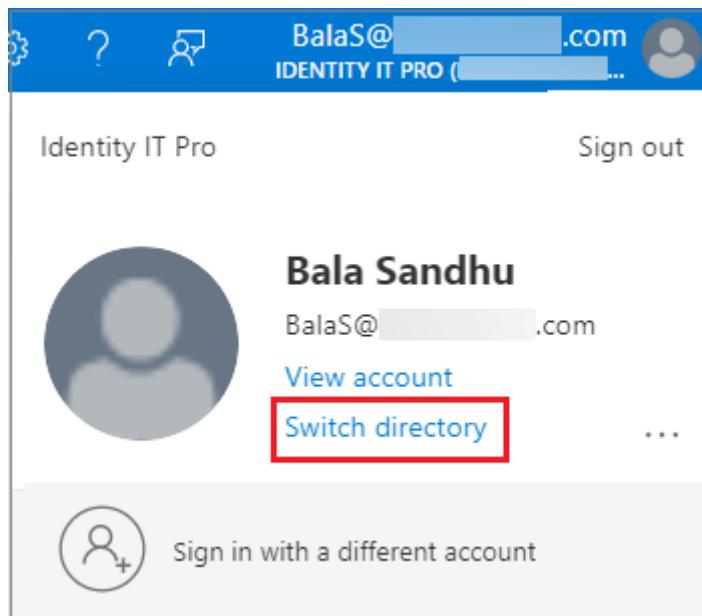
Switch directory

An external identity such as a B2B user may need to switch the directory to change the security registration information for a third-party tenant. In addition, users who access a resource tenant may be confused when they change settings in their home tenant but don't see the changes reflected in the resource tenant.

For example, a user sets Microsoft Authenticator app push notification as the primary authentication to sign-in to home tenant and also has SMS/Text as another option. This user is also configured with SMS/Text option on a resource tenant. If this user removes

SMS/Text as one of the authentication options on their home tenant, they get confused when access to the resource tenant asks them to respond to SMS/Text message.

To switch the directory in the Microsoft Entra admin center, select the user account name in the upper right corner and select **Switch directory**.



Or, you can specify a tenant by URL to access security information.

`https://mysignins.microsoft.com/security-info?tenant=<Tenant Name>`

`https://mysignins.microsoft.com/security-info/?tenantId=<Tenant ID>`

ⓘ Note

Customers attempting to register or manage security info through combined registration or the My Sign-ins page should use a modern browser such as Microsoft Edge.

IE11 isn't officially supported for creating a webview or browser in applications as it doesn't work as expected in all scenarios.

Applications that aren't updated and are still using Azure AD Authentication Library (ADAL) that rely on legacy webviews can fallback to older versions of Internet Explorer. In these scenarios, users experience a blank page when directed to the My Sign-ins page. To resolve this issue, switch to a modern browser.

Next steps

To get started, see the tutorials to [enable self-service password reset](#) and [enable Microsoft Entra multifactor authentication](#).

Learn how to [enable combined registration in your tenant](#) or [force users to re-register authentication methods](#).

You can also review the [available methods for Microsoft Entra multifactor authentication](#) and [SSPR](#).

Feedback

Was this page helpful?



Yes



No

[Provide product feedback](#) ↗

Create a resilient access control management strategy with Microsoft Entra ID

Article • 03/04/2025

ⓘ Note

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft can't guarantee the accuracy of any information presented after the date of publication.

Organizations that rely on a single access control, such as multifactor authentication or a single network location, to secure their IT systems are susceptible to access failures to their apps and resources if that single access control becomes unavailable or misconfigured. For example, a natural disaster can result in the unavailability of large segments of telecommunications infrastructure or corporate networks. Such a disruption could prevent end users and administrators from being able to sign in.

This document provides guidance on strategies an organization should adopt to provide resilience to reduce the risk of lockout during unforeseen disruptions with the following scenarios:

- Organizations can increase their resiliency to reduce the risk of lockout **before a disruption** by implementing mitigation strategies or contingency plans.
- Organizations can continue to access apps and resources they choose **during a disruption** by having mitigation strategies and contingency plans in place.
- Organizations should make sure they preserve information, such as logs, **after a disruption** and before they roll back any contingencies they implemented.
- Organizations that haven't implemented prevention strategies or alternative plans may be able to implement **emergency options** to deal with the disruption.

Key guidance

There are four key takeaways in this document:

- Avoid administrator lockout by using emergency access accounts.

- Implement MFA using Conditional Access rather than per-user MFA.
- Mitigate user lockout by using multiple Conditional Access controls.
- Mitigate user lockout by provisioning multiple authentication methods or equivalents for each user.

Before a disruption

Mitigating an actual disruption must be an organization's primary focus in dealing with access control issues that may arise. Mitigating includes planning for an actual event plus implementing strategies to make sure access controls and operations are unaffected during disruptions.

Why do you need resilient access control?

Identity is the control plane of users accessing apps and resources. Your identity system controls which users and under which conditions, such as access controls or authentication requirements, users get access to the applications. When one or more authentication or access control requirements aren't available for users to authenticate due to unforeseen circumstances, organizations can experience one or both of the following issues:

- **Administrator lockout:** Administrators can't manage the tenant or services.
- **User lockout:** Users can't access apps or resources.

Administrator lockout contingency

Microsoft recommends that organizations have two cloud-only emergency access accounts permanently assigned the [Global Administrator](#) role. These accounts are highly privileged and aren't assigned to specific individuals. The accounts are limited to emergency or "break glass" scenarios where normal accounts can't be used or all other administrators are accidentally locked out. These accounts should be created following the [emergency access account recommendations](#).

Mitigating user lockout

To mitigate the risk of user lockout, use Conditional Access policies with multiple controls to give users a choice of how they access apps and resources. By giving a user the choice between, for example, signing in with MFA **or** signing in from a managed device **or** signing in from the corporate network, if one of the access controls is unavailable the user has other options to continue to work.

Microsoft recommendations

Incorporate the following access controls in your existing Conditional Access policies for organization:

- Provision multiple authentication methods for each user that rely on different communication channels, for example, the Microsoft Authenticator app (internet-based), OATH token (generated on-device), and SMS (telephonic).
- Deploy Windows Hello for Business on Windows 10 devices to satisfy MFA requirements directly from device sign-in.
- Use trusted devices via [Microsoft Entra hybrid join](#) or [Microsoft Intune](#). Trusted devices improve user experience because the trusted device itself can satisfy the strong authentication requirements of policy without an MFA challenge to the user. MFA will then be required when enrolling a new device and when accessing apps or resources from untrusted devices.
- Use Microsoft Entra ID Protection risk-based policies that prevent access when the user or sign-in is at risk in place of fixed MFA policies.
- If you're protecting VPN access using Microsoft Entra multifactor authentication NPS extension, consider federating your VPN solution as a [SAML app](#) and determine the app category as recommended below.

 **Note**

Risk-based policies require [Microsoft Entra ID P2](#) licenses.

The following example describes policies you must create to provide a resilient access control for user to access their apps and resources. In this example, you require a security group **AppUsers** with the target users you want to give access to, a group named **CoreAdmins** with the core administrators, and a group named **EmergencyAccess** with the emergency access accounts. This example policy set will grant selected users in **AppUsers**, access to selected apps if they're connecting from a trusted device OR provide strong authentication, for example MFA. It excludes emergency accounts and core administrators.

Conditional Access mitigation policies set:

- Policy 1: Block access to people outside target groups
 - Users and Groups: Include all users. Exclude AppUsers, CoreAdmins, and EmergencyAccess
 - Cloud Apps: Include all apps
 - Conditions: (None)
 - Grant Control: Block

- Policy 2: Grant access to AppUsers requiring MFA OR trusted device.
 - Users and Groups: Include AppUsers. Exclude CoreAdmins, and EmergencyAccess
 - Cloud Apps: Include all apps
 - Conditions: (None)
 - Grant Control: Grant access, require multifactor authentication, require device to be compliant. For multiple controls: Require one of the selected controls.

Contingencies for user lockout

Alternatively, your organization can also create contingency policies. To create contingency policies, you must define tradeoff criteria between business continuity, operational cost, financial cost, and security risks. For example, you may activate a contingency policy only to a subset of users, for a subset of apps, for a subset of clients, or from a subset of locations. Contingency policies give administrators and end users access to apps and resources, during a disruption when no mitigation method was implemented. Microsoft recommends enabling contingency policies in [report-only mode](#) when not in use so that administrators can monitor the potential impact of the policies should they need to be turned on.

Understanding your exposure during a disruption helps reduce your risk and is a critical part of your planning process. To create your contingency plan, first determine the following business requirements of your organization:

1. Determine your mission critical apps ahead of time: What are the apps that you must give access to, even with a lower risk/security posture? Build a list of these apps and make sure your other stakeholders (business, security, legal, leadership) all agree that if all access control goes away, these apps still must continue to run. You're likely going to end up with categories of:
 - **Category 1 mission critical apps** that can't be unavailable for more than a few minutes, for example Apps that directly affect the revenue of the organization.
 - **Category 2 important apps** that the business needs to be accessible within a few hours.
 - **Category 3 low-priority apps** that can withstand a disruption of a few days.
2. For apps in category 1 and 2, Microsoft recommends you preplan what type of level of access you want to allow:
 - Do you want to allow full access or restricted session, like limiting downloads?

- Do you want to allow access to part of the app but not the whole app?
- Do you want to allow information worker access and block administrator access until the access control is restored?

3. For those apps, Microsoft also recommends you plan which avenues of access you'll deliberately open and which ones you'll close:

- Do you want to allow browser only access and block rich clients that can save offline data?
- Do you want to allow access only for users inside the corporate network and keep outside users blocked?
- Do you want to allow access from certain countries or regions only during the disruption?
- Do you want policies to the contingency policies, especially for mission critical apps, to fail or succeed if an alternative access control isn't available?

Microsoft recommendations

A contingency Conditional Access policy is a **backup policy** that omits Microsoft Entra multifactor authentication, third-party MFA, risk-based or device-based controls. In order to minimize unexpected disruption when a contingency policy is enabled, the policy should remain in report-only mode when not in use. Administrators can monitor the potential impact of their contingency policies using the Conditional Access Insights workbook. When your organization decides to activate your contingency plan, administrators can enable the policy and disable the regular control-based policies.

Important

Disabling policies that enforce security on your users, even temporarily, will reduce your security posture while the contingency plan is in place.

- Configure a set of fallback policies if a disruption in one credential type or one access control mechanism impacts access to your apps. Configure a policy in report-only state that requires Domain Join as a control, as a backup for an active policy that requires a third-party MFA provider.
- Reduce the risk of bad actors guessing passwords, when MFA isn't required, by following the practices in the [password guidance](#) white paper.
- Deploy [Microsoft Entra Self-Service Password Reset \(SSPR\)](#) and [Microsoft Entra Password Protection](#) to make sure users don't use common password and terms you choose to ban.
- Use policies that restrict the access within the apps if a certain authentication level isn't attained instead of simply falling back to full access. For example:

- Configure a backup policy that sends the restricted session claim to Exchange and SharePoint.
- If your organization uses Microsoft Defender for Cloud Apps, consider falling back to a policy that engages Defender for Cloud Apps and then allow read-only access but not uploads.
- Name your policies to make sure it's easy to find them during a disruption. Include the following elements in the policy name:
 - A *label number* for the policy.
 - Text to show, this policy is for emergencies only. For example: **ENABLE IN EMERGENCY**
 - The *disruption* it applies to. For example: **During MFA Disruption**
 - A *sequence number* to show the order you must activate the policies.
 - The *apps* it applies to.
 - The *controls* it will apply.
 - The *conditions* it requires.

This naming standard for the contingency policies are as follows:

EMnnn - ENABLE IN EMERGENCY: [Disruption][i/n] - [Apps] - [Controls]
[Conditions]

The following example: **Example A - Contingency Conditional Access policy to restore Access to mission-critical Collaboration Apps**, is a typical corporate contingency. In this scenario, the organization typically requires MFA for all Exchange Online and SharePoint Online access, and the disruption in this case is the MFA provider for the customer has an outage (whether Microsoft Entra multifactor authentication, on-premises MFA provider, or third-party MFA). This policy mitigates this outage by allowing specific targeted users access to these apps from trusted Windows devices only when they're accessing the app from their trusted corporate network. It will also exclude emergency accounts and core administrators from these restrictions. The targeted users will then gain access to Exchange Online and SharePoint Online, while other users will still not have access to the apps due to the outage. This example requires a named network location **CorpNetwork** and a security group **ContingencyAccess** with the target users, a group named **CoreAdmins** with the core administrators, and a group named **EmergencyAccess** with the emergency access accounts. The contingency requires four policies to provide the desired access.

Example A - Contingency Conditional Access policies to restore Access to mission-critical Collaboration Apps:

- Policy 1: Require Domain Joined devices for Exchange and SharePoint
 - Name: EM001 - ENABLE IN EMERGENCY: MFA Disruption[1/4] - Exchange SharePoint - Require Microsoft Entra hybrid join
 - Users and Groups: Include ContingencyAccess. Exclude CoreAdmins, and EmergencyAccess
 - Cloud Apps: Exchange Online and SharePoint Online
 - Conditions: Any
 - Grant Control: Require Domain Joined
 - State: Report-only
- Policy 2: Block platforms other than Windows
 - Name: EM002 - ENABLE IN EMERGENCY: MFA Disruption[2/4] - Exchange SharePoint - Block access except Windows
 - Users and Groups: Include all users. Exclude CoreAdmins, and EmergencyAccess
 - Cloud Apps: Exchange Online and SharePoint Online
 - Conditions: Device Platform Include All Platforms, exclude Windows
 - Grant Control: Block
 - State: Report-only
- Policy 3: Block networks other than CorpNetwork
 - Name: EM003 - ENABLE IN EMERGENCY: MFA Disruption[3/4] - Exchange SharePoint - Block access except Corporate Network
 - Users and Groups: Include all users. Exclude CoreAdmins, and EmergencyAccess
 - Cloud Apps: Exchange Online and SharePoint Online
 - Conditions: Locations Include any location, exclude CorpNetwork
 - Grant Control: Block
 - State: Report-only
- Policy 4: Block EAS Explicitly
 - Name: EM004 - ENABLE IN EMERGENCY: MFA Disruption[4/4] - Exchange - Block EAS for all users
 - Users and Groups: Include all users
 - Cloud Apps: Include Exchange Online
 - Conditions: Client apps: Exchange Active Sync
 - Grant Control: Block
 - State: Report-only

Order of activation:

1. Exclude ContingencyAccess, CoreAdmins, and EmergencyAccess from the existing MFA policy. Verify a user in ContingencyAccess can access SharePoint Online and Exchange Online.
2. Enable Policy 1: Verify users on Domain Joined devices who aren't in the exclude groups are able to access Exchange Online and SharePoint Online. Verify users in

the Exclude group can access SharePoint Online and Exchange from any device.

3. Enable Policy 2: Verify users who aren't in the exclude group can't get to SharePoint Online and Exchange Online from their mobile devices. Verify users in the Exclude group can access SharePoint and Exchange from any device (Windows/iOS/Android).
4. Enable Policy 3: Verify users who aren't in the exclude groups can't access SharePoint and Exchange off the corporate network, even with a domain joined machine. Verify users in the Exclude group can access SharePoint and Exchange from any network.
5. Enable Policy 4: Verify all users can't get Exchange Online from the native mail applications on mobile devices.
6. Disable the existing MFA policy for SharePoint Online and Exchange Online.

In this next example, **Example B - Contingency Conditional Access policies to allow mobile access to Salesforce**, a business app's access is restored. In this scenario, the customer typically requires their sales employees access to Salesforce (configured for single-sign on with Microsoft Entra ID) from mobile devices to only be allowed from compliant devices. The disruption in this case is that there's an issue with evaluating device compliance and the outage is happening at a sensitive time where the sales team needs access to Salesforce to close deals. These contingency policies grants critical users access to Salesforce from a mobile device so that they can continue to close deals and not disrupt the business. In this example, **SalesforceContingency** contains all the Sales employees who need to retain access and **SalesAdmins** contains necessary admins of Salesforce.

Example B - Contingency Conditional Access policies:

- Policy 1: Block everyone not in the SalesContingency team
 - Name: EM001 - ENABLE IN EMERGENCY: Device Compliance Disruption[1/2] - Salesforce - Block All users except SalesforceContingency
 - Users and Groups: Include all users. Exclude SalesAdmins and SalesforceContingency
 - Cloud Apps: Salesforce.
 - Conditions: None
 - Grant Control: Block
 - State: Report-only
- Policy 2: Block the Sales team from any platform other than mobile (to reduce surface area of attack)
 - Name: EM002 - ENABLE IN EMERGENCY: Device Compliance Disruption[2/2] - Salesforce - Block All platforms except iOS and Android
 - Users and Groups: Include SalesforceContingency. Exclude SalesAdmins
 - Cloud Apps: Salesforce

- Conditions: Device Platform Include All Platforms, exclude iOS and Android
- Grant Control: Block
- State: Report-only

Order of activation:

1. Exclude SalesAdmins and SalesforceContingency from the existing device compliance policy for Salesforce. Verify a user in the SalesforceContingency group can access Salesforce.
2. Enable Policy 1: Verify users outside of SalesContingency can't access Salesforce. Verify users in the SalesAdmins and SalesforceContingency can access Salesforce.
3. Enable Policy 2: Verify users in the SalesContingency group can't access Salesforce from their Windows/Mac laptops but can still access from their mobile devices. Verify SalesAdmin can still access Salesforce from any device.
4. Disable the existing device compliance policy for Salesforce.

Contingencies for user lockout from on-premises resources (NPS extension)

If you're protecting VPN access using Microsoft Entra multifactor authentication NPS extension, consider federating your VPN solution as a [SAML app](#) and determine the app category as recommended below.

If you have deployed Microsoft Entra multifactor authentication NPS extension to protect on-premises resources, such as VPN and Remote Desktop Gateway, with MFA - you should consider in advance if you're ready to disable MFA in a case of emergency.

In this case, you can disable the NPS extension, as a result, the NPS server will only verify primary authentication and won't enforce MFA on the users.

Disable NPS extension:

- Export the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AuthSrv\Parameters registry key as a backup.
- Delete the registry values for "AuthorizationDLLs" and "ExtensionDLLs", not the Parameters key.
- Restart the Network Policy Service (IAS) service for the changes to take effect
- Determine if primary authentication for VPN is successful.

Once the service has recovered and you're ready to enforce MFA on your users again, enable the NPS extension:

- Import the registry key from backup
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AuthSrv\Parameters
- Restart the Network Policy Service (IAS) service for the changes to take effect
- Determine if primary authentication and secondary authentication for VPN is successful.
- Review NPS server and the VPN log to determine which users have signed in during the emergency window.

Deploy password hash sync even if you're federated or use pass-through authentication

User lockout can also occur if the following conditions are true:

- Your organization uses a hybrid identity solution with pass-through authentication or federation.
- Your on-premises identity systems (such as Active Directory, AD FS, or a dependent component) are unavailable.

To be more resilient, your organization should [enable password hash sync](#), because it enables you to [switch to using password hash sync](#) if your on-premises identity systems are down.

Microsoft recommendations

Enable password hash sync using the Microsoft Entra Connect wizard, regardless whether your organization uses federation or pass-through authentication.

Important

It isn't required to convert users from federated to managed authentication to use password hash sync.

During a disruption

If you opted for implementing a mitigation plan, you're able to automatically survive a single access control disruption. However, if you opted to create a contingency plan, you're able to activate your contingency policies during the access control disruption:

1. Enable your contingency policies that grant targeted users, access to specific apps, from specific networks.

2. Disable your regular control-based policies.

Microsoft recommendations

Depending on which mitigations or contingencies are used during a disruption, your organization could be granting access with just passwords. No safeguard is a considerable security risk that must be weighed carefully. Organizations must:

1. As part of your change control strategy, document every change and the previous state to be able to roll back any contingencies you implemented as soon as the access controls are fully operational.
2. Assume that malicious actors will attempt to harvest passwords through password spray or phishing attacks while you disabled MFA. Also, bad actors might already have passwords that previously didn't grant access to any resource that can be attempted during this window. For critical users such as executives, you can partially mitigate this risk by resetting their passwords before disabling MFA for them.
3. Archive all sign-in activity to identify who accessed what during the time MFA was disabled.
4. [Triage all risk detections reported](#) during this window.

After a disruption

Undo the changes you made as part of the activated contingency plan once the service is restored that caused the disruption.

1. Enable the regular policies
2. Disable your contingency policies back to report-only mode.
3. Roll back any other changes you made and documented during the disruption.
4. If you used an emergency access account, remember to regenerate credentials and physically secure the new credentials details as part of your emergency access account procedures.
5. Continue to [Triage all risk detections reported](#) after the disruption for suspicious activity.
6. [Revoke all refresh tokens](#) that were issued to target a set of users. Revoking all refresh tokens is important for privileged accounts used during the disruption and doing it will force them to reauthenticate and meet the control of the restored policies.

Emergency options

In an emergency and your organization didn't previously implement a mitigation or contingency plan, then follow the recommendations in the [Contingencies for user lockout](#) section if they already use Conditional Access policies to enforce MFA. If your organization is using per-user MFA legacy policies, then you can consider the following alternative:

- If you have the corporate network outbound IP address, you can add them as trusted IPs to enable authentication only to the corporate network.
- If you don't have the inventory of outbound IP addresses, or you required to enable access inside and outside the corporate network, you can add the entire IPv4 address space as trusted IPs by specifying 0.0.0.0/1 and 128.0.0.0/1.

 **Important**

If you broaden the trusted IP addresses to unblock access, risk detections associated with IP addresses (for example, impossible travel or unfamiliar locations) won't be generated.

 **Note**

Configuring [trusted IPs](#) for Microsoft Entra multifactor authentication is only available with [Microsoft Entra ID P1 or P2 licenses](#).

Learn more

- [Microsoft Entra authentication Documentation](#)
- [Manage emergency-access administrative accounts in Microsoft Entra ID](#)
- [Configure named locations in Microsoft Entra ID](#)
- [How to configure Microsoft Entra hybrid joined devices](#)
- [Windows Hello for Business Deployment Guide](#)
 - [Password Guidance - Microsoft Research ↗](#)
- [What are conditions in Microsoft Entra Conditional Access?](#)
- [What are access controls in Microsoft Entra Conditional Access?](#)
- [What is Conditional Access report-only mode?](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Web browser cookies used in Microsoft Entra authentication

Article • 03/04/2025

During authentication against Microsoft Entra ID through a web browser, multiple cookies are involved in the process. Some of the cookies are common on all requests. Other cookies are used for specific authentication flows or specific client-side conditions.

Persistent session tokens are stored as persistent cookies on the web browser's cookie jar. Non-persistent session tokens are stored as session cookies on the web browser, and are destroyed when the browser session is closed.

[] [Expand table](#)

Cookie Name	Type	Comments
ESTSAUTH	Common	Contains user's session information to facilitate SSO. Transient.
ESTSAUTHPERSISTENT	Common	Contains user's session information to facilitate SSO. Persistent.
ESTSAUTHLIGHT	Common	Contains Session GUID Information. Lite session state cookie used exclusively by client-side JavaScript in order to facilitate OIDC sign-out. Security feature.
SignInStateCookie	Common	Contains list of services accessed to facilitate sign-out. No user information. Security feature.
CCState	Common	Contains session information state to be used between Microsoft Entra ID and the Microsoft Entra Backup Authentication Service .
buid	Common	Tracks browser related information. Used for service telemetry and protection mechanisms.
fpc	Common	Tracks browser related information. Used for tracking requests and throttling.
esctx	Common	Session context cookie information. For CSRF protection. Binds a request to a specific browser instance so the request can't be replayed outside the browser. No user information.
ch	Common	ProofOfPossessionCookie. Stores the Proof of Possession

Cookie Name	Type	Comments
		cookie hash to the user agent.
ESTSSC	Common	Legacy cookie containing session count information no longer used.
ESTSSSOTILES	Common	Tracks session sign-out. When present and not expired, with value "ESTSSSOTILES=1", it interrupts SSO, for specific SSO authentication model, and presents tiles for user account selection.
AADSSOTILES	Common	Tracks session sign-out. Similar to ESTSSSOTILES but for other specific SSO authentication model.
ESTSUSERLIST	Common	Tracks Browser SSO user's list.
SSOCOOKIEPULLED	Common	Prevents looping on specific scenarios. No user information.
cltm	Common	For telemetry purposes. Tracks AppVersion, ClientFlight, and Network type.
brcap	Common	Client-side cookie (set by JavaScript) to validate client/web browser's touch capabilities.
clrc	Common	Client-side cookie (set by JavaScript) to control local cached sessions on the client.
CkTst	Common	Client-side cookie (set by JavaScript). No longer in active use.
wlidperf	Common	Client-side cookie (set by JavaScript) that tracks local time for performance purposes.
x-ms-gateway-slice	Common	Microsoft Entra Gateway cookie used for tracking and load balance purposes.
stsservicecookie	Common	Microsoft Entra Gateway cookie also used for tracking purposes.
x-ms-refreshtokencredential	Specific	Available when Primary Refresh Token (PRT) is in use.
estsStateTransient	Specific	Applicable to new session information model only. Transient.
estsStatePersistent	Specific	Same as estsStateTransient, but persistent.
ESTSNLOGIN	Specific	National Cloud Login related Cookie.
UsGovTraffic	Specific	US Gov Cloud Traffic Cookie.

Cookie Name	Type	Comments
ESTSWCTXFLOWTOKEN	Specific	Saves flowToken information when redirecting to ADFS.
CcsNtv	Specific	To control when Microsoft Entra Gateway sends requests to Microsoft Entra Backup Authentication Service . Native flows.
CcsWeb	Specific	To control when Microsoft Entra Gateway sends requests to Microsoft Entra Backup Authentication Service . Web flows.
Ccs*	Specific	Cookies with prefix Ccs*, have the same purpose as the ones without prefix, but only apply when Microsoft Entra Backup Authentication Service is in use.
threxp	Specific	Used for throttling control.
rrc	Specific	Cookie used to identify a recent B2B invitation redemption.
debug	Specific	Cookie used to track if user's browser session is enabled for DebugMode.
MSFPC	Specific	This cookie is not specific to any ESTS flow, but is sometimes present. It applies to all Microsoft Sites (when accepted by users). Identifies unique web browsers visiting Microsoft sites. It's used for advertising, site analytics, and other operational purposes.

 **Note**

Cookies identified as client-side cookies are set locally on the client device by JavaScript, hence, will be marked with `HttpOnly=false`.

Cookie definitions and respective names are subject to change at any moment in time according to Microsoft Entra service requirements.

Next steps

To learn more about self-service password reset concepts, see [How Microsoft Entra self-service password reset works](#).

To learn more about multifactor authentication concepts, see [How Microsoft Entra multifactor authentication works](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

How to migrate MFA and SSPR policy settings to the Authentication methods policy for Microsoft Entra ID

Article • 03/04/2025

You can migrate Microsoft Entra ID [legacy policy settings](#) that separately control multifactor authentication (MFA) and self-service password reset (SSPR) to unified management with the [Authentication methods policy](#).

You can use the authentication methods migration guide in the Microsoft Entra admin center to automate the migration. The guide provides a wizard to help audit your current policy settings for MFA and SSPR. Then it consolidates those settings in the Authentication methods policy, where they can be managed together more easily.

You can also migrate policy settings manually on your own schedule. The migration process is fully reversible. You can continue to use tenant-wide MFA and SSPR policies while you configure authentication methods more precisely for users and groups in the Authentication methods policy.

For more information about how these policies work together during migration, see [Manage authentication methods for Microsoft Entra ID](#).

Automated migration guide

The automated migration guide lets you migrate where you manage authentication methods in just a few clicks. It can be accessed from the [Microsoft Entra admin center](#) by browsing to **Protection > Authentication methods > Policies**.

Microsoft Azure

Search resources, services, and docs (G+/)

Copilot

Home > Authentication methods | Policies

Microsoft Entra ID Security

Search Add external method (Preview) Refresh Got feedback?

Manage

- Policies (selected)
- Password protection
- Registration campaign
- Authentication strengths
- Settings

Monitoring

- Activity
- User registration details
- Registration and reset events
- Bulk operation results

Manage migration

On September 30th, 2025, the legacy multifactor authentication (MFA) and self-service password reset (SSPR) policies will be deprecated and the settings will be managed here. Use the options below to manage your migration status – how your policies are respected – and utilize the migration wizard to quickly migrate legacy policies to the new unified policies. [Learn more](#)

Migration status: In progress (change) | Begin automated guide

Authentication method policies

Use authentication methods policies to configure the authentication methods your users may register and use. If a user is in scope for a method, they may use it to authenticate and for password reset (some methods aren't supported for some scenarios). [Learn more](#)

Method	Target	Enabled
Passkey (FIDO2)	All users	Yes
Microsoft Authenticator	All users, excluding 1 group	Yes
SMS	All users	Yes
Temporary Access Pass	All users	Yes
Hardware OATH tokens (Preview)	All users	Yes
Third-party software OATH tokens	All users	Yes
Voice call		No
Email OTP	All users	Yes
Certificate-based authentication		No

The first page of the wizard explains what it is and how it works. It also provides links to each of the legacy policies for your reference.

Authentication method settings migration

[Overview](#)[Review + migrate](#)

...

Migrating to the new authentication method policies

The authentication methods policy is the recommended way to manage authentication methods, including modern methods like passwordless authentication. This guide will help you automatically convert the authentication methods enabled in your legacy multifactor authentication (MFA) and legacy self-service password reset (SSPR) policies to corresponding authentication method policies. We will offer recommendations to help you increase your security posture.

[Why is this important?](#)

Current authentication method settings

[Legacy MFA method settings](#)

[Legacy SSPR method settings](#)

[Migrating to the new authentication method policies](#)

[Next](#)

The wizard then configures the Authentication method policy based on what your organization currently has enabled in the legacy MFA and SSPR policies. If a method is enabled in either legacy policy, the recommendation is to also enable it in the Authentication method policy. With that configuration, users can continue to sign in and reset their password by using the same method they used previously.

In addition, we recommend you enable the latest modern, secure methods like passkeys, Temporary Access Pass, and Microsoft Authenticator to help improve your organization's security posture. To edit the recommended configuration, select the pencil icon next to each method.

Authentication method settings migration

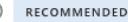
 Overview  Review + migrate

Authentication method policies

If a method is enabled in either legacy MFA or SSPR policy, it'll be enabled in the authentication methods policy. If a user is in scope for a method, they may use it to authenticate and for password reset (some methods aren't supported for some scenarios).

[Learn more](#)

 As part of migrating your authentication methods, we recommend enabling passkeys and temporary access pass for users to leverage the most seamless and secure methods for authentication and recovery. See [passkey-settings](#) for the latest on which passkeys are enabled by default

Method	Status	Targeted users & groups	Security	
Passkey (FIDO2) 	On	All users	 High security	
Temporary access pass 	On	All users	 High security	
Microsoft Authenticator (push & passwordless) 	On	All users	 High security	
Hardware OATH tokens	On	All users	 Medium security	
Third-party software OATH tokens	On	All users	 Medium security	
Email OTP	On	All users	 Low security	
SMS	On	All users	 Low security	
Voice call	Off		 Low security	

 Voice call is a method previously configured but now unavailable due to licensing. Please upgrade to regain access to this feature. [Learn more](#)

[Previous](#)

[Migrate](#)

Once you're happy with the configuration, select **Migrate**, and then confirm the migration. The Authentication methods policy gets updated to match the configuration specified in the wizard. Authentication methods in the legacy MFA and SSPR policies become grayed out and no longer apply.

Your migration status is updated to **Migration Complete**. You can change this status back to **In Progress** anytime to re-enable methods in the legacy policies if needed.

Manual migration

Begin by doing an audit of your existing policy settings for each authentication method that's available for users. If you roll back during migration, you might want a record of the authentication method settings from each of these policies:

- MFA policy
- SSPR policy (if used)
- Authentication methods policy (if used)

If you aren't using SSPR and aren't yet using the Authentication methods policy, you only need to get settings from the MFA policy.

Review the legacy MFA policy

Start by documenting which methods are available in the legacy MFA policy.

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Identity > Users > All users > Per-user MFA > service settings** to view the settings.

These settings are tenant-wide, so there's no need for user or group information.

The screenshot shows the 'multi-factor authentication' service settings page. It includes sections for 'app passwords', 'trusted ips', and 'verification options'. A red box highlights the 'Methods available to users' section under 'verification options', which lists four options: 'Call to phone', 'Text message to phone', 'Notification through mobile app', and 'Verification code from mobile app or hardware token'. The 'Text message to phone' option is checked. Below this section is a link to 'remember multi-factor authentication on trusted device' and a checkbox for 'Allow users to remember multi-factor authentication on devices they trust (between one to 365 days)'.

For each method, note whether or not it's enabled for the tenant. The following table lists methods available in the legacy MFA policy and corresponding methods in the Authentication method policy.

[\[+\] Expand table](#)

Multifactor authentication policy	Authentication method policy
Call to phone	Voice calls
Text message to phone	SMS

Multifactor authentication policy	Authentication method policy
Notification through mobile app	Microsoft Authenticator
Verification code from mobile app or hardware token	Third party software OATH tokens Hardware OATH tokens Microsoft Authenticator

Review the legacy SSPR policy

To get the authentication methods available in the legacy SSPR policy, go to **Identity > Users > All users > Password reset > Authentication methods**. The following table lists the available methods in the legacy SSPR policy and corresponding methods in the Authentication method policy.

Home > Conditional Access | Policies > Users | Password reset > Password reset

Password reset | Authentication methods ...

Contoso - Microsoft Entra ID for workforce

« Save Discard

Diagnose and solve problems

Manage

- Properties**
- Authentication methods**
- Registration
- Notifications
- Customization
- On-premises integration
- Administrator Policy

Activity

- Audit logs
- Usage & insights

Troubleshooting + Support

- New support request

Number of methods required to reset ⓘ 1 2

Methods available to users

- Mobile app notification
- Mobile app code
- Email
- Mobile phone (SMS only)
- Office phone ⓘ
- Security questions

ⓘ Authentication Methods for SSPR and Signin can now be managed in one converged policy. [Learn more](#)

ⓘ These settings only apply to end users in your organization. Admins are always enabled for self-service password reset and are required to use two authentication methods to reset their password. Click here to learn more about administrator password policies.

Record which users are in scope for SSPR (either all users, one specific group, or no users) and the authentication methods they can use. While security questions aren't yet available to manage in the Authentication methods policy, make sure you record them for later when they are. You can find this information by going to **Identity > Users > All users > Password reset > Properties**.

[] [Expand table](#)

SSPR authentication methods	Authentication method policy
Mobile app notification	Microsoft Authenticator
Mobile app code	Microsoft Authenticator Software OATH tokens
Email	Email OTP
Mobile phone	Voice calls SMS
Office phone	Voice calls
Security questions	Not yet available; copy questions for later use

Authentication methods policy

To check settings in the Authentication methods policy, sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#) and browse to **Protection > Authentication methods > Policies**. A new tenant has all methods **Off** by default, which makes migration easier because legacy policy settings don't need to be merged with existing settings.

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods >**

Method	Target	Enabled
FIDO2 security key	All users	Yes
Microsoft Authenticator	All users	Yes
SMS		No
Temporary Access Pass	All users	Yes
Hardware OATH tokens (Preview)		No
Third-party software OATH tokens	All users	Yes
Voice call	All users	Yes
Email OTP		Yes
Certificate-based authentication		No

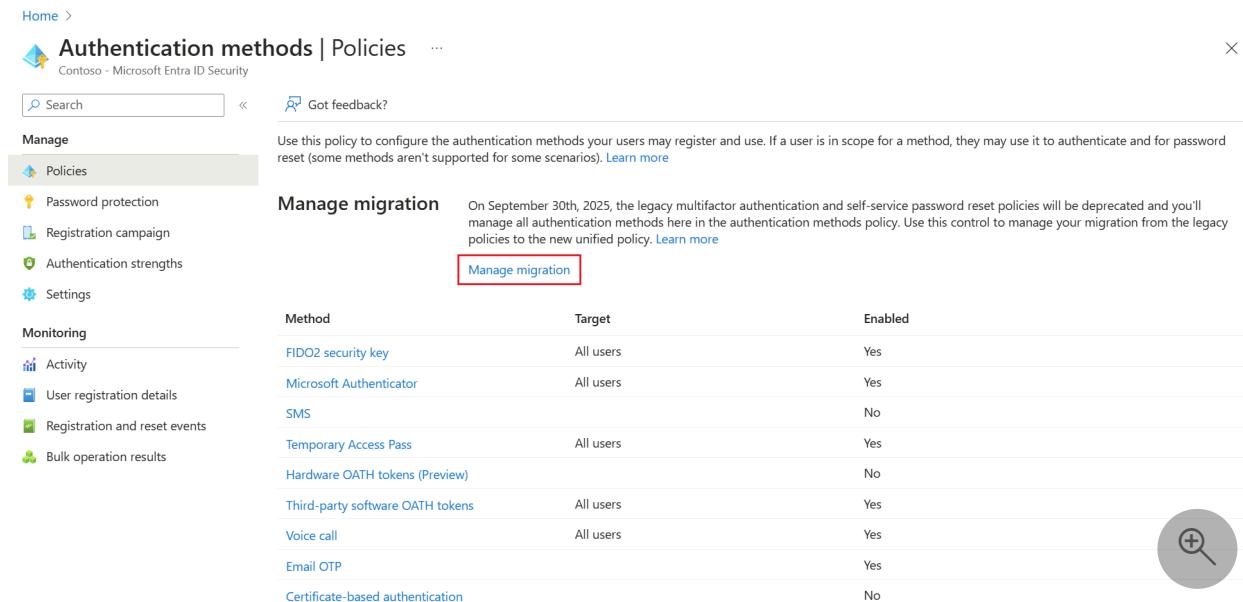
The Authentication methods policy has other methods that aren't available in the legacy policies, such as FIDO2 security key, Temporary Access Pass, and Microsoft Entra

certificate-based authentication. These methods aren't in scope for migration and you won't need to make any changes to them if you've configured them already.

If you've enabled other methods in the Authentication methods policy, write down the users and groups who can or can't use those methods. Take a note of the configuration parameters that govern how the method can be used. For example, you can configure Microsoft Authenticator to provide location in push notifications. Make a record of which users and groups are enabled for similar configuration parameters associated with each method.

Start the migration

After you capture available authentication methods from the policies you're currently using, you can start the migration. Open the Authentication methods policy, select **Manage migration**, and select **Migration in progress**.



The screenshot shows the 'Authentication methods | Policies' page in the Microsoft Entra ID Security portal. The left sidebar has 'Manage' selected, with 'Policies' highlighted. The main content area is titled 'Manage migration'. It contains a note about legacy multifactor authentication and self-service password reset policies being deprecated on September 30th, 2025. A red box highlights the 'Manage migration' button. Below it is a table listing various authentication methods with their target and enable status. A circular icon with a plus sign is in the bottom right corner.

Method	Target	Enabled
FIDO2 security key	All users	Yes
Microsoft Authenticator	All users	Yes
SMS		No
Temporary Access Pass	All users	Yes
Hardware OATH tokens (Preview)		No
Third-party software OATH tokens	All users	Yes
Voice call	All users	Yes
Email OTP		Yes
Certificate-based authentication		No

You set this option before you make any changes as it applies your new policy to both sign-in and password reset scenarios.

Manage migration

X

On September 30th, 2025, the legacy multifactor authentication and self-service password reset policies will be deprecated and you'll manage all authentication methods here in the authentication methods policy. Use this control to manage your migration from the legacy policies to the new unified policy.

[Learn more](#)

Pre-migration:

Use policy for authentication only, respect legacy policies.

Migration In Progress:

Use policy for authentication and SSPR, respect legacy policies.

Migration Complete:

Use policy for authentication and SSPR, ignore legacy policies.

The next step is to update the Authentication methods policy to match your audit. You'll want to review each method one-by-one. If your tenant is only using the legacy MFA policy, and isn't using SSPR, the update is straightforward - you can enable each method for all users and precisely match your existing policy.

If your tenant is using both MFA and SSPR, you'll need to consider each method:

- If the method is enabled in both legacy policies, enable it for all users in the Authentication methods policy.
- If the method is off in both legacy policies, leave it off for all users in the Authentication methods policy.
- If the method is enabled only in one policy, you need to decide whether, or not it should be available in all situations.

Where the policies match, you can easily match your current state. Where there's a mismatch, you'll need to decide whether to enable or disable the method altogether. For example, suppose **Notification through mobile app** is enabled to allow push notifications for MFA. In the legacy SSPR policy, the **Mobile app notification** method isn't enabled. In that case, the legacy policies allow push notifications for MFA but not SSPR.

In the Authentication methods policy, you'll then need to choose whether to enable **Microsoft Authenticator** for both SSPR and MFA or disable it (we recommend enabling Microsoft Authenticator).

Note that in the Authentication methods policy you have the option to enable methods for groups of users in addition to all users, and you can also exclude groups of users from being able to use a given method. This means you have a lot of flexibility to control what users can use which methods. For example, you can enable **Microsoft Authenticator** for all users and limit **SMS** and **Voice call** to 1 group of 20 users that need those methods.

As you update each method in the Authentication methods policy, some methods have configurable parameters that allow you to control how that method can be used. For example, if you enable **Voice calls** as authentication method, you can choose to allow both office phone and mobile phones, or mobile only. Step through the process to configure each authentication method from your audit.

You aren't required to match your existing policy! It's a great opportunity to review your enabled methods and choose a new policy that maximizes security and usability for your tenant. Just note that disabling methods for users who are already using them may require those users to register new authentication methods and prevent them from using previously registered methods.

The next sections cover specific migration guidance for each method.

Email one-time passcode

There are two controls for **Email one-time passcode**:

Under the **Enable and Target** section: Tenant members may be enabled to allow Email OTP for use in **Password Reset** with specific groups included or excluded (or enabled for all member users).

Under the **Configure** section: A separate **Allow external users to use email OTP** control enables use of email OTP for **sign-in** by B2B users. The Email OTP authentication method can't be disabled if this setting is enabled.

Microsoft Authenticator

If **Notification through mobile app** is enabled in the legacy MFA policy, enable **Microsoft Authenticator** for **All users** in the Authentication methods policy. Set the authentication mode to **Any** to allow either push notifications or passwordless authentication.

If **Verification code from mobile app or hardware token** is enabled in the legacy MFA policy, set **Allow use of Microsoft Authenticator OTP** to **Yes**.

Microsoft Authenticator settings

X

The Microsoft Authenticator app is a flagship authentication method, usable in passwordless or simple push notification approval modes. The app is free to download and use on Android/iOS mobile devices. [Learn more](#).

Enable and Target [Configure](#)

Note: Users must be included as part of the Microsoft Authenticator targeted groups under the 'Enable and Target' tab.

GENERAL

Allow use of Microsoft Authenticator OTP [Yes](#) [No](#)

Require number matching for push notifications

Note: This feature has been enabled for all users of the Microsoft Authenticator. [Learn more](#)

Status [Enabled](#)

Target [Include](#)

- All users
- Select group

ⓘ Note

If users register Microsoft Authenticator only for OTP code using the **I want to use a different authenticator app** wizard, it's needed to enable **Third-party software OATH tokens** policy.

SMS and voice calls

The legacy MFA policy has separate controls for **SMS** and **Phone calls**. But there's also a **Mobile phone** control that enables mobile phones for both SMS and voice calls. And another control for **Office phone** enables an office phone only for voice call.

The Authentication methods policy has controls for **SMS** and **Voice calls**, matching the legacy MFA policy. If your tenant is using SSPR and **Mobile phone** is enabled, you'll want to enable both **SMS** and **Voice calls** in the Authentication methods policy. If your tenant is using SSPR and **Office phone** is enabled, you'll want to enable **Voice calls** in the Authentication methods policy, and ensure that the **Office phone** option is enabled.

ⓘ Note

The **Use for sign-in** option is default enabled on **SMS** settings. This option enables SMS sign-in. If SMS sign-in is enabled for users, they're skipped from cross-tenant synchronization. If you are using cross-tenant synchronization or don't want to enable SMS sign-in, disable SMS Sign-in for target users.

OATH tokens

The OATH token controls in the legacy MFA and SSPR policies were single controls that enabled the use of three different types of OATH tokens: the Microsoft Authenticator

app, third-party software OATH TOTP code generator apps, and hardware OATH tokens.

The Authentication methods policy has granular control with separate controls for each type of OATH token. Use of OTP from Microsoft Authenticator is controlled by the **Allow use of Microsoft Authenticator OTP** control in the **Microsoft Authenticator** section of the policy. Third-party apps are controlled by the **Third party software OATH tokens** section of the policy. Hardware OATH tokens are controlled by the **Hardware OATH tokens** section of the policy.

Security questions

A control for **Security questions** is coming soon. If you use security questions, and don't want to disable them, make sure to keep them enabled in the legacy SSPR policy until the new control is available. You *can* finish migration as described in the next section with security questions enabled.

Finish the migration

After you update the Authentication methods policy, go through the legacy MFA, and SSPR policies and remove each authentication method one-by-one. Test and validate the changes for each method.

When you determine that MFA and SSPR work as expected and you no longer need the legacy MFA and SSPR policies, you can change the migration process to **Migration Complete**. In this mode, Microsoft Entra-only follows the Authentication methods policy. No changes can be made to the legacy policies if **Migration Complete** is set, except for security questions in the SSPR policy. If you need to go back to the legacy policies for some reason, you can move the migration state back to **Migration in Progress** at any time.

Manage migration

X

On September 30th, 2025, the legacy multifactor authentication and self-service password reset policies will be deprecated and you'll manage all authentication methods here in the authentication methods policy. Use this control to manage your migration from the legacy policies to the new unified policy.

[Learn more](#)

Pre-migration:

Use policy for authentication only, respect legacy policies.

Migration In Progress:

Use policy for authentication and SSPR, respect legacy policies.

Migration Complete:

Use policy for authentication and SSPR, ignore legacy policies.

Next steps

- [Manage authentication methods for Microsoft Entra ID](#)
- [What authentication and verification methods are available in Microsoft Entra ID?](#)
- [How Microsoft Entra multifactor authentication works](#)
- [Microsoft Graph REST API](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Configure Temporary Access Pass to register passwordless authentication methods

Article • 03/04/2025

Passwordless authentication methods like a passkey (FIDO2) let users sign in securely without a password. Users can bootstrap passwordless methods in one of two ways:

- Use existing Microsoft Entra multifactor authentication methods
- Use a Temporary Access Pass

A Temporary Access Pass (TAP) is a time-limited passcode that can be configured for single use or multiple sign-ins. Users can sign in with a TAP to onboard other passwordless authentication methods. A TAP also makes recovery easier when a user loses or forgets a strong authentication method.

This article shows you how to enable and use a TAP using the [Microsoft Entra admin center](#). You can also perform these actions using REST APIs.

Enable the Temporary Access Pass policy

A TAP policy defines settings, such as the lifetime of passes created in the tenant, or the users and groups who can use a TAP to sign-in.

Before users can sign-in with a TAP, you need to enable this method in the Authentication methods policy and choose which users and groups can sign in by using a TAP.

Although you can create a TAP for any user, only users included in the policy can sign-in with it. You need the [Authentication Policy Administrator](#) role to update the TAP Authentication methods policy.

To configure TAP in the Authentication methods policy:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Policies**.
3. From the list of available authentication methods, select **Temporary Access Pass**.

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation sidebar with various options like User settings, Groups, Devices, Applications, Roles & admins, Billing, Settings, Protection, Identity Protection, Conditional Access, Security Center, Identity Secure Score, Multifactor authentication, and Authentication methods. The 'Authentication methods' link is highlighted with a red box. The main content area shows the 'Authentication methods | Policies' page for Contoso - Microsoft Entra ID Security. It includes sections for Manage (with Policies highlighted with a red box), Manage migration (with a note about legacy policies being deprecated), and a table of authentication methods. The 'Temporary Access Pass' method is highlighted with a red box in the table.

4. Select **Enable** and then select users to include or exclude from the policy.

The screenshot shows the 'Temporary Access Pass settings' page. The left sidebar has links for Home, Favorites, Identity (Overview, Users, Groups, Devices, Applications, Protection, Authentication methods, Password reset, Custom security attributes), and Learn & support. The main content area shows the 'Temporary Access Pass settings' page with a description of what TAP is. It has sections for 'Enable and Target' (with the 'Enable' button highlighted with a red box) and 'Include' (with 'All users' selected). A table lists the target users as 'All users' type 'Group'.

5. (Optional) Select **Configure** to modify the default Temporary Access Pass settings, such as setting maximum lifetime, or length, and select **Update**.

The screenshot shows the 'Temporary Access Pass settings' page with more detailed configuration options. The left sidebar is the same as the previous screenshot. The main content area shows the 'Temporary Access Pass settings' page with a description of TAP. It has sections for 'Enable and Target' (with 'Configure' highlighted with a red box), 'GENERAL' (with 'Edit' highlighted with a red box), and configuration fields for 'Minimum lifetime', 'Maximum lifetime', 'Default lifetime', 'Length', 'Require one-time use', and 'Length (characters)'. The 'Update' button at the bottom right is highlighted with a red box.

6. Select **Save** to apply the policy.

The default value and the range of allowed values are described in the following table.

 Expand table

Setting	Default values	Allowed values	Comments
Minimum lifetime	1 hour	10 – 43,200 Minutes (30 days)	Minimum number of minutes that the TAP is valid.
Maximum lifetime	8 hours	10 – 43,200 Minutes (30 days)	Maximum number of minutes that the TAP is valid.
Default lifetime	1 hour	10 – 43,200 Minutes (30 days)	Individual passes within the minimum and maximum lifetime configured by the policy can override default value.
One-time use	False	True/False	When the policy is set to false, passes in the tenant can be used either once or more than once during its validity (maximum lifetime). By enforcing one-time use in the TAP policy, all passes created in the tenant are one-time use.
Length	8	8-48 characters	Defines the length of the passcode.

Create a Temporary Access Pass

After you enable a TAP policy, you can create a TAP policy for users in Microsoft Entra ID. The following roles can perform various actions related to a TAP.

- **Privileged Authentication Administrators** can create, delete, and view a TAP for admins and members (except themselves).
- **Authentication Administrators** can create, delete, and view a TAP for members (except themselves).
- **Authentication Policy Administrators** can enable TAP, include or exclude groups, and edit the Authentication methods policy.
- **Global Readers** can view TAP details for the user (without reading the code itself).

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Administrator**.

2. Browse to Identity > Users.

3. Select the user you would like to create a TAP for.

4. Select Authentication methods and select Add authentication method.

The screenshot shows the Microsoft Entra admin center interface. On the left, the navigation pane is open with categories like Home, Favorites, Identity, Overview, Users, Groups, Devices, Applications, Roles & admins, Billing, Settings, and Learn & support. Under the 'Users' section, 'All users' is selected. In the main content area, 'Adele Vance | Authentication methods' is displayed. A red box highlights the '+ Add authentication method' button. To the right, a modal window titled 'Add authentication method' is open. It has a dropdown menu 'Choose method' set to 'Temporary Access Pass'. Below it, instructions say 'Create a Temporary Access Pass for Adele Vance. While the pass is valid, the user can use it to sign in and register strong credentials.' There are checkboxes for 'Delayed start time' (unchecked) and 'Activation duration' (set to 1 hour). A radio button 'One-time use' is set to 'Yes'. At the bottom right of the modal is a blue 'Add' button.

5. Select Temporary Access Pass.

6. Define a custom activation time or duration and select Add.

This screenshot shows the 'Add authentication method' dialog with the following configuration: 'Choose method' is set to 'Temporary Access Pass'. The activation duration is set to 1 hour. The 'One-time use' option is selected. A red box highlights the blue 'Add' button at the bottom right.

7. Once added, the details of the TAP are shown.

ⓘ Important

Make a note of the actual TAP value, because you provide this value to the user. You can't view this value after you select Ok.

Temporary Access Pass details

X

Provide Pass

Provide this Temporary Access Pass to the user so they can set their strong credentials.

%3&L&CRM



Secure registration

To register their credentials, have the user go to My Security Info.

<https://aka.ms/mysecurityinfo>



Additional information

Valid from 2/27/2024, 8:08:22 PM

Valid until 2/27/2024, 9:08:22 PM

Created 2/27/2024, 8:08:23 PM



Remove lost devices from the user's account. This is especially important for devices used for user authentication.

Ok

8. Select **OK** when you're done.

The following commands show how to create and get a TAP using PowerShell.

PowerShell

```
# Create a Temporary Access Pass for a user
$properties = @{}
$properties.isUsableOnce = $True
$properties.startDateTime = '2022-05-23 06:00:00'
$propertiesJSON = $properties | ConvertTo-Json

New-MgUserAuthenticationTemporaryAccessPassMethod -UserId user2@contoso.com
-BodyParameter $propertiesJSON
```

Id	CreatedDateTime	IsUsable	
IsUsableOnce	LifetimeInMinutes	MethodUsabilityReason	StartTime
TemporaryAccessPass			
--			

00aa00aa-bb11-cc22-dd33-44ee44ee44ee 5/22/2022 11:19:17 PM False True			
60	NotYetValid	23/05/2022 6:00:00 AM	TAP Rocks!
# Get a user's Temporary Access Pass			
Get-MgUserAuthenticationTemporaryAccessPassMethod -UserId user3@contoso.com			
Id	CreatedDateTime	IsUsable	
IsUsableOnce	LifetimeInMinutes	MethodUsabilityReason	StartTime
TemporaryAccessPass			
--			

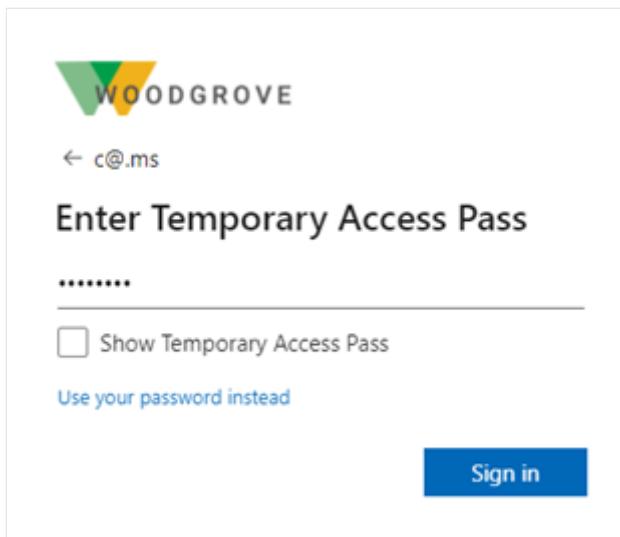
00aa00aa-bb11-cc22-dd33-44ee44ee44ee 5/22/2022 11:19:17 PM False True			
60	NotYetValid	23/05/2022 6:00:00 AM	

For more information, see [New-MgUserAuthenticationTemporaryAccessPassMethod](#) and [Get-MgUserAuthenticationTemporaryAccessPassMethod](#).

Use a Temporary Access Pass

The most common use for a TAP is for a user to register authentication details during the first sign-in or device setup, without the need to complete extra security prompts. Authentication methods are registered at <https://aka.ms/mysecurityinfo>. Users can also update existing authentication methods here.

1. Open a web browser to <https://aka.ms/mysecurityinfo>.
2. Enter the UPN of the account you created the TAP for, such as *tapuser@contoso.com*.
3. If the user is included in the TAP policy, they see a screen to enter their TAP.
4. Enter the TAP that was displayed in the Microsoft Entra admin center.



ⓘ Note

For federated domains, a TAP is preferred over federation. A user with a TAP completes the authentication in Microsoft Entra ID and isn't redirected to the federated Identity Provider (IdP).

The user is now signed in and can update or register a method such as FIDO2 security key. Users who update their authentication methods due to losing their credentials or device should make sure they remove the old authentication methods. Users can also continue to sign-in by using their password; a TAP doesn't replace a user's password.

User management of Temporary Access Pass

Users managing their security information at <https://aka.ms/mysecurityinfo> see an entry for the Temporary Access Pass. If a user doesn't have any other registered methods, they get a banner at the top of the screen that says to add a new sign-in method. Users can also see the TAP expiration time, and delete the TAP if it's no longer needed.

A screenshot of the 'My Sign-Ins' page in the Microsoft Entra portal. The left sidebar shows 'Overview', 'Security info' (which is selected), 'Organizations', 'Devices', and 'Privacy'. The main area has a green banner at the top with the text 'ⓘ To maintain access to your account, add a sign in method.' Below this is a section titled 'Security info' with the sub-instruction 'These are the methods you use to sign into your account or reset your password.' It lists a single method: 'Temporary access pass' (with a delete link) and a link 'Lost device? Sign out everywhere'.

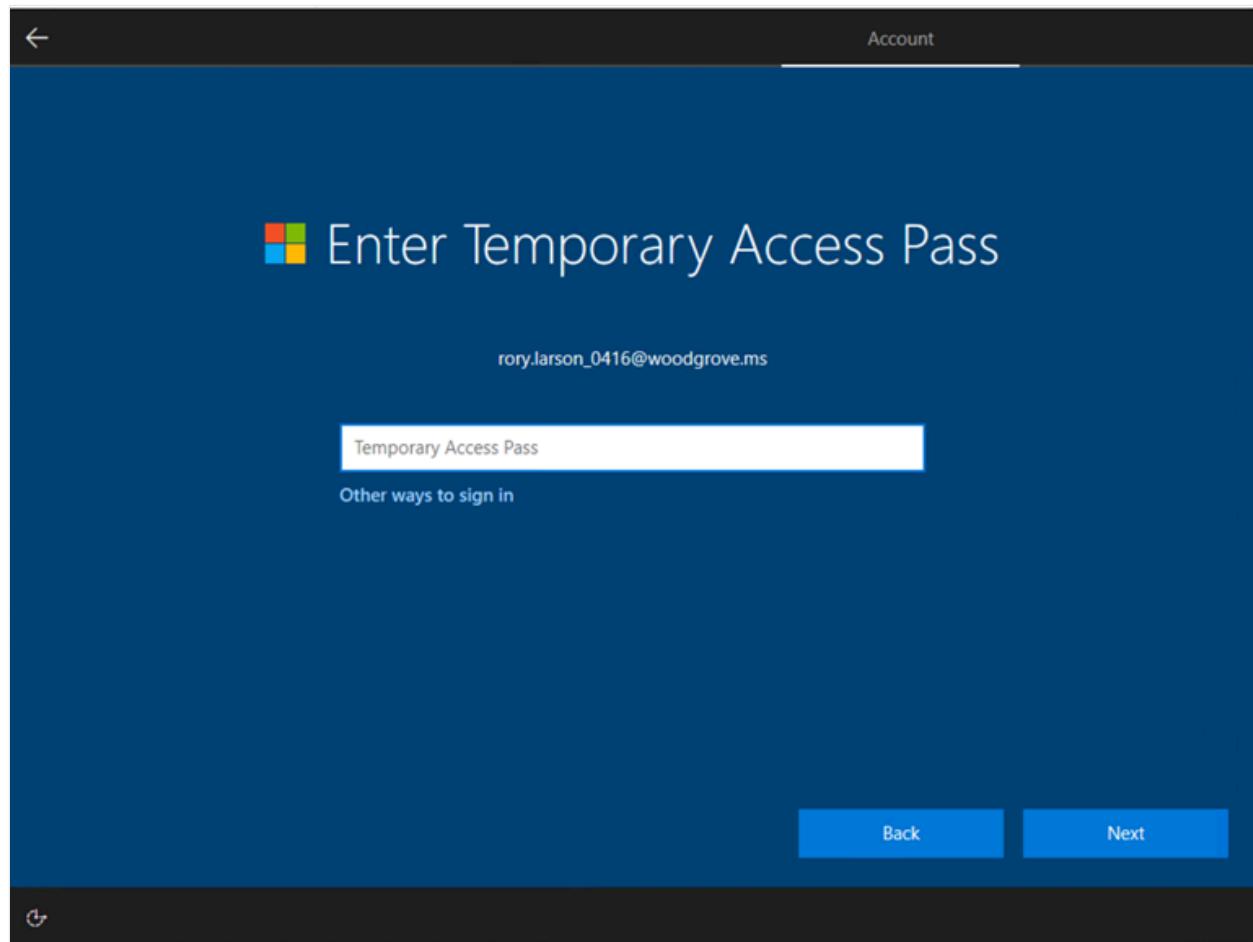
Windows device setup

Users with a TAP can navigate the setup process on Windows 10 and 11 to perform device join operations and configure Windows Hello for Business. TAP usage for setting up Windows Hello for Business varies based on the devices joined state.

For joined devices to Microsoft Entra ID:

- During the domain-join setup process, users can authenticate with a TAP (no password required) to join the device and register Windows Hello for Business.
- On already-joined devices, users must first authenticate with another method such as a password, smartcard, or FIDO2 key, before using TAP to set up Windows Hello for Business.
- If the [Web sign-in](#) feature on Windows is also enabled, the user can use TAP to sign into the device. This is intended only for completing initial device setup, or recovery when the user doesn't know or have a password.

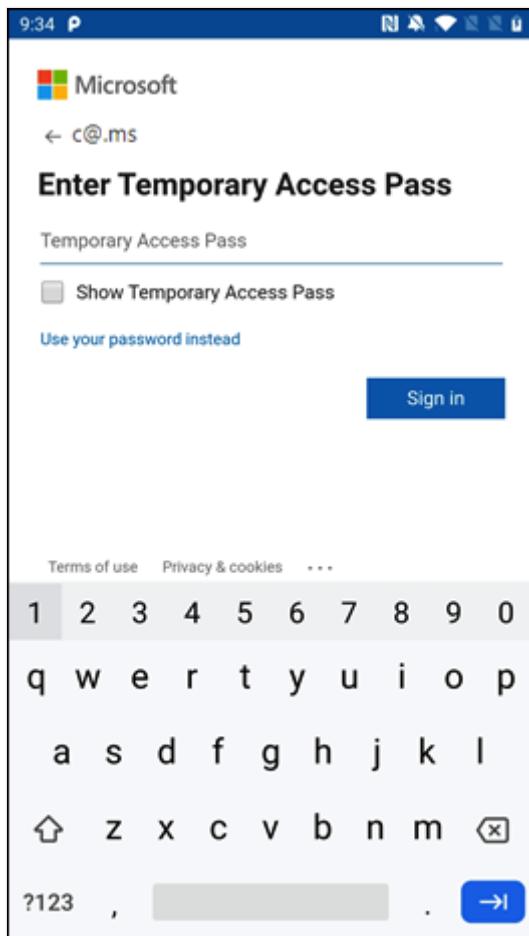
For hybrid-joined devices, users must first authenticate with another method such as a password, smartcard or FIDO2 key, before using TAP to set up Windows Hello for Business.



Using TAP with Microsoft Authenticator

Users can also use their TAP to register Microsoft Authenticator with their account. By adding a work or school account and signing in with a TAP users can register both passkeys and passwordless phone sign-in directly from the Authenticator app.

For more information, see [Add your work or school account to the Microsoft Authenticator app](#).



Guest access

You can add a TAP as a sign-in method to an internal guest, but not other types of guests. An internal guest has user object **UserType** set to **Guest**. They have authentication methods registered in Microsoft Entra ID. For more information about internal guests and other guest accounts, see [B2B guest user properties](#).

If you try to add a TAP to an external guest account in the Microsoft Entra admin center or in Microsoft Graph, you'll receive an error stating **Temporary Access Pass cannot be added to an external guest user**.

External guest users can sign-in to a resource tenant with a TAP issued by their home tenant if the TAP meets the home tenant authentication requirements and Cross Tenant Access policies have been configured to trust MFA from the users home tenant, see [Manage cross-tenant access settings for B2B collaboration](#).

Expiration

An expired or deleted TAP can't be used for interactive or non-interactive authentication.

Users need to reauthenticate with different authentication methods after the TAP is expired or deleted.

The token lifetime (session token, refresh token, access token, and so on) obtained by using a TAP sign-in is limited to the TAP lifetime. When a TAP expires, it leads to the expiration of the associated token.

Delete an expired Temporary Access Pass

Under the **Authentication methods** for a user, the **Detail** column shows when the TAP expired. You can delete an expired TAP using the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Administrator](#).
2. Browse to **Identity > Users**, select a user, such as *Tap User*, then choose **Authentication methods**.
3. On the right-hand side of the **Temporary Access Pass** authentication method shown in the list, select **Delete**.

You can also use PowerShell:

```
PowerShell
```

```
# Remove a user's Temporary Access Pass
Remove-MgUserAuthenticationTemporaryAccessPassMethod -UserId
user3@contoso.com -TemporaryAccessPassAuthenticationMethodId 00aa00aa-bb11-
cc22-dd33-44ee44ee44ee
```

For more information, see [Remove-MgUserAuthenticationTemporaryAccessPassMethod](#).

Replace a Temporary Access Pass

- Each user can only have one TAP. The passcode can be used during the start and end time of the TAP.
- If a user requires a new TAP:
 - If the existing TAP is valid, the admin can create a new TAP to override the existing valid TAP.
 - If the existing TAP has expired, a new TAP overrides the existing TAP.

For more information about NIST standards for onboarding and recovery, see [NIST Special Publication 800-63A](#).

Limitations

Keep these limitations in mind:

- When using a one-time TAP to register a passwordless method such as a FIDO2 security key or phone sign-in, the user must complete the registration within 10 minutes of sign-in with the one-time TAP. This limitation doesn't apply to a TAP that can be used more than once.
- Users in scope for self service password reset (SSPR) registration policy or [Microsoft Entra ID Protection multifactor authentication registration policy](#) are required to register authentication methods after they've signed in with a TAP using a browser. Users in scope for these policies are redirected to the [Interrupt mode of the combined registration](#). This experience doesn't currently support FIDO2 and phone sign-in registration.
- A TAP can't be used with the Network Policy Server (NPS) extension and Active Directory Federation Services (AD FS) adapter.
- It can take a few minutes for changes to replicate. Because of this, after a TAP is added to an account, it can take a while for the prompt to appear. For the same reason, after a TAP expires, users may still see a prompt for TAP.

Troubleshooting

- If a TAP isn't offered to a user during sign-in:
 - Make sure the user is in scope for TAP use in the Authentication methods policy.
 - Make sure the user has a valid TAP, and if it's one-time use, it wasn't used yet.
- If **Temporary Access Pass sign in was blocked due to User Credential Policy** appears during sign-in with a TAP:
 - Check that the user is in scope for the TAP policy
 - Make sure the user doesn't have a TAP for multiple use while the Authentication methods policy requires a one-time TAP.
 - Check if a one-time TAP was already used.
- If **Temporary Access Pass cannot be added to an external guest user** appears when you try to add a TAP to an account as an authentication method, the account is an external guest. Both internal and external guest accounts have an option to add a TAP for sign-in in the Microsoft Entra admin center and Microsoft Graph APIs. However, only internal guest accounts can be issued a TAP.

Next steps

- Plan a passwordless authentication deployment in Microsoft Entra ID
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

How to enable the QR code authentication method in Microsoft Entra ID (Preview)

Article • 03/21/2025

This topic covers how to enable the QR code authentication method in the Authentication methods policy in Microsoft Entra ID. It also covers how to manage the QR code authentication method for users, and how they can sign in with a QR code and PIN.

Prerequisites to enable the QR code authentication method

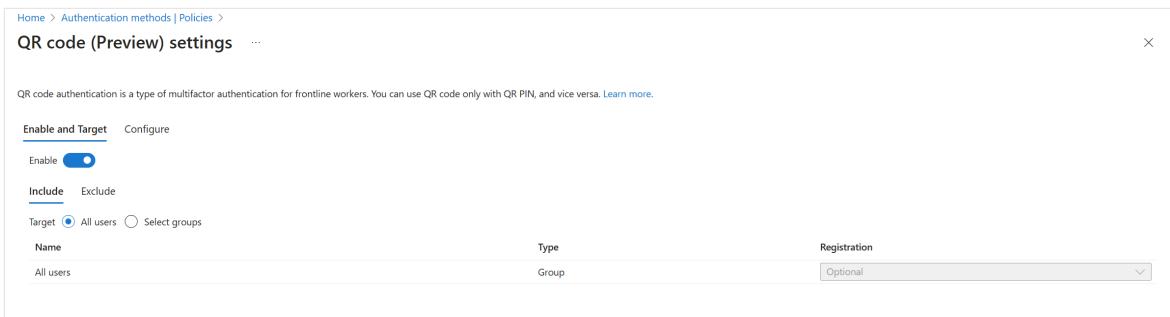
- An active Azure subscription.
 - If you don't have an Azure subscription, [create an account](#).
- A Microsoft Entra tenant associated with your subscription.
 - If needed, [create a Microsoft Entra tenant](#) or [associate an Azure subscription with your account](#).
- You need at least the [Authentication Policy Administrator](#) role in your Microsoft Entra tenant to enable the QR code authentication method.
- Each user that's enabled in the QR code authentication method policy must be licensed, even if they don't use it. Each enabled user must have one of the following Microsoft Entra ID, EMS, Microsoft 365 licenses:
 - [Microsoft 365 F1 or F3](#)
 - [Microsoft Entra ID P1 or P2][azure-ad-pricing]
 - [Enterprise Mobility + Security \(EMS\) E3 or E5](#) or [Microsoft 365 E3 or E5](#)
 - [Office 365 F3](#)
- Android, iOS, or iPadOS (iOS/iPadOS version 15.0 or later) shared devices.
- Shared device mode enabled on the shared devices (optional but highly recommended).
- A printer to print 2" x 2" QR codes.
- To access QR code authentication on Teams, Teams app installed on the shared device would require these versions: Android version 1.0.0.2024143204 or later, and iOS version 1.0.0.77.2024132501 or later.
- [Enable and setup My Staff portal](#) if you plan for frontline managers to use My Staff to provision, manage, and reset QR code and PINs.

Enable QR code authentication method

You can enable the QR code authentication method by using the Microsoft Entra admin center or Microsoft Graph API.

Enable QR code authentication method in the Microsoft Entra admin center

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Go to **Protection > Authentication methods > Policies**.
3. Click **QR code > Enable and target > Add target** > select a group of users who need to sign in with a QR code.



4. Update default QR code settings as needed:

- By default, the PIN length is 8 digits. The PIN length can be 8 to 20 digits. If you increase the PIN length, the new value becomes the minimum number of digits required for the PIN. For example, if you increase the PIN length to 10, a user needs to provide a 10-digit PIN during next sign-in.
- The default lifetime of a standard QR code (provided to the users for long term use) is 365 days. The range is between 1-395 days. You can change the lifetime of a standard QR code for specific user when you add the QR code authentication method for them.

QR code settings

QR code authentication is a simple and fast authentication method for frontline workers. To use it, you need both a QR code and PIN.

Enable and Target [Configure](#)

Defaults

QR PIN Length * ⓘ

8

Lifetime of standard QR code (days) * ⓘ

200

- When you're done, click **Save**.

Enable QR code authentication method in Microsoft Graph API

This example enables QR code authentication for a group, with a PIN length of 10 digits, and a Standard QR code lifetime of 395 days:

- Request

```
https
PATCH
https://graph.microsoft.com/beta/policies/authenticationMethodsPolicy/authenticationsMethodConfigurations/qrCodePin
{
  "@odata.type" :
  "microsoft.graph.qrCodePinAuthenticationMethodConfiguration",
  "id": "qrCodePin",
  "state": "enabled",
  "includeTargets": [
    {
      "targetType": "group",
      "id": "b185b746-e7db-4fa2-bafc-69ecf18850dd",
    }
  ],
  "excludeTargets": [],
  "standardQRCodeLifetimeInDays":395,
  "pinLength": 10
}
```

- Response

```
https
```

```
204 No Response
```

Add QR code authentication method for a user

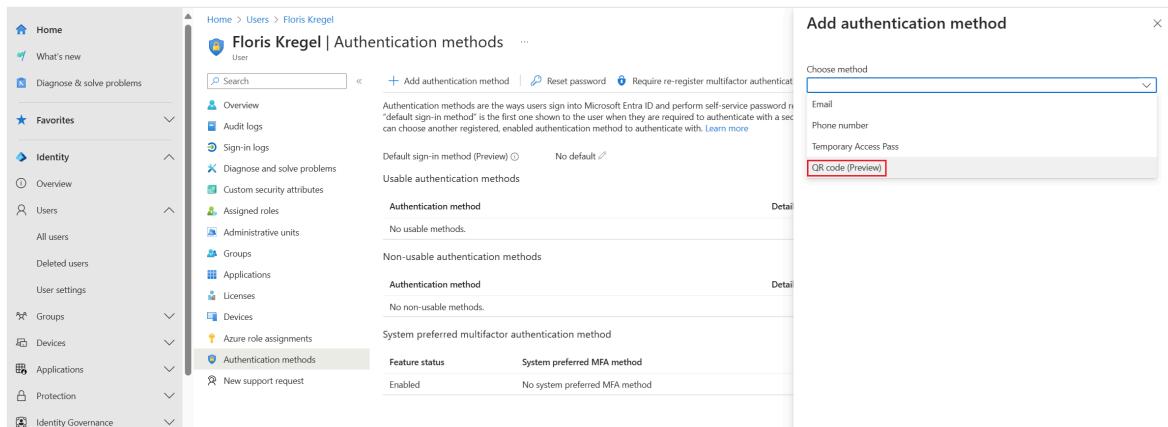
You can add a QR code authentication method for a user by using the Microsoft Entra admin center, My Staff, or Microsoft Graph API. At a time, only one active QR code auth method is allowed. Standard QR code is generated during 'Add authentication method'. You can add Temporary QR code, which is short-lived, if user is not carrying Standard QR code. You can delete Standard/Temporary QR code to add a new Standard/Temporary QR code. A user can have only one Standard and one Temporary QR code active at any point of time.

Add QR code authentication method for a user in the Microsoft Entra admin center

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Administrator**.

2. Go to **Users**, select a user, and click **Authentication methods**.

3. Click **Add authentication method** and choose **QR code**.



4. Modify the expiration date for the user if needed. Set **Activation time** to now or later. Provide or generate a temporary PIN. The custom PIN can be specified only when you add the QR code authentication method. A PIN is autogenerated during reset events. When ready, click **Add** to add the QR code authentication method for the user.

Add authentication method

X

Choose method

QR code (Preview)



Standard QR code

This is the primary QR code associated with the authentication method. Once the standard QR code and PIN are generated, you can add a temporary QR code. [Learn more](#)

Expiration * ⓘ

08/22/2025



11:59 PM

(UTC-08:00) Pacific Time (US & Canada)

Activation time *

Now

Later

PIN

This PIN works for both standard and temporary QR codes. Users must change it when they first sign in.

PIN * ⓘ

52979809



[Generate PIN](#)

5. Save the PIN, and click **Download image** to download and print the QR code. The QR code image download has the smallest optimal print size. If you reduce the size of the QR code, it may impact QR code scan performance.

You can't regenerate the same QR code because it has a unique secret. If the QR code can't work for some reason, delete it. Create a new QR code for the user.

Download QR code and Save PIN

X

⚠ This QR code is only available for download right now. Download and save it in a secure place. If you need a QR code in future, simply delete this one and generate a new standard QR code.

Copy PIN

PIN

52979809 

Standard QR code



QR code ID: 38c198ca-1785-4331-ac21-c5408b9c3ca2

Activation time: 2/4/2025, 11:30:25 AM

Expiration: 8/22/2025, 11:59:59 PM

Download image

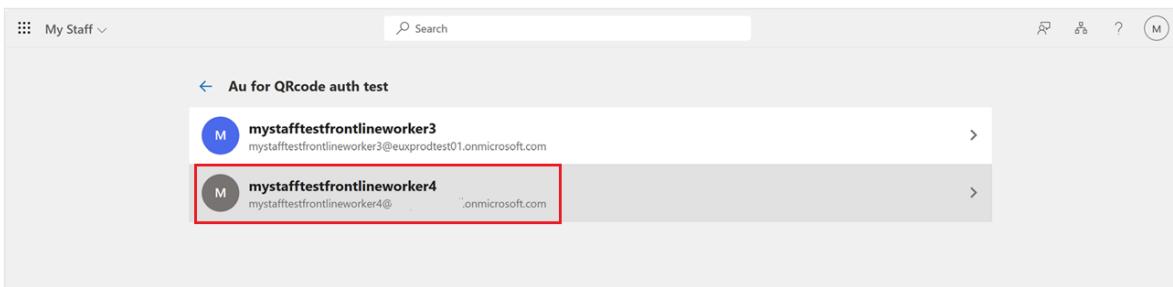
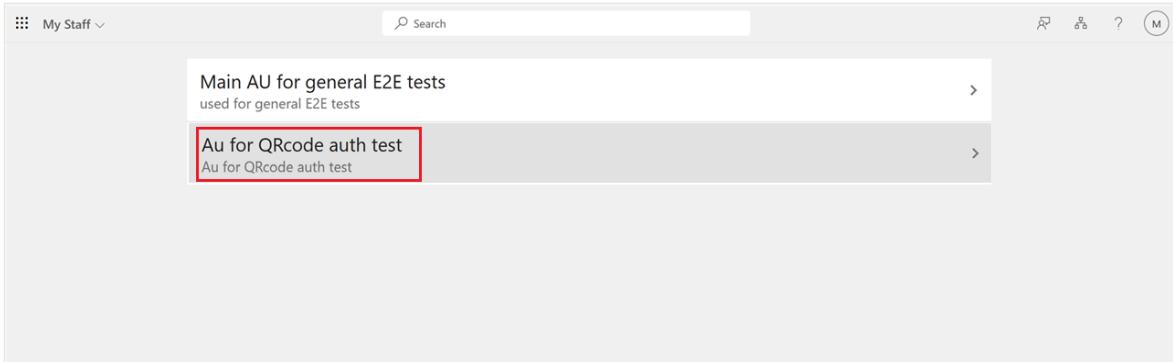
- After you add the QR code authentication method, it appears as a usable authentication method for the user.

The screenshot shows the Microsoft Entra ID portal. On the left, there's a navigation sidebar with links like Home, What's new, Diagnose & solve problems, Favorites, Identity, Overview, Users (All users, Deleted users, User settings), Groups, Devices, Applications, Protection, and Identity Governance. The main area is titled "Floris Kregel | Authentication methods". It shows a table of authentication methods:

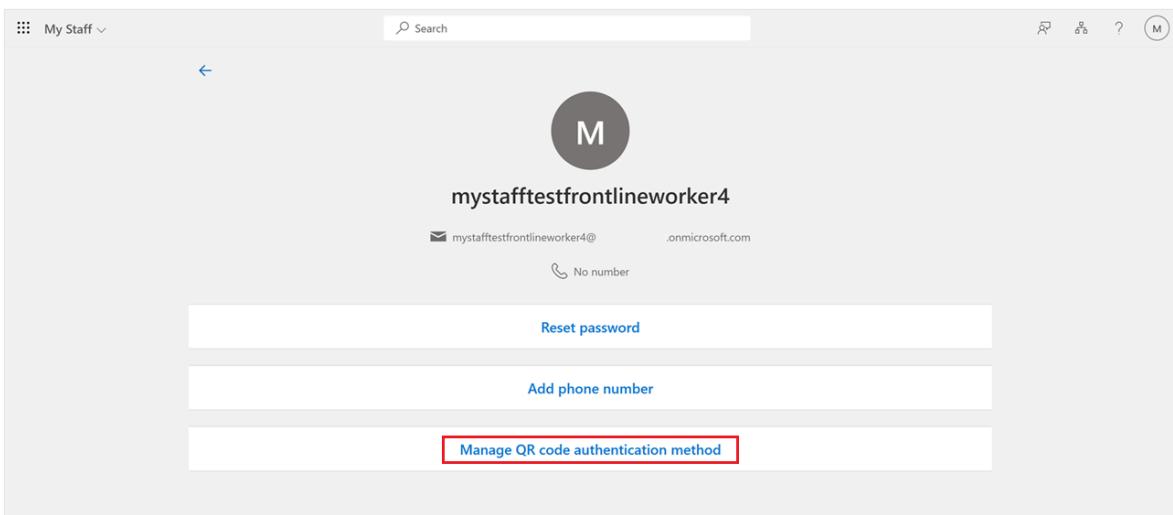
Authentication method	Detail
QR code (Preview)	QR code expires 8/22/2025, 11:59:59 PM
No non-usuable methods.	
System preferred multifactor authentication method	
Feature status	System preferred MFA method
Enabled	No system preferred MFA method

Add the QR code authentication method for a user in My Staff

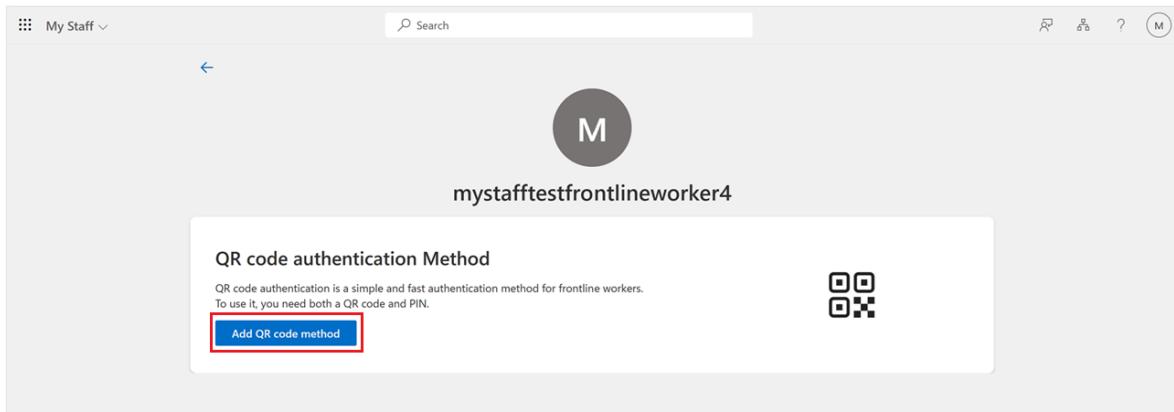
1. Sign in to the My Staff portal as a frontline manager. Select an administrative unit and a frontline worker.



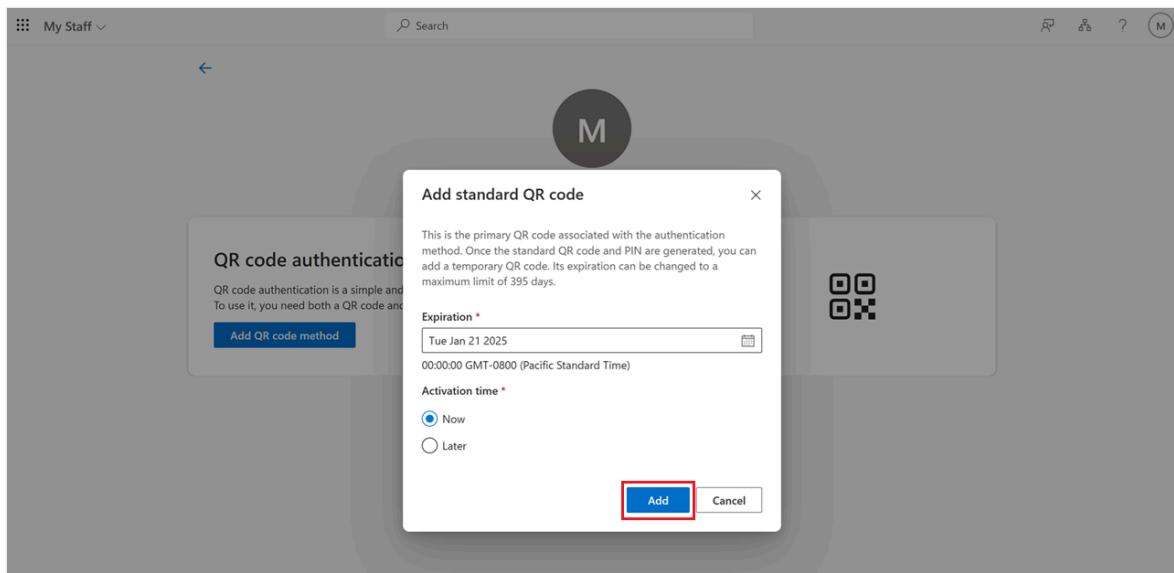
2. Click Manage QR code authentication method.



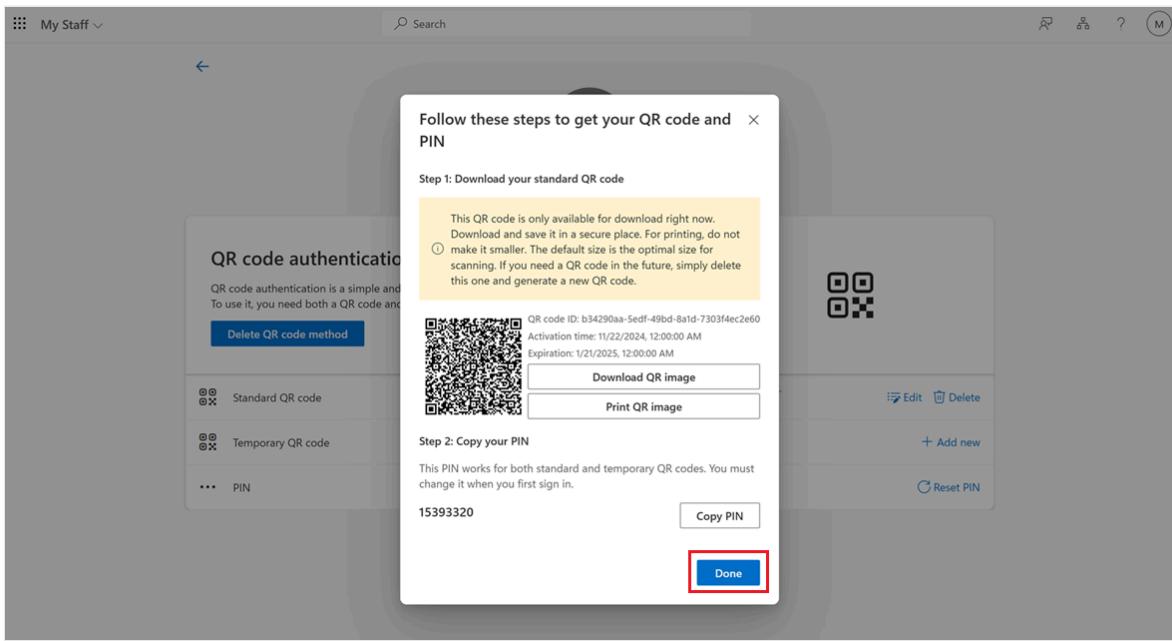
3. Click Add QR code method.



4. Specify the expiration and activation date, and click **Add** to generate a QR code and PIN for the user.



5. Save the PIN, download or print the QR code, and then click **Done**. The QR code image download has the smallest optimum print size. If you reduce the size, the QR code is hard to scan. You can't regenerate the same QR code because it has a unique secret. If the QR code can't work for some reason, delete it. Create a new QR code for the user.



Add QR code authentication method for a user in Microsoft Graph API

This example adds QR code authentication method for a user:

- Request

```
https

HTTP PUT/users/{id | userPrincipalName}/authentication/qrCodePinMethod

{
  "standardQRCode": {
    "expireDateTime": "2024-12-30T12:00:00Z",
    "startDateTime": "2024-10-30T12:00:00Z"
  },
  "pin": {
    "code": "<PIN>"
  }
}
```

- Response

```
https

HTTP/1.1 201 Created
Location: /beta/users/aaaaaaaa-bbbb-cccc-1111-
2222222222/authentication/qrCodePinMethod
Content-type: application/json

{
```

```
"standardQRCode": {
    "id": "BBBBBBBBB-1C1C-2D2D-3E3E-444444444444"
    "expireDateTime": "2024-12-30T12:00:00Z",
    "startDateTime": "2024-10-30T12:00:00Z"
    "createdDateTime": "2024-10-30T12:00:00Z",
    "lastUsedDateTime": null,
    "image":
        {
    "binaryValue": "<binaryImageData>",
        "version": 1,
        "errorCorrectionLevel": "H".
        "rawContent": <binary data encoded in QR>
    }
},
"temporaryQRCode": null,
"pin": {
    "code": "<PIN>",
    "isForcePinChangeRequired": true,
    "createdDateTime": "2024-10-30T12:00:00Z",
    "updatedDateTime": null
}
}
```

This example confirms whether QR code authentication method is added for the user:

- **Request**

```
https

GET
https://graph.microsoft.com/beta/users/flokreg@contoso.com/authentication/qrCodePinMethod`
```

- **Response**

```
https

HTTP/1.1 200 OK
Content-type: application/json

{
    "id": "<id>",
    "standardQRCode": {
        "id": "BBBBBBBBB-1C1C-2D2D-3E3E-444444444444"
        "image": null,
        "expireDateTime": "2024-12-30T12:00:00Z",
        "startDateTime": "2024-10-30T12:00:00Z"
        "createdDateTime": "2024-10-30T12:00:00Z",
        "lastUsedDateTime": "2024-12-30T12:00:00Z"
    },
    "temporaryQRCode": {
```

```

    "id": "CCCCCCCC-2D2D-3E3E-4F4F-555555555555",
    "image": null,
    "expireDateTime": "2024-12-30T12:00:00Z",
    "startDateTime": "2024-10-30T12:00:00Z",
    "createdDateTime": "2024-10-30T12:00:00Z",
    "lastUsedDateTime": "2024-12-30T12:00:00Z"
  },
  "pin": {
    "code": null,
    "isForcePinChangeRequired": false,
    "createdDateTime": "2024-10-30T12:00:00Z",
    "updatedDateTime": "2024-11-30T12:00:00Z"
  }
}

```

Edit the QR code authentication method for a user

You can edit QR code authentication method for a user by using the Microsoft Entra admin center, My Staff, or Microsoft Graph API.

Edit the QR code authentication method for a user in the Microsoft Entra admin center

- Navigate to the usable authentication methods for a user, and click **Edit** to edit the properties of the QR code authentication method.

The screenshot shows the Microsoft Entra admin center interface for managing user authentication methods. The left sidebar lists user management options: Overview, Audit logs, Sign-in logs, Diagnose and solve problems, Custom security attributes, Assigned roles, Administrative units, Groups, Applications, Licenses, Devices, Azure role assignments, Authentication methods (which is selected), and New support request. The main content area is titled 'Floris Kregel | Authentication methods'. It includes a search bar and several action buttons: Add authentication method, Reset password, Require re-register multifactor authentication, Revoke multifactor authentication sessions, and more. The 'Default sign-in method (Preview)' is set to 'No default'. Under 'Usable authentication methods', the 'QR code (Preview)' method is listed with a detail row showing it expires on 8/22/2025 at 11:59:59 PM. A red box highlights the 'Edit' button in the detail row's context menu. Other sections include 'Non-usuable authentication methods' (empty) and 'System preferred multifactor authentication method' (disabled).

- Change the expiration time for the standard QR code, and click **Save**. After you make edits, click **Done**.

Edit authentication method

User
Floris Kregel

Standard QR code
This is the user's primary QR code. Its expiration can be changed to a maximum limit of 13 months. [Learn more](#)

QR code ID: 38c198ca-1785-4331-ac21-c5408b9c3ca2

Activation time: 2/4/2025, 11:30:25 AM Pacific Standard Time

Expiration: 8/22/2025, 11:59:59 PM Pacific Standard Time [Edit](#)

Delete

PIN
This PIN works for both standard and temporary QR codes. Users must change it when they first sign in. [Learn more](#)

Pin: ***** [Edit](#)

Temporary QR code
This is a temporary QR code, and should only be given to a user who has lost or didn't bring their standard QR code. [Learn more](#)

Temporary QR code [+ Add Temporary QR code](#)

Done **Save**

- Delete a standard QR code. You might want to delete the standard QR code if it's reported as expired, compromised, or stolen.

Edit authentication method

User
Floris Kregel

Standard QR code
This is the user's primary QR code. Its expiration can be changed to a maximum limit of 13 months. [Learn more](#)

QR code ID: 38c198ca-1785-4331-ac21-c5408b9c3ca2

Activation time: 2/4/2025, 11:30:25 AM Pacific Standard Time

Expiration: 3/4/2025, 11:59:59 PM Pacific Standard Time [Edit](#)

Delete

PIN
This PIN works for both standard and temporary QR codes. Users must change it when they first sign in. [Learn more](#)

Pin: ***** [Edit](#)

Temporary QR code
This is a temporary QR code, and should only be given to a user who has lost or didn't bring their standard QR code. [Learn more](#)

Temporary QR code [+ Add Temporary QR code](#)

Done **Delete**

Delete Standard QR code

The following Standard QR code will be permanently deleted

Standard QR code
User: e1210b52-d9fe-4e43-867b-7a31f2935be0
QR code ID: 38c198ca-1785-4331-ac21-c5408b9c3ca2
Activation time: 2/4/2025, 11:30:25 AM Pacific Standard Time
Expiration: 3/4/2025, 11:59:59 PM Pacific Standard Time

Delete Confirmation

- Deleting this QR code is a permanent action and cannot be undone.

Delete **Go back**

After you delete the standard QR code, click the add symbol (+) to add a new standard QR code for the user. The deleted QR code is no longer valid for login.

You need to print and distribute the new QR code to the user. The user can continue to use their existing PIN.

The screenshot shows the 'Edit authentication method' page for a user named Floris Kregel. In the top right corner, there is a success message: 'Successfully deleted method' and 'Successfully deleted Standard QR code'. On the left, under 'Standard QR code', it says 'This is the user's primary QR code. Its expiration can be changed to a maximum limit of 13 months. Learn more'. Below this is a 'Pin' section with a masked PIN (*****). A note says 'You can't view the PIN, but you can change it.' On the right, there is a 'Temporary QR code' section with a note about temporary QR codes. At the bottom, there is a 'Done' button.

- Reset a PIN. If you need to reset a user PIN, generate a temporary one and distribute it to the user. The user will be required to change the temporary PIN at the next sign-in. Click the pencil icon after the masked PIN. Click **Generate new PIN** to create a new temporary PIN. Click **OK** to confirm that the user is forced to change the temporary PIN when they next sign in. Copy the temporary PIN and share it with the user.

The screenshot shows the 'Edit authentication method' page for a user named Floris Kregel. On the right, there is an 'Edit PIN' panel. It shows a text input field with the PIN '44802906' and a 'Generate PIN' button below it. A warning message says '⚠ This PIN is only visible right now. Please copy it and save it in a secure place.' On the left, under 'Standard QR code', it says 'This is the user's primary QR code. Its expiration can be changed to a maximum limit of 13 months. Learn more'. Below this is a 'Pin' section with a masked PIN (*****). A note says 'You can't view the PIN, but you can change it.' On the right, there is a 'Temporary QR code' section with a note about temporary QR codes. At the bottom, there is a 'Done' button.

- Add or delete a temporary QR code. A temporary QR code reduces admin overhead of provisioning and deprovisioning the QR code on a badge if a user didn't bring their badge to work. It also reduces the stress of retaining the QR code after their shift. A temporary QR code has a lifetime of 1-12 hours and can be activated instantly or later. To deprovision the QR code, you can delete the temporary QR code or let it expire as it's unusable after expiry.

Home > Users > Floris Kregel | Authentication methods >

Edit authentication method

User
Floris Kregel

Standard QR code
This is the user's primary QR code. Its expiration can be changed to a maximum limit of 13 months. [Learn more](#)

QR code ID: 769c91a6-b6a1-440e-99fb-55ad367209d2

Activation time: 2/5/2025, 12:03:23 AM Pacific Standard Time

Expiration: 8/5/2025, 11:59:59 PM Pacific Standard Time [Edit](#)

[Delete](#)

PIN
This PIN works for both standard and temporary QR codes. Users must change it when they first sign in. [Learn more](#)

Pin: *****

! You can't view the PIN, but you can change it.

Temporary QR code
This is a temporary QR code, and should only be given to a user who has lost or didn't bring their standard QR code. [Learn more](#)

[Temporary QR code](#) [+ Add Temporary QR code](#)

Add Temporary QR code

Lifetime in hours * 3

Activation time *
 Now Later

Start time *
02/05/2025 12:09:55 AM
(UTC-08:00) Pacific Time (US & Canada)

Download QR code and Save PIN

⚠️ This QR code is only available for download right now. Download and save it in a secure place. If you need a QR code in future, simply delete this one and generate a new standard QR code.

Standard QR code



QR code ID: de58518f-1f14-4922-ae16-f3a1ef5a2162

Activation time: 2/5/2025, 12:09:55 AM

Expiration: 2/5/2025, 3:09:55 AM

[Download image](#)

Edit the QR code authentication method for a user in My Staff

- To edit the expiration date for a standard QR code, click **Edit**. Edit the expiration date and save the changes.

The screenshot shows the 'QR code authentication Method' section for a user named 'mystafftestfrontlineworker4'. It displays a standard QR code with a red border and a PIN code below it. A 'Delete QR code method' button is visible. Below the QR code, there is a table with three rows:

		Activated on	Expires on	
Standard QR code	Not Added	Fri Nov 22 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Tue Jan 21 2025 00:00:00 GMT-0800 (Pacific Standard Time)	Edit Delete
Temporary QR code	Not Added			+ Add new

A 'Reset PIN' link is also present.

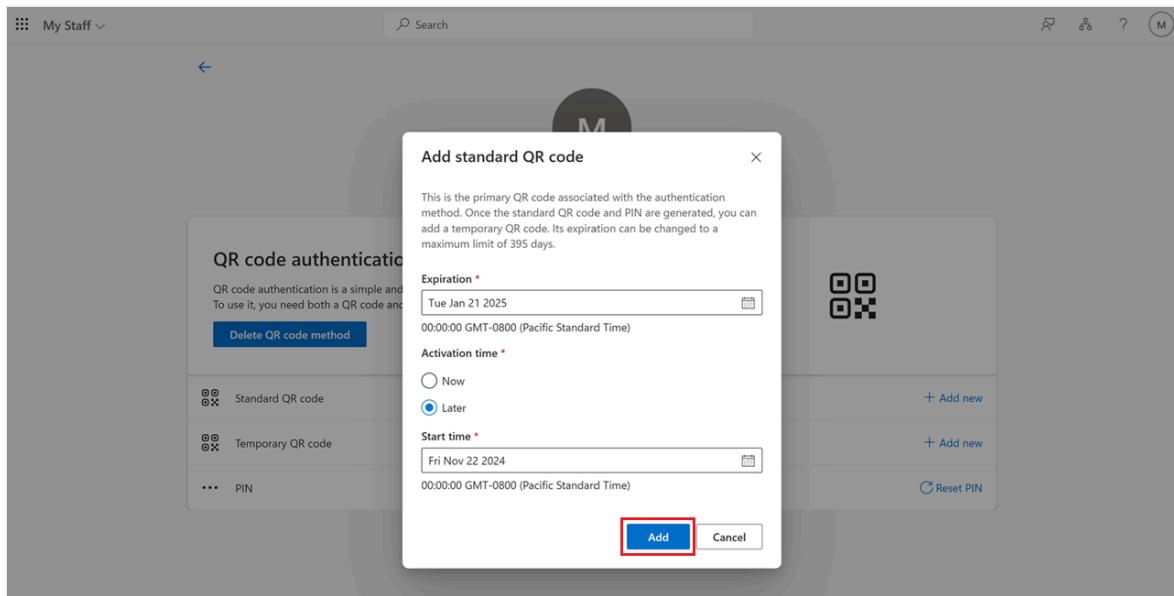
- To delete a standard QR code, click **Delete**, and confirm the action.

The screenshot shows the same 'QR code authentication Method' section. The 'Delete QR code method' button for the standard QR code row has been clicked, and a red box highlights the 'Delete' link in the confirmation dialog. The table now shows the standard QR code row with a crossed-out icon and a note indicating it has been deleted.

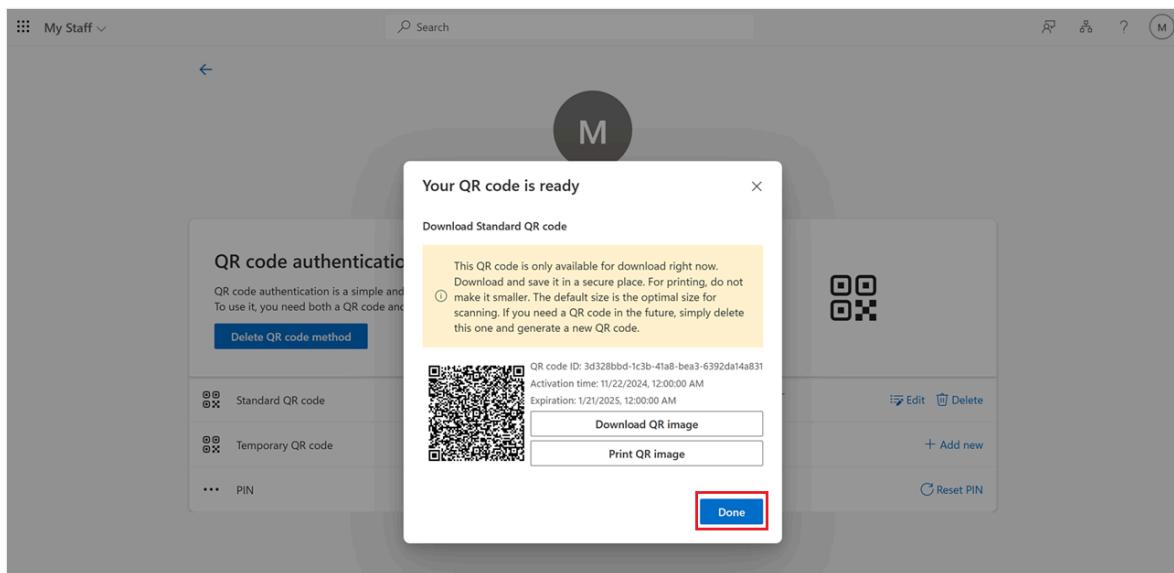
- To add a new standard QR code, click **Add new** next to the standard QR code.

The screenshot shows the same 'QR code authentication Method' section. The '+ Add new' link next to the standard QR code row has been clicked, and a red box highlights this link. The table now shows the standard QR code row with a plus sign icon and a note indicating it is not added yet.

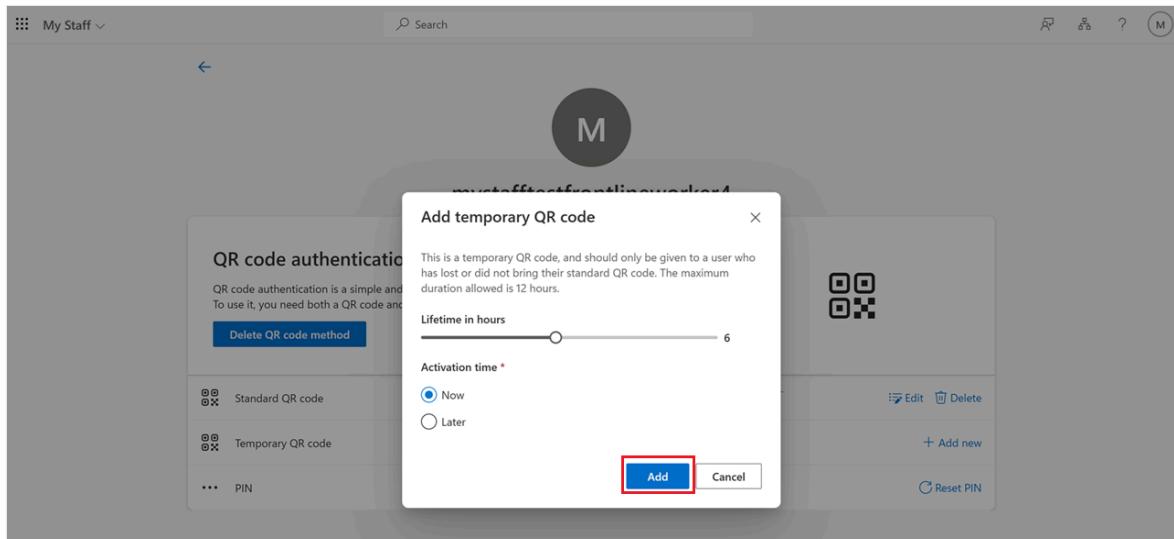
Select the activation time and expiration date for the QR code, and click **Add**.



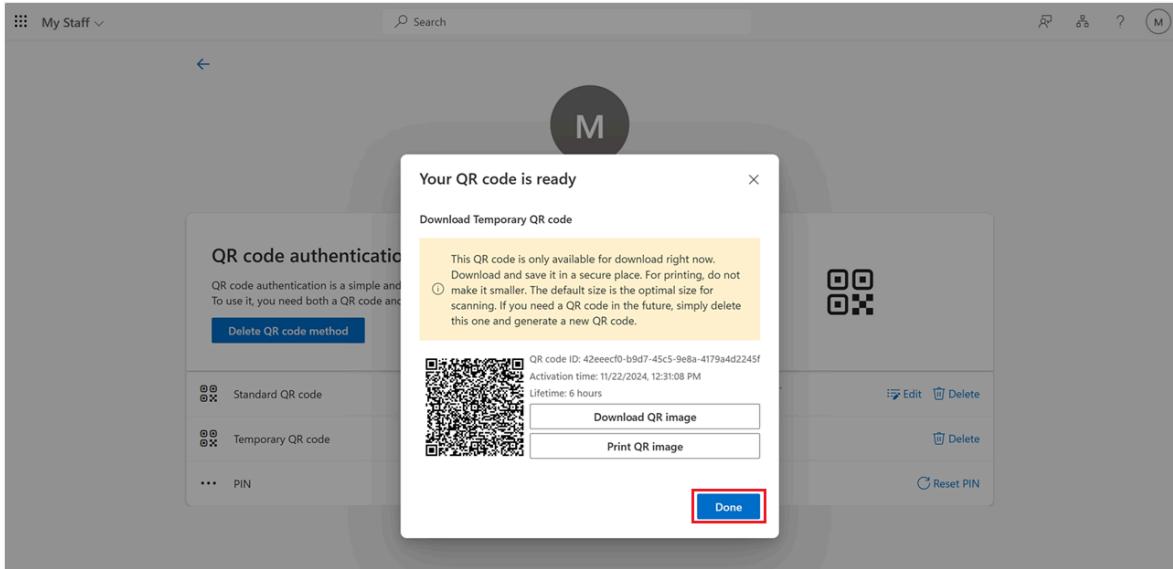
Download or print the QR code, and click **Done**.



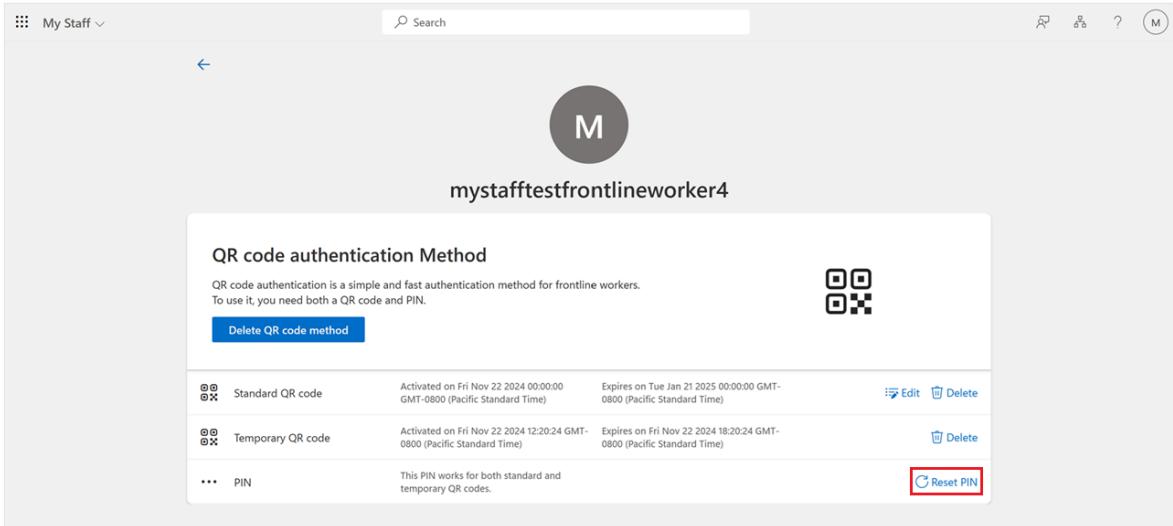
- To add a temporary QR code, click **Add new** next to the temporary QR code. Specify the **Lifetime in hours** and the **Activation date**, and click **Add**.



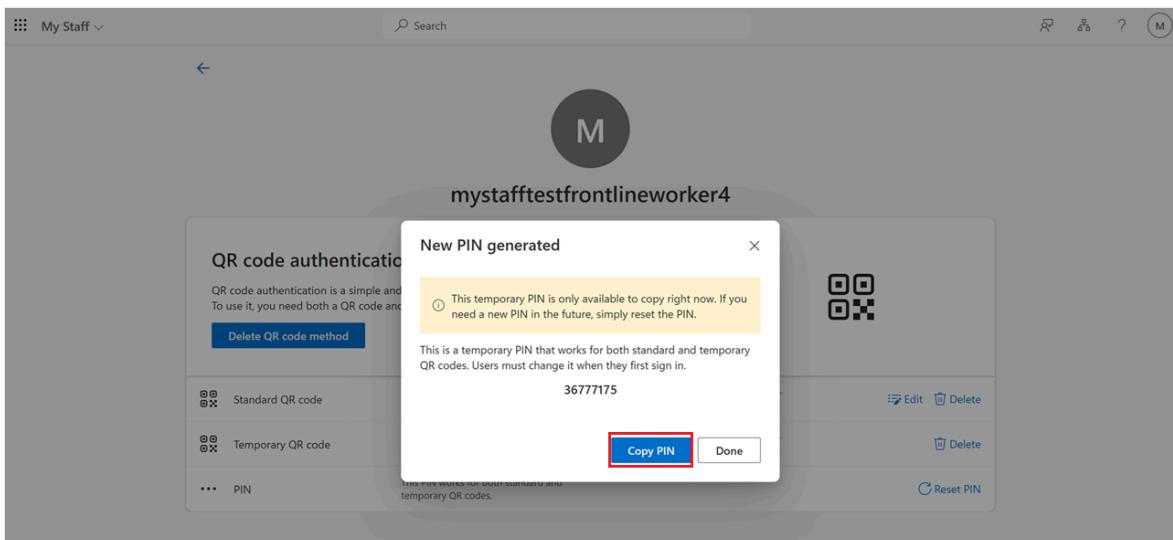
Download or print the QR code, and click **Done**.



- To reset a PIN, click **Reset PIN**.



Click **Copy PIN** to copy the PIN to your clipboard.



Edit the QR code authentication method for a user in Microsoft Graph API

This example shows how to delete the standard QR code for a user if they lose their badge, and create a new standard QR code. The user isn't required to change their PIN.

Delete a standard QR code:

- Request

```
https  
  
DELETE  
https://graph.microsoft.com/beta/users/flokreg@contoso.com/authentication/qrCodePinMethod/standardQRCode`
```

- Response

```
https  
  
HTTP/1.1 204 No Content
```

Create a standard QR code:

- Request

```
https  
  
HTTP PATCH/users/{id |  
userPrincipalName}/authentication/qrCodePinMethod/standardQRCode`  
  
{  
    "startDateTime": "2024-10-30T12:00:00Z",  
    "expireDateTime": "2024-12-30T12:00:00Z"  
}
```

- Response

```
https  
  
HTTP/1.1 201 Created  
Location: /beta/users/aaaaaaaa-bbbb-cccc-1111-  
222222222222/authentication/qrCodePinMethod/standardQRCode`  
Content-type: application/json  
  
{
```

```
"id": "BBBBBBBB-1C1C-2D2D-3E3E-444444444444",
"expireDateTime": "2024-12-30T12:00:00Z",
"startDateTime": "2024-10-30T12:00:00Z",
"createdDateTime": "2024-10-30T12:00:00Z",
"lastUsedDateTime": null,
"image":
{
  "binaryValue": "<binaryImageData>",
  "version": 1,
  "errorCorrectionLevel": "H".
  "rawContent": <binary data encoded in QR>
}
}
```

Get a standard QR code:

- **Request**

```
https

GET
https://graph.microsoft.com/beta/users/{id|UPN}/authentication/qrCodePinMethod/standardQRCode`
```

- **Response**

```
https

HTTP/1.1 200 OK
Content-type: application/json

{
  "id": "BBBBBBBB-1C1C-2D2D-3E3E-444444444444",
  "image": null,
  "expireDateTime": "2024-12-30T12:00:00Z",
  "startDateTime": "2024-10-30T12:00:00Z",
  "createdDateTime": "2024-10-30T12:00:00Z",
  "lastUsedDateTime": "2024-12-30T12:00:00Z"
}
```

This example shows how to create a temporary QR code for a user. The user can use the existing PIN. This operation returns an error if a temporary QR code already exists for the user, or if the `expireDateTime` is more than 12 hours past the `startDateTime`.

- **Request**

```
https
```

```
HTTP PATCH/users/{id |  
userPrincipalName}/authentication/qrCodePinMethod/temporaryQRCode`  
  
{  
    "startDateTime": "2024-10-30T12:00:00Z",  
    "expireDateTime": "2024-10-30T22:00:00Z"  
}
```

- **Response**

```
https
```

```
HTTP/1.1 201 Created  
Location: /beta/users/aaaaaaaa-bbbb-cccc-1111-  
222222222222/authentication/qrCodePinMethod/temporaryQRCode`  
Content-type: application/json  
  
{  
    "id": "EEEEEEE-4F$F-5A5A-6B6B-777777777777"  
    "expireDateTime": "2024-10-30T22:00:00Z",  
    "startDateTime": "2024-10-30T12:00:00Z"  
    "createdDateTime": "2024-10-30T12:00:00Z",  
    "lastUsedDateTime": null,  
    "image":  
        {  
            "binaryValue": "<binaryImageData>",  
            "version": 1,  
            "errorCorrectionLevel": "H".  
            "rawContent": <binary data encoded in QR>  
        }  
}
```

Get a temporary QR code:

- **Request**

```
https
```

```
GET  
https://graph.microsoft.com/beta/users/{id|UPN}/authentication/qrCodePi  
nMethod/temporaryQRCode`
```

- **Response**

```
https
```

```
HTTP/1.1 200 OK
Content-type: application/json

{
    "id": "EEEEEEEEE-4F$F-5A5A-6B6B-777777777777",
    "image": null,
    "expireDateTime": "2024-10-30T22:00:00Z",
    "startDateTime": "2024-10-30T12:00:00Z",
    "createdDateTime": "2024-10-30T12:00:00Z",
    "lastUsedDateTime": "2024-10-30T20:00:00Z"
}
```

This example shows how to delete a temporary QR code for a user.

- **Request**

```
https

DELETE
https://graph.microsoft.com/beta/users/flokreg@contoso.com/authentication/qrCodePinMethod/temporaryQRCode`
```

- **Response**

```
https

HTTP/1.1 204 No Content
```

This example shows how to reset the PIN a QR code authentication method:

- **Request**

```
https

PATCH
https://graph.microsoft.com/beta/users/flokreg@contoso.com/authentication/qrCodePinMethod/pin`
```

- **Response**

```
https

{
    "code": <PIN>,
    "forceChangePinNextSignIn": true,
    "createdDateTime": "2024-10-30T12:00:00Z",
```

```
    "updatedDateTime": null  
}
```

This example shows how to force a user to change their PIN for a QR code authentication method:

- Request

```
https  
  
PATCH  
https://graph.microsoft.com/beta/users/flokreg@contoso.com/authentication/qrcodePinMethod/updatePin`  
  
{  
  "currentPin": "<Old PIN>",  
  "newPin": "<New PIN>"  
}
```

- Response

```
https  
  
HTTP/1.1 204 No Content
```

Delete the QR code authentication method for a user

You can delete the QR code authentication method for a user by using the Microsoft Entra admin center, My Staff, or Microsoft Graph API.

Delete the QR code authentication method for a user in the Microsoft Entra admin center

If a QR code authentication method is deleted for a user, they can no longer sign in by using that authentication method.

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Administrator](#).
2. Go to **Users**, select a user, and click **Authentication methods**.

- Under Usable authentication methods, click the ellipsis on the right side of the QR code, and click Delete.

Home > Users > Floris Kregel

Floris Kregel | Authentication methods

User

Search

Add authentication method | Reset password | Require re-register multifactor authentication | Revoke multifactor authentication sessions | ...

Overview | Audit logs | Sign-in logs | Diagnose and solve problems | Custom security attributes | Assigned roles | Administrative units | Groups | Applications | Licenses | Devices | Azure role assignments | Authentication methods | New support request

Authentication methods

Authentication method	Detail
QR code (Preview)	QR code expires 8/5/2025, 11:59:59 PM

Non-usuable authentication methods

Authentication method	Detail
No non-usuable methods.	

System preferred multifactor authentication method

Feature status	System preferred MFA method
Enabled	No system preferred MFA method

Delete the QR code authentication method for a user in My Staff

- To delete the QR code auth method itself, click Delete QR code method.

My Staff

Search

M

mystafftestfrontlineworker4

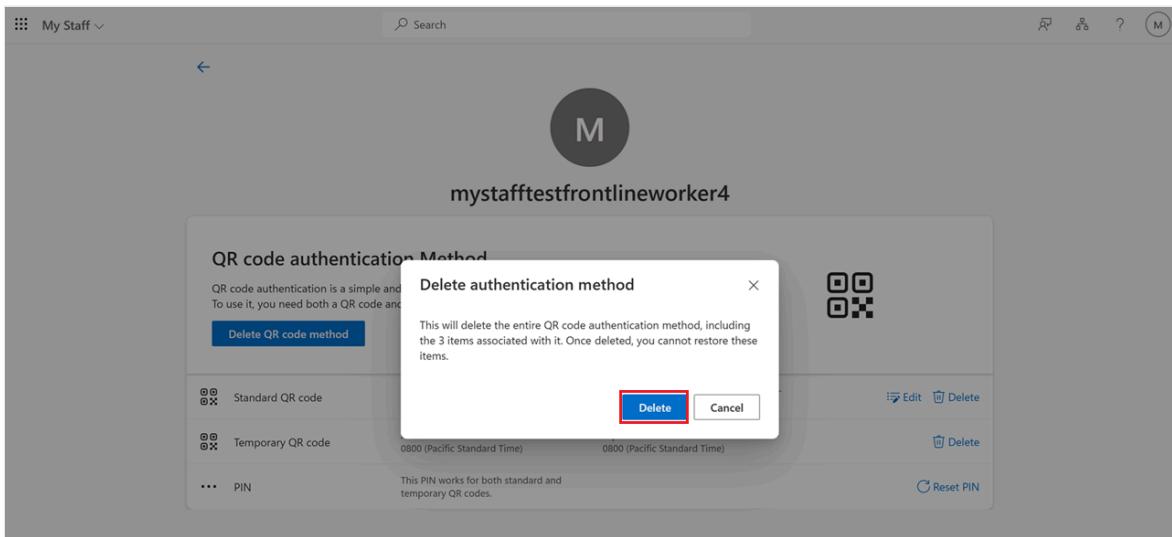
QR code authentication Method

QR code authentication is a simple and fast authentication method for frontline workers.
To use it, you need both a QR code and PIN.

Delete QR code method

Method Type	Activation Date	Expiration Date	Action
Standard QR code	Activated on Fri Nov 22 2024 00:00:00 GMT-0800 (Pacific Standard Time)	Expires on Tue Jan 21 2025 00:00:00 GMT-0800 (Pacific Standard Time)	Edit Delete
Temporary QR code	Activated on Fri Nov 22 2024 12:20:24 GMT-0800 (Pacific Standard Time)	Expires on Fri Nov 22 2024 18:20:24 GMT-0800 (Pacific Standard Time)	Delete
PIN	This PIN works for both standard and temporary QR codes.		Reset PIN

- Click Delete to confirm the action.



Delete the QR code authentication method for a user in Microsoft Graph API

This example shows how to delete a standard QR code for a user.

- Request

```
https  
  
DELETE  
https://graph.microsoft.com/beta/users/flokreg@contoso.com/authentication/qrcodePinMethod/standardQRCode`
```

- Response

```
https  
  
HTTP/1.1 204 No Content
```

Sign in to Microsoft Teams or Managed Home Screen (MHS) with QR code

Microsoft Teams and Managed Home Screen (MHS) have an optimized QR code sign-in experience. An Authentication Policy Administrator needs to configure Intune or another mobile device management (MDM) solution to enable the QR code authentication method for mobile devices.

Enable sign-in with a QR code in Teams or MHS

When configuring with Intune, assign Microsoft Authenticator as a required app for all devices you want to add QR code authentication for.

[+] Expand table

Platform	MDM app config key	Value	Configuration location
iOS	preferred_auth_config	qrpin	Device management profile, which configures a single sign-on (SSO) extension
Android	preferred_auth_config	qrpin	Microsoft Authenticator

ⓘ Note

MHS is only available on Android devices.

QR code authentication Teams sign-in experience

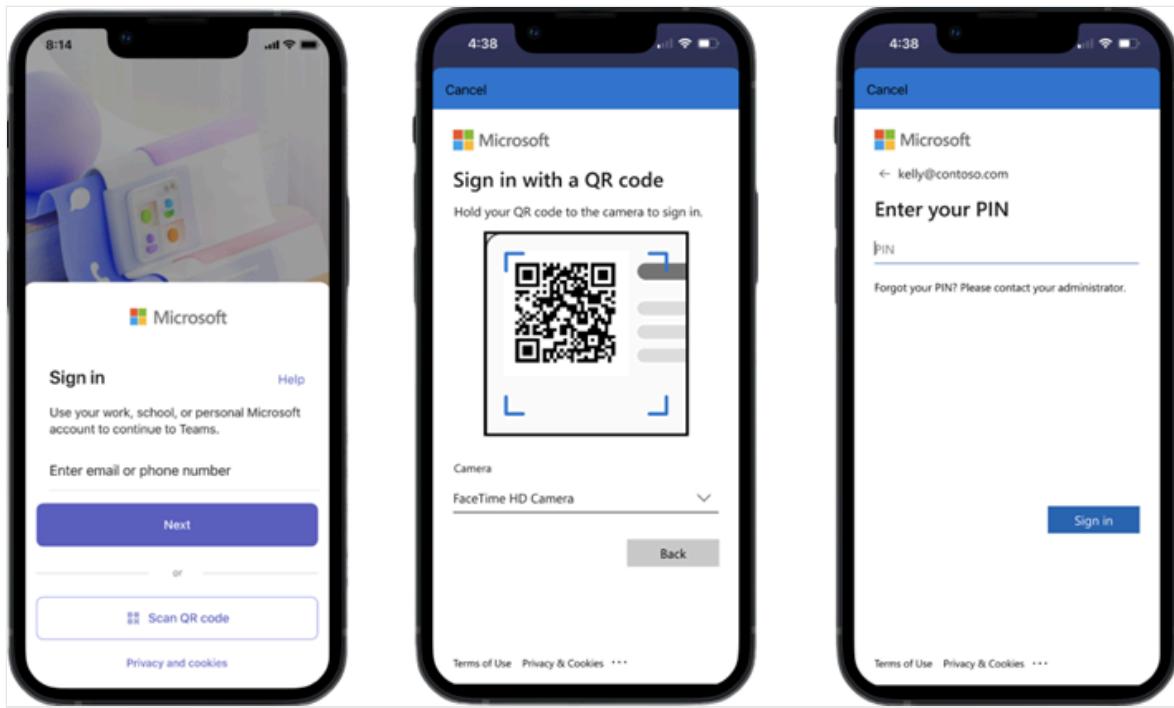
Users need to [download Teams](#). The following table lists the minimum Teams version for mobile operating systems. For more information about Teams versions, see [Version update history for the new and classic Microsoft Teams app](#).

[+] Expand table

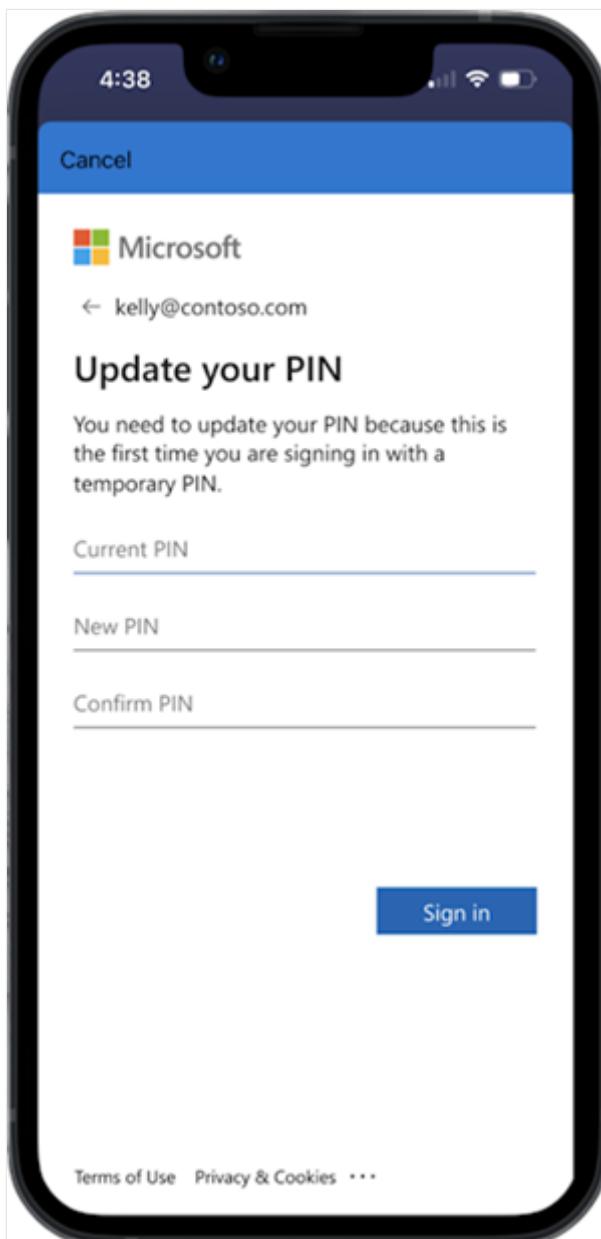
Mobile OS	Release date	Teams version
iOS and iPadOS	July 21, 2024	6.13.1 (1.0.0.77.2024132501)
Android	August 08, 2024	1416/1.0.0.2024143204 (2024143204)

Users can follow these steps to sign in with a QR code in Teams:

1. Click **Scan QR code** in Microsoft Teams.
2. Scan the QR code. Give consent if you're asked for camera permission.
3. Enter your PIN.
4. You're now signed in to the app.

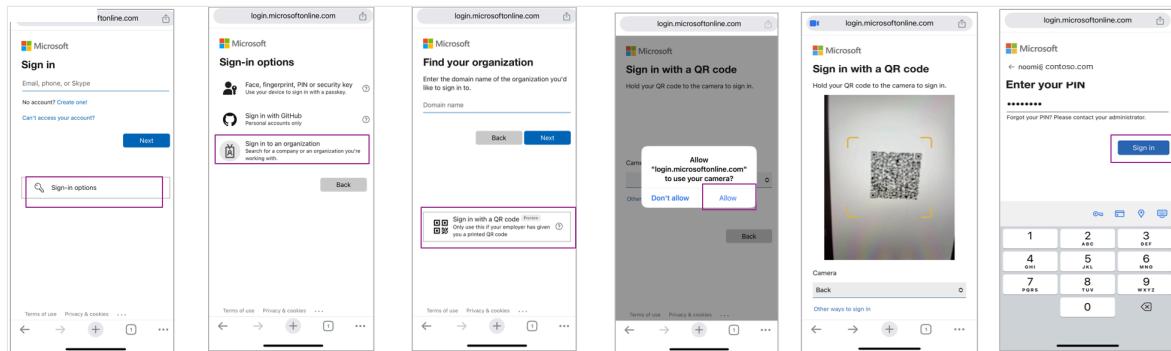


5. When you sign in with a temporary PIN, you need to change it.



QR code authentication web sign-in experience (login.microsoftonline.com)

1. Click More sign-in options > Sign in to an organization > Sign in with QR code.
2. Allow the camera when prompted > scan the QR code > enter your PIN > you're successfully signed in.



Add security with QR code authentication using Conditional Access policies

Restrict the QR code authentication method to only frontline workers, compliant, and shared devices. This section covers how to create policies that restrict QR code authentication method to only frontline workers and shared devices.

Restrict QR code authentication to frontline workers

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > QR code > Enable and target**.
3. Click **Add target** > select a group that only includes frontline workers, such as **Frontline workers** in the following screenshot. This group selection restricts enablement of the QR code authentication method only to frontline workers added to the **Frontline workers** group.

The screenshot shows the 'QR code (Preview) settings' page in the Microsoft Entra admin center. At the top, there's a breadcrumb navigation: Home > Authentication methods | Policies > QR code (Preview) settings. Below the title, a note says: 'QR code authentication is a simple and fast authentication method for frontline workers. To use it, you need both a QR code and PIN.' There are two tabs: 'Enable and Target' (selected) and 'Configure'. Under 'Enable and Target', the 'Enable' switch is turned on. The 'Include' tab is selected, showing 'All users' (radio button) is chosen over 'Select groups'. A 'Target' section allows adding groups, with 'Frontline workers' listed under 'Name' and 'Group' under 'Type'. The 'Registration' field is set to 'Optional'. A red box highlights the 'Frontline workers' group entry.

Restrict QR code authentication to shared devices

1. Sign in to the Microsoft Entra admin center [as a Conditional Access Administrator](#).
2. Click **Conditional Access > Authentication strengths > New authentication strength**.

The screenshot shows the 'Conditional Access | Authentication strengths' page. The left sidebar includes links for Overview, Policies, Insights and reporting, Diagnose and solve problems, Manage (with Named locations, Custom controls (Preview), Terms of use, VPN connectivity, Authentication contexts, and Authentication strengths selected), Classic policies, Monitoring (Sign-in logs, Audit logs), Troubleshooting + Support (New support request), and Home. The main area displays a table of authentication strengths. The table has columns: Authentication strength, Type, Authentication methods, and Conditional access policies. It lists four rows:

Authentication strength	Type	Authentication methods	Conditional access policies
Phish-resistant + TAP	Custom	Windows Hello For Business and 4 more	Not configured in any policy yet
Multifactor authentication	Built-in	Windows Hello For Business and 16 more	Not configured in any policy yet
Passwordless MFA	Built-in	Windows Hello For Business and 3 more	Not configured in any policy yet
Phishing-resistant MFA	Built-in	Windows Hello For Business and 2 more	Not configured in any policy yet

A red box highlights the 'New authentication strength' button at the top of the page.

3. Create a custom authentication strength Conditional Access policy. Select authentication **QR code (Preview)**.
4. Create a Conditional Access policy that requires shared devices be marked as compliant with policies from Intune or another MDM solution. This policy makes sure that frontline workers can access only specific resources from a compliant, shared device that they signed into with a QR code.
 - a. Under **Users or workload identities > Include** > select **Users and groups**, and choose your **Frontline workers** frontline worker group.

b. Under **Target resources** > **Include** > select specific resources that frontline workers can access.

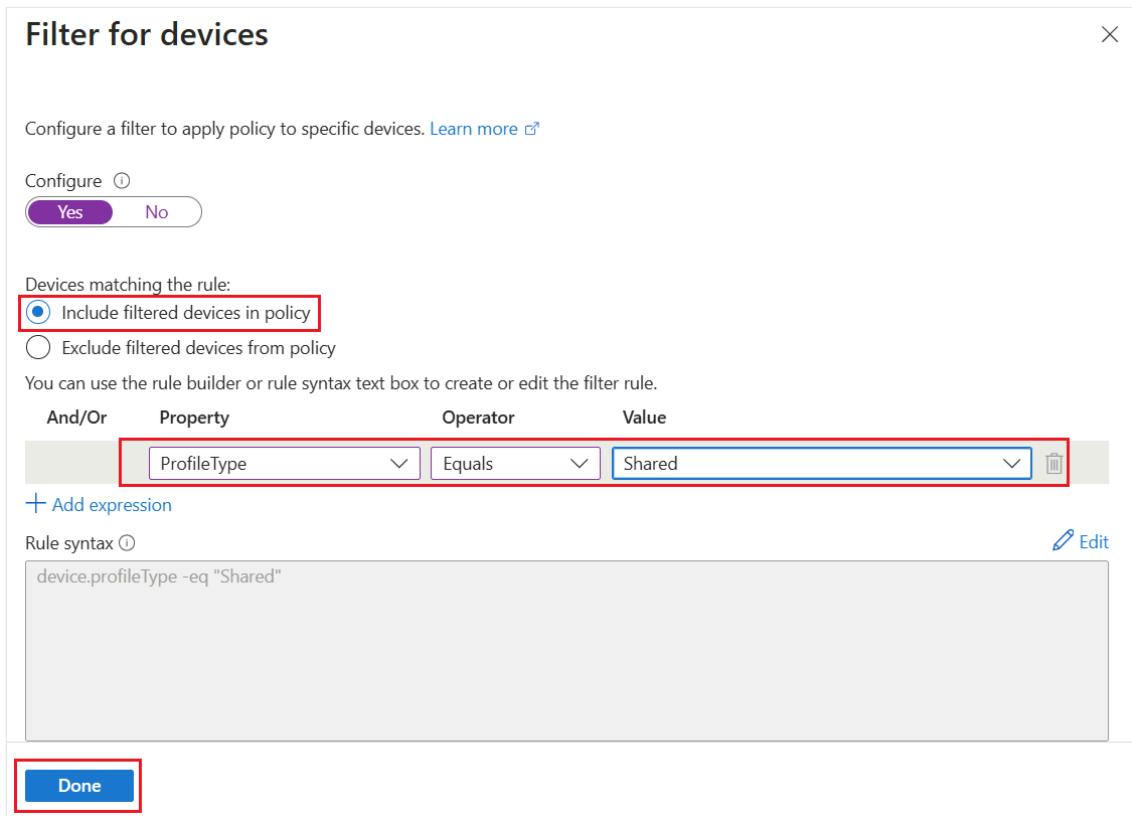
c. Under **Conditions**, click **Filter for devices**, set **Configure** to **Yes**.

d. Click **Include filtered devices from policy**.

e. For **Property**, select **ProfileType**.

f. For **Operator**, select **Equals**.

g. For **Value**, select **Shared**.



h. Under **Access controls** > **Grant** > select **Require device to be marked as compliant**, and click **Select**.

i. Click **Create**.

Related content

- [Authentication methods in Microsoft Entra ID - QR code authentication method \(Preview\)](#)
- [Manage your users with My Staff](#)
- [What authentication and verification methods are available in Microsoft Entra ID?](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Get started with phishing-resistant passwordless authentication deployment in Microsoft Entra ID

Article • 03/04/2025

Passwords are the primary attack vector for modern adversaries, and a source of friction for users and administrators. As part of an overall [Zero Trust security strategy](#), Microsoft recommends [moving to phishing-resistant passwordless](#) in your authentication solution. This guide helps you select, prepare, and deploy the right phishing-resistant passwordless credentials for your organization. Use this guide to plan and execute your phishing-resistant passwordless project.

Features like multifactor authentication (MFA) are a great way to secure your organization. But users often get frustrated with the extra security layer on top of their need to remember passwords. Phishing-resistant passwordless authentication methods are more convenient. For example, an analysis of Microsoft consumer accounts shows that sign-in with a password can take up to 9 seconds on average, but passkeys only take around 3 seconds in most cases. The speed and ease of passkey sign-in is even greater when compared with traditional password and MFA sign in. Passkey users don't need to remember their password, or wait around for SMS messages.

ⓘ Note

This data is based on analysis of Microsoft consumer account sign-ins.

Phishing-resistant passwordless methods also have extra security baked in. They automatically count as MFA by using something that the user has (a physical device or security key) and something the user knows or is, like a biometric or PIN. And unlike traditional MFA, phishing-resistant passwordless methods deflect phishing attacks against your users by using hardware-backed credentials that can't be easily compromised.

Microsoft Entra ID offers the following phishing-resistant passwordless authentication options:

- Passkeys (FIDO2)
 - Windows Hello for Business
 - Platform credential for macOS (preview)
 - Microsoft Authenticator app passkeys

- FIDO2 security keys
- Other passkeys and providers, such as iCloud Keychain - [on roadmap ↗](#)
- Certificate-based authentication/smart cards

Prerequisites

Before you start your Microsoft Entra phishing-resistant passwordless deployment project, complete these prerequisites:

- Review license requirements
- Review the roles needed to perform privileged actions
- Identify stakeholder teams that need to collaborate

License requirements

Registration and passwordless sign in with Microsoft Entra doesn't require a license, but we recommend at least a Microsoft Entra ID P1 license for the full set of capabilities associated with a passwordless deployment. For example, a Microsoft Entra ID P1 license helps you enforce passwordless sign in through Conditional Access, and track deployment with an authentication method activity report. Refer to the licensing requirements guidance for features referenced in this guide for specific licensing requirements.

Integrate apps with Microsoft Entra ID

Microsoft Entra ID is a cloud-based Identity and Access Management (IAM) service that integrates with many types of applications, including Software-as-a-Service (SaaS) apps, line-of-business (LOB) apps, on-premises apps, and more. You need to integrate your applications with Microsoft Entra ID to get the most benefit from your investment in passwordless and phishing-resistant authentication. As you integrate more apps with Microsoft Entra ID, you can protect more of your environment with Conditional Access policies that enforce the use of phishing-resistant authentication methods. To learn more about how to integrate apps with Microsoft Entra ID, see [Five steps to integrate your apps with Microsoft Entra ID](#).

When you develop your own applications, follow the developer guidance for supporting passwordless and phishing-resistant authentication. For more information, see [Support passwordless authentication with FIDO2 keys in apps you develop](#).

Required roles

The following table lists least privileged role requirements for phishing-resistant passwordless deployment. We recommend that you enable phishing-resistant passwordless authentication for all privileged accounts.

[\[+\] Expand table](#)

Microsoft Entra role	Description
User Administrator	To implement combined registration experience
Authentication Administrator	To implement and manage authentication methods
Authentication Policy Administrator	To implement and manage the Authentication methods policy
User	To configure Authenticator app on device; to enroll security key device for web or Windows 10/11 sign-in

Customer stakeholder teams

To ensure success, make sure that you engage with the right stakeholders, and that they understand their roles before you begin your planning and rollout. The following table lists commonly recommended stakeholder teams.

[\[+\] Expand table](#)

Stakeholder team	Description
Identity and Access Management (IAM)	Manages day-to-day operations of the IAM system
Information Security Architecture	Plans and designs the organization's information security practices
Information Security Operations	Runs and monitors information security practices for Information Security Architecture
Security Assurance and Audit	Helps ensure IT processes are secure and compliant. They conduct regular audits, assess risks, and recommend security measures to mitigate identified vulnerabilities and enhance the overall security posture.
Help Desk and Support	Assists end users who encounter issues during deployments of new technologies and policies, or when issues occur
End-User Communications	Messages changes to end users in preparation to aid in driving user-facing technology rollouts

Next steps

[Deploy a phishing-resistant passwordless authentication deployment in Microsoft Entra ID](#)

[Considerations for specific personas in a phishing-resistant passwordless authentication deployment in Microsoft Entra ID](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) ↗

Plan a phishing-resistant passwordless authentication deployment in Microsoft Entra ID

Article • 03/04/2025

When you deploy and operationalize phishing-resistant passwordless authentication in your environment, we recommend a user persona-based approach. Different phishing-resistant passwordless methods are more effective than others for certain user personas. This deployment guide helps you see which types of methods and rollout plans make sense for user personas in your environment. The phishing-resistant passwordless deployment approach commonly has 6 steps, which roughly flow in order, but don't have to be 100% completed before moving on to other steps:

Determine your user personas

Determine the user personas relevant for your organization. This step is critical to your project because different personas have different needs. Microsoft recommends you consider and evaluate at least 4 generic user personas in your organization.

[+] Expand table

User persona	Description
Information workers	<ul style="list-style-type: none">Examples include office productivity staff, such as in marketing, finance, or human resources.Other types of information workers may be executives and other high-sensitivity workers who need special controlsTypically have a 1:1 relationship with their mobile and computing devicesMay bring their own devices (BYOD), especially for mobile
Frontline workers	<ul style="list-style-type: none">Examples include retail store workers, factory workers, manufacturing workersTypically work only on shared devices or kiosksMay not be allowed to carry mobile phones
IT Pros/DevOps workers	<ul style="list-style-type: none">Examples include IT admins for on-premises Active Directory, Microsoft Entra ID, or other privileged accounts. Other examples would be DevOps workers or DevSecOps workers who manage and deploy automations.Typically have multiple user accounts, including a "normal" user account, and one or more administrative accountsCommonly use remote access protocols, such as Remote Desktop Protocol (RDP) and Secure Shell Protocol (SSH), to administer remote systems

User persona	Description
	<ul style="list-style-type: none"> • May work on locked down devices with Bluetooth disabled • May use secondary accounts to run non-interactive automations and scripts
Highly regulated workers	<ul style="list-style-type: none"> • Examples include US federal government workers subject to Executive Order 14028 requirements, state and local government workers, or workers subject to specific security regulations • Typically have a 1:1 relationship with their devices, but have specific regulatory controls that must be met on those devices and for authentication • Mobile phones may not be allowed in secure areas • May access air-gapped environments without internet connectivity • May work on locked down devices with Bluetooth disabled

Microsoft recommends that you broadly deploy phishing-resistant passwordless authentication across your organization. Traditionally, information workers are the easiest user persona to begin with. Don't delay rollout of secure credentials for information workers while you resolve issues that affect IT Pros. Take the approach of "*don't let perfect be the enemy of good*" and deploy secure credentials as much as possible. As more users sign in using phishing-resistant passwordless credentials, you reduce the attack surface of your environment.

Microsoft recommends that you define your personas, and then place each persona into a Microsoft Entra ID group specifically for that user persona. These groups are used in later steps to [roll out credentials](#) to different types of users, and when you begin to [enforce the use of phishing-resistant passwordless credentials](#).

Plan device readiness

Devices are an essential aspect of any successful phishing-resistant passwordless deployment, since one of the goals of phishing-resistant passwordless credentials is to protect credentials with the hardware of modern devices. First, become familiar with [FIDO2 supportability for Microsoft Entra ID](#).

Ensure that your devices are prepared for phishing-resistant passwordless by patching to the latest supported versions of each operating system. Microsoft recommends your devices are running these versions at a minimum:

- Windows 10 22H2 (for Windows Hello for Business)
- Windows 11 22H2 (for the best user experience when using passkeys)
- macOS 13 Ventura
- iOS 17
- Android 14

These versions provide the best support for natively integrated features like passkeys, Windows Hello for Business, and macOS Platform Credential. Older operating systems may require external authenticators, like FIDO2 security keys, to support phishing-resistant passwordless authentication.

Register users for phishing-resistant credentials

Credential registration and bootstrapping are the first major end-user facing activities in your phishing-resistant passwordless deployment project. This section covers the rollout of **portable** and **local** credentials.

[+] Expand table

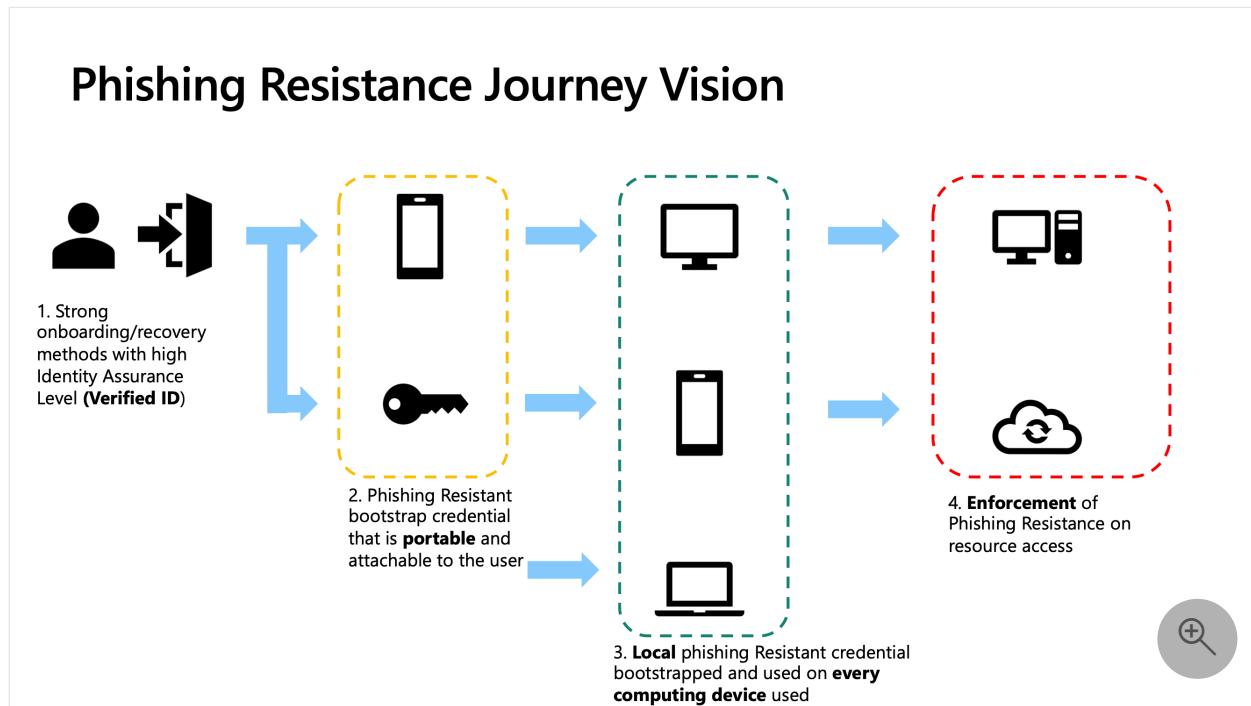
Credentials	Description	Benefits
Portable	Can be used across devices . You can use portable credentials to sign in to another device, or to register credentials on other devices.	The most important type of credential to register for most users, as they can be used across devices, and provide phishing-resistant authentication in many scenarios.
Local	You can use local credentials to authenticate on a device without needing to rely on external hardware, such as using Windows Hello for Business biometric recognition to sign into an app in Microsoft Edge browser on the same PC	They have two main benefits over the portable credentials: <ul style="list-style-type: none">• They provide redundancy. If users lose their portable device, forget it at home, or have other issues, then the local credential provides them with a backup method to continue to work on their computing device.• They provide a great user experience. With a local credential, users don't need to pull phones out of their pocket, scan QR codes, or perform other tasks that slow down authentication and add friction. Locally available phishing-resistant credentials help users sign in more easily on the devices they regularly use.

- For *new users*, the registration and bootstrapping process takes a user with no existing enterprise credentials, and verifies their identity. It bootstraps them into their first portable credential, and uses that portable credential to bootstrap other local credentials on each of their computing devices. After registration, the admin may enforce phishing-resistant authentication for users in Microsoft Entra ID.
- For *existing users*, this phase gets users to register for phishing-resistant passwordless on their existing devices directly, or using existing MFA credentials to bootstrap phishing-resistant passwordless credentials. The end goal is the same as

new users - most users should have at least one **portable** credential, and then **local** credentials on each computing device. If you're an admin who deploys phishing-resistant passwordless for existing users, then you may be able to skip ahead to the [Onboarding Step 2: Bootstrapping a Portable Credential section](#).

Before you start, Microsoft recommends enabling passkey and other credentials for enterprise users in the tenant. If users are motivated to self-register for strong credentials, it's beneficial to allow it. At a minimum, you should enable the [Passkey \(FIDO2\) policy](#) so that users can register for passkeys and security keys if they prefer them.

This section focuses on phases 1-3:



Users should have at least two authentication methods registered. With another method registered, the user has a backup method available if something happens to their primary method, like when a device is lost or stolen. For example, it's a good practice for users to have passkeys registered both on their phone, and locally on their workstation in Windows Hello for Business.

ⓘ Note

It is always recommended that users have at least two authentication methods registered. This ensures the user has a backup method available if something happens to their primary method, such as in cases of device loss or theft. For example, it is a good practice for users to have passkeys registered both on their phone and locally on their workstation in Windows Hello for Business.

Note

This guidance is tailored for currently existing support for passkeys in Microsoft Entra ID, which includes device-bound passkeys in Microsoft Authenticator and device-bound passkeys on physical security keys. Microsoft Entra ID plans to add support for synced passkeys. For more information, see [Public preview: Expanding passkey support in Microsoft Entra ID](#). This guide will be updated to include synced passkey guidance once available. For example, many organizations may benefit from relying on sync for phase 3 in the preceding diagram rather than bootstrapping entirely new credentials.

Onboarding step 1: Identity verification

For remote users who haven't proven their identity, enterprise onboarding is a significant challenge. Without proper identity verification, an organization cannot be completely certain that they are onboarding the person that they intend to. Microsoft Entra Verified ID can provide high assurance identity verification. Organizations can work with an identity verification partner (IDV) to verify the identities of new remote users in the onboarding process. After processing a user's government-issued ID, the IDV can provide a Verified ID that affirms the user's identity. The new user presents this identity-affirming Verified ID to the hiring organization to establish trust and confirm that the organization is onboarding the right person. Organizations can add Face Check with Microsoft Entra Verified ID which adds a facial matching layer to the verification, ensuring that the trusted user is presenting the identity-affirming Verified ID in that moment.

After verifying their identity through the proofing process, new hires are given a Temporary Access Pass (TAP) that they can use to bootstrap their first portable credential.

Refer to the following guides to enable Microsoft Entra Verified ID onboarding and TAP issuance:

- [Onboard new remote employees using ID verification](#)
- [Using Face Check with Microsoft Entra Verified ID to unlock high assurance verifications at scale](#)
- [Enable the Temporary Access Pass policy](#)

Refer to the following links for licensing details for Microsoft Entra Verified ID:

- [Face Check with Microsoft Entra Verified ID pricing](#)

- Microsoft Entra Plans and Pricing ↗

Some organizations might choose other methods than Microsoft Entra Verified ID to onboard users and issue them their first credential. Microsoft recommends those organizations still use TAPs, or another way that lets a user onboard without a password. For example, you can [provision FIDO2 security keys using Microsoft Graph API](#).

Onboarding step 2: Bootstrap a portable credential

To bootstrap existing users to phishing-resistant passwordless credentials, first determine if your users are registered for traditional MFA already. Users with traditional MFA methods registered can be targeted with phishing-resistant passwordless registration policies. They can use their traditional MFA to register for their first portable phishing-resistant credential, and then move on to register for local credentials as needed.

For new users or users without MFA, go through a process to issue users a Temporary Access Pass (TAP). You can issue a TAP the same way you give new users their first credential, or by using Microsoft Entra Verified ID integrations. Once users have a TAP, they're ready to bootstrap their first phishing-resistant credential.

It's important for the user's first passwordless credential to be a portable credential that can be used to authenticate on other computing devices. For example, passkeys can be used to authenticate locally on an iOS phone, but they can also be used to authenticate on a Windows PC using a cross-device authentication flow. This cross-device capability allows that portable passkey to be used to bootstrap Windows Hello for Business on the Windows PC.

It's also important that each device the user regularly works on has a locally available credential to give the user the smoothest user experience possible. Locally available credentials reduce the time needed to authenticate because users don't need to use multiple devices, and there are fewer steps. Using the TAP from Step 1 to register a portable credential that can bootstrap these other credentials enables the user to use phishing-resistant credentials natively across the many devices they may possess.

The following table lists recommendations for different personas:

[+] [Expand table](#)

User Persona	Recommended portable credential	Alternative portable credentials
Information worker	Passkey (Authenticator app)	Security key, smart card

User Persona	Recommended portable credential	Alternative portable credentials
Frontline worker	Security key	Passkey (Authenticator app), smart card
IT pro/DevOps worker	Passkey (Authenticator app)	Security key, smart card
Highly regulated worker	Certificate (smart card)	Passkey (Authenticator app), security key

Use the following guidance to enable recommended and alternative portable credentials for the relevant user personas for your organization:

[+] Expand table

Method	Guidance
Passkeys	<ul style="list-style-type: none"> Microsoft recommends that users sign in to Microsoft Authenticator directly to bootstrap a passkey in the app. Users can use their TAP to sign into Microsoft Authenticator directly on their iOS or Android device. Passkeys are disabled by default in Microsoft Entra ID. You can enable passkeys in Authentication methods policy. Register passkeys in Authenticator on Android or iOS devices.
Security keys	<ul style="list-style-type: none"> Security keys are turned off by default in Microsoft Entra ID. You can enable FIDO2 security keys in the Authentication methods policy. Consider registering keys on behalf of your users with the Microsoft Entra ID provisioning APIs. For more information, see Provision FIDO2 security keys using Microsoft Graph API.
Smart card/certificate-based authentication (CBA)	<ul style="list-style-type: none"> Certificate-based authentication is more complicated to configure than passkeys or other methods. Consider only using it if necessary. How to configure Microsoft Entra certificate-based authentication. Make sure to configure your on-premises PKI and Microsoft Entra ID CBA policies so that users truly complete multifactor authentication to sign in. The configuration generally requires the smart card Policy Object Identifier (OID) and the necessary affinity binding settings. For more advanced CBA configurations, see Understanding the authentication binding policy.

Onboarding step 3: Bootstrap local credentials on computing devices

After users have registered a portable credential, they're ready to bootstrap other credentials on each computing device they regularly use in a 1:1 relationship, which

benefits their day-to-day user experience. This type of credential is common for information workers and IT pros, but not for frontline workers who share devices. Users who only share devices should only use portable credentials.

Your organization needs to determine which type of credential is preferred for each user persona at this stage. Microsoft recommends:

[\[+\] Expand table](#)

User persona	Recommended local credential - Windows	Recommended local credential - macOS	Recommended local credential - iOS	Recommended local credential - Android	Recommended Local Credential - Linux
Information worker	Windows Hello for Business	Platform Single Sign-on (SSO) Secure Enclave Key	Passkey (Authenticator app)	Passkey (Authenticator app)	N/A (use portable credential instead)
Frontline worker	N/A (use portable credential instead)	N/A (use portable credential instead)	N/A (use portable credential instead)	N/A (use portable credential instead)	N/A (use portable credential instead)
IT pro/DevOps worker	Windows Hello for Business	Platform SSO Secure Enclave Key	Passkey (Authenticator app)	Passkey (Authenticator app)	N/A (use portable credential instead)
Highly Regulated worker	Windows Hello for Business or CBA	Platform SSO Secure Enclave Key or CBA	Passkey (Authenticator app) or CBA	Passkey (Authenticator app) or CBA	N/A (use smart card instead)

Use the following guidance to enable the recommended local credentials in your environment for the relevant user personas for your organization:

[\[+\] Expand table](#)

Method	Guidance
Windows Hello for Business	<ul style="list-style-type: none"> Microsoft recommends using the Cloud Kerberos Trust method to deploy Windows Hello for Business. For more information, see the Cloud Kerberos trust deployment guide. The Cloud Kerberos Trust method applies to any environment where users are synced from on-premises Active Directory to Microsoft Entra ID. It helps synced users on PCs that are either Microsoft Entra joined or Microsoft Entra hybrid joined. Windows Hello for Business should only be used when each user on a PC is signing into that PC as themselves. It shouldn't be used on kiosk devices that use

Method	Guidance
	<p>a shared user account.</p> <ul style="list-style-type: none"> Windows Hello for Business supports up to 10 users per device. If your shared devices need to support more users, then use a portable credential instead, such as security keys. Biometrics are optional, but recommended. For more information, see Prepare users to provision and use Windows Hello for Business.
Platform SSO Secure Enclave Key	<ul style="list-style-type: none"> Platform SSO supports 3 different user authentication methods (Secure Enclave key, smart card, and password). Deploy the Secure Enclave key method to mirror your Windows Hello for Business on your Macs. Platform SSO requires that Macs are enrolled in Mobile Device Management (MDM). For specific instructions for Intune, see Configure Platform SSO for macOS devices in Microsoft Intune. Refer to your MDM vendor's documentation if you use another MDM service on your Macs.
Passkeys	<ul style="list-style-type: none"> Microsoft recommends the same device registration option to bootstrap passkeys in Microsoft Authenticator (rather than the cross-device registration option). Users use their TAP to sign into Microsoft Authenticator directly on their iOS or Android device. Passkeys are disabled by default in Microsoft Entra ID, enable them in the Authentication methods policy. For more information, see Enable passkeys in Microsoft Authenticator. Register passkeys in Authenticator on Android or iOS devices.

Persona-specific considerations

Each persona has its own challenges and considerations that commonly come up during phishing-resistant passwordless deployments. As you identify which personas you need to accommodate, you should factor these considerations into your deployment project planning. The following links have specific guidance for each persona:

- [Information workers](#)
- [Frontline workers](#)
- [IT pros/DevOps workers](#)
- [Highly regulated workers](#)

Drive usage of phishing-resistant credentials

This step covers how to make it easier for users to adopt phishing-resistant credentials. You should test your deployment strategy, plan for the rollout, and communicate the

plan to end users. Then you can create reports and monitor progress before you enforce phishing-resistant credentials across your organization.

Test deployment strategy

Microsoft recommends that you test the deployment strategy created in the previous step with a set of test and pilot users. This phase should include the following steps:

- Create a list of test users and early adopters. These users should represent your different user personas, and not just IT Admins.
- Create a Microsoft Entra ID group, and add your test users to the group.
- Enable your [Authentication methods policies](#) in Microsoft Entra ID, and scope the test group to the methods that you enable.
- Measure the registration rollout for your pilot users by using the [Authentication Methods Activity](#) report.
- Create Conditional Access policies to enforce the use of phishing-resistant passwordless credentials on each operating system type, and target your pilot group.
- Measure the success of the enforcement using [Azure Monitor and Workbooks](#).
- Gather feedback from users on the success of the rollout.

Plan rollout strategy

Microsoft recommends driving usage based on which user personas are most ready for deployment. Typically, this means deploying for users in this order, but this may change depending on your organization:

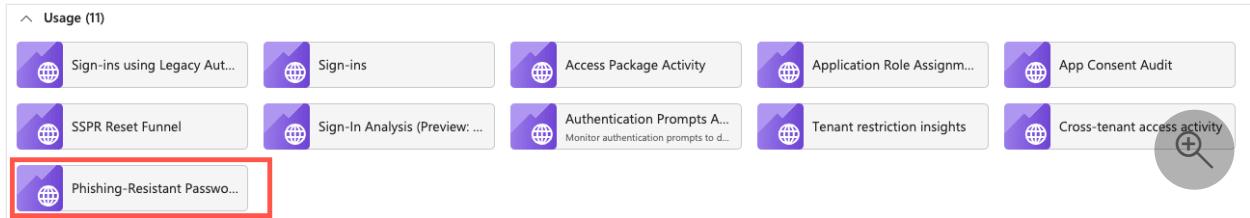
1. Information workers
2. Frontline workers
3. IT pros/DevOps workers
4. Highly regulated workers

Use the following sections to create end user communications for each persona group, scope and rollout the passkeys registration feature, and user reporting and monitoring to track rollout progress.

Driving readiness with the Phishing-Resistant Passwordless Workbook (Preview)

Organizations may optionally choose to export their Microsoft Entra ID sign-in logs to [Azure Monitor](#) for long-term retention, threat hunting, and other purposes. Microsoft

has released a [workbook](#) that organizations with logs in Azure Monitor may use to help with various phases of a phishing-resistant passwordless deployment. The Phishing-Resistant Passwordless Workbook can be accessed here: aka.ms/PasswordlessWorkbook. Choose the workbook titled ***Phishing-Resistant Passwordless Deployment (Preview)***:

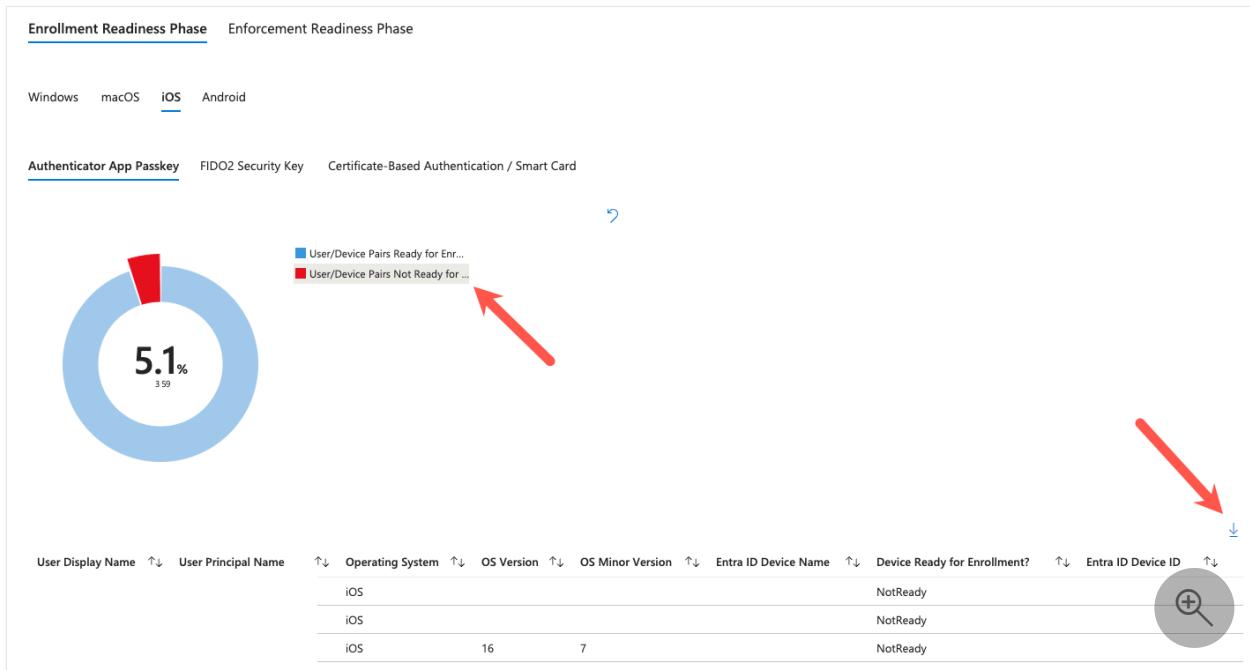


The workbook has two primary sections:

1. Enrollment Readiness Phase
2. Enforcement Readiness Phase

Enrollment readiness phase

Use the Enrollment Readiness Phase tab to analyze sign-in logs in your tenant, determining which users are ready for registration and which users may be blocked from registration. For example, with the Enrollment Readiness Phase tab you can select iOS as the OS platform and then Authenticator App Passkey as the type of credential you would like to assess your readiness for. You can then click on the workbook visualizations to filter down to users who are expected to have registration issues and export the list:



The Enrollment Readiness Phase tab of the workbook can help you evaluate readiness for the following OSes and credentials:

- Windows
 - Windows Hello for Business
 - FIDO2 Security Key
 - Authenticator App Passkey
 - Certificate-Based Authentication / Smart Card
- macOS
 - Platform SSO Secure Enclave Key
 - FIDO2 Security Key
 - Authenticator App Passkey
 - Certificate-Based Authentication / Smart Card
- iOS
 - FIDO2 Security Key
 - Authenticator App Passkey
 - Certificate-Based Authentication / Smart Card
- Android
 - FIDO2 Security Key
 - Authenticator App Passkey
 - Certificate-Based Authentication / Smart Card

Use each exported list to triage users who may have registration issues. Responses to registration issues should include assisting users in upgrading device OS versions, replacing aging devices, and choosing alternative credentials where the preferred option is not viable. For example, your organization may choose to provide physical FIDO2 security keys to Android 13 users who cannot use Passkeys in the Microsoft Authenticator app.

Similarly, use the enrollment readiness report to assist you in building out lists of users who are ready to begin enrollment communications and campaigns, in alignment with your overall [rollout strategy](#).

Enforcement readiness phase

The first step of the enforcement readiness phase is creating a Conditional Access policy in Report-Only mode. This policy will populate your sign-in logs with data regarding whether or not access would have been blocked if you were to put users/devices in scope for phishing-resistant enforcement. Create a new Conditional Access policy in your tenant with these settings:

[Expand table](#)

Setting	Value
User/Group Assignment	All users, excluding break glass accounts
App Assignment	All resources
Grant Controls	Require Authentication Strength - Phishing-resistant MFA
Enable policy	Report-only

Create this policy as early as possible in your rollout, preferably before even beginning your enrollment campaigns. This will ensure that you have a good historical dataset of which users and sign-ins would have been blocked by the policy if it was enforced.

Next, use the workbook to analyze where user/device pairs are ready for enforcement. Download lists of users who are ready for enforcement and add them to groups created in alignment with your [enforcement policies](#). Begin by selecting the read-only Conditional Access policy in the policy filter:

The screenshot shows the 'Phishing-Resistant Passwordless Workbook (Preview)' interface. At the top, there's a toolbar with icons for Workbooks, Edit, Refresh, Help, and Auto refresh: Off. Below the toolbar, the title 'Phishing-Resistant Passwordless Workbook (Preview)' is displayed. Underneath the title, there's a 'Explanation Text' dropdown set to 'Hide Explanation Text'. Further down, there are filters for 'Workspace: Select - All' and 'Subscription: Select - All'. A 'Time Range' dropdown is set to 'Last 30 days'. On the left, there are two buttons: 'Enrollment Readiness Phase' and 'Enforcement Readiness Phase', with 'Enforcement Readiness Phase' being highlighted by a red box and a red arrow pointing to it. Below these buttons is a dropdown menu showing 'Phishing-Resistant Pass... : RO - Require PR MFA for all users all apps'. At the bottom, there are links for 'Users Ready for Enforcement' and 'Further Data Analysis', along with a search icon.

The report will provide you with a list of users who would have been able to successfully pass the phishing-resistant passwordless requirement on each device platform. Download each list and put the appropriate users in enforcement group that aligns to the device platform.

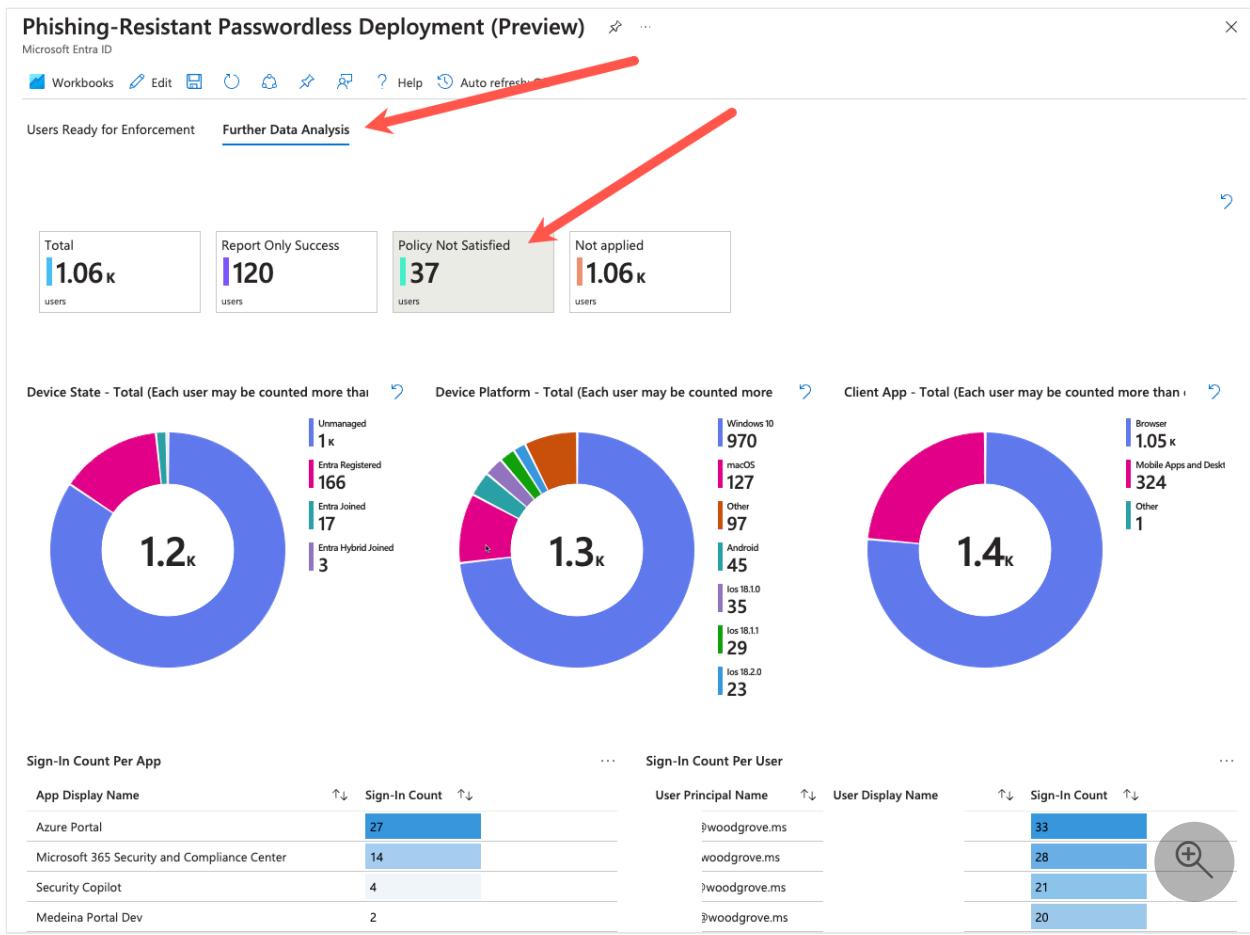
Users Ready for Enforcement Further Data Analysis

Windows Users Ready for Enforcement	macOS Users Ready for Enforcement	iOS Users Ready for Enforcement	Android Users Ready for Enforcement
Windows	macOS	iOS	Android
UserPrincipalName ↑↓	UserPrincipalName ↑↓	UserPrincipalName ↑↓	UserPrincipalName ↑↓
@woodgrove.ms @woodgrove.ms @woodgrove.ms woodgrove.ms \\woodgrove.ms \\woodgrove.ms \\woodgrove.ms @woodgrove.... @woodgrove.ms @woodgrove.... @woodgrove.ms @woodgrove.ms	>woodgrove.ms >woodgrove.ms \\woodgrove.ms woodgrove.ms @woodgrove.... @woodgrove.... \\woodgrove.ms @woodgrove.... @woodgrove.ms @woodgrove....	@woodgrove.ms @woodgrove.ms	The query returned no results.

Repeat this process over time, until you reach the point where each enforcement group contains most or all users. Eventually, you should be able to enable the report-only policy to provide enforcement for all users and device platforms in the tenant. Once you have reached this completed state you can remove the individual enforcement policies for each device OS, reducing the number of Conditional Access policies needed.

Investigating users not ready for enforcement

Use the **Further Data Analysis** tab to investigate why certain users are not ready for enforcement on various platforms. Select the **Policy Not Satisfied** box to filter the data down to user sign-ins that would have been blocked by the report-only Conditional Access policy.



Use the data provided by this report to determine which users would have been blocked, which device OSes they were on, what type of client apps they were using, and what resources they were trying to access. This data should help you target those users for various remediation or enrollment actions, so that they can be effectively moved into scope for enforcement.

Plan end user communications

Microsoft provides communication templates for end users. The [authentication rollout material](#) includes customizable email templates to inform users about phishing-resistant passwordless authentication deployment. Use the following templates to communicate to your users so they understand the phishing-resistant passwordless deployment:

- [Passkeys for Helpdesk](#)
- [Passkeys coming soon](#)
- [Register for Authenticator App Passkey](#)
- [Reminder to register for Authenticator App Passkey](#)

Communications should be repeated multiple times to help catch as many users as possible. For example, your organization may choose to communicate the different phases and timelines with a pattern like this:

1. 60 days out from enforcement: message the value of phishing-resistant authentication methods and encourage users to proactively enroll
2. 45 days out from enforcement: repeat message
3. 30 days out from enforcement: message that in 30 days phishing-resistant enforcement will begin, encourage users to proactively enroll
4. 15 days out from enforcement: repeat message, inform them of how to contact the help desk
5. 7 days out from enforcement: repeat message, inform them of how to contact the help desk
6. 1 day out from enforcement: inform them enforcement will occur in 24 hours, inform them of how to contact the help desk

Microsoft recommends communicating to users through other channels beyond just email. Other options may include Microsoft Teams messages, break room posters, and champion programs where select employees are trained to advocate for the program to their peers.

Reporting and monitoring

Use the previously covered [Phishing-Resistant Passwordless Workbook](#) to assist with monitoring and reporting on your rollout. Additionally use the reports discussed below, or rely on them if you cannot use the Phishing-Resistant Passwordless Workbook.

Microsoft Entra ID reports (such as [Authentication Methods Activity](#) and [Sign-in event details for Microsoft Entra multifactor authentication](#)) provide technical and business insights that can help you measure and drive adoption.

From the Authentication methods activity dashboard, you can view registration and usage.

- **Registration** shows the number of users capable of phishing-resistant passwordless authentication, and other authentication methods. You can see graphs that show which authentication methods users registered, and recent registration for each method.
- **Usage** shows which authentication methods were used for sign-in.

Business and technical application owners should own and receive reports based on organization requirements.

- Track phishing-resistant passwordless credentials rollout with Authentication Methods registration activity reports.
- Track user adoption of phishing-resistant passwordless credentials with Authentication Methods sign in activity reports and sign in logs.

- Use the [sign-in activity report](#) to track the authentication methods used to sign in to the various applications. Select the user row; select **Authentication Details** to view authentication method and its corresponding sign-in activity.

Microsoft Entra ID adds entries to audit logs when these conditions occur:

- An administrator changes authentication methods.
- A user makes any kind of change to their credentials within Microsoft Entra ID.

Microsoft Entra ID retains most auditing data for 30 days. We recommend longer retention for auditing, trend analysis, and other business needs.

Access auditing data in the Microsoft Entra admin center or API and download into your analysis systems. If you require longer retention, export and consume logs in a Security Information and Event Management (SIEM) tool, such as Microsoft Sentinel, Splunk, or Sumo Logic.

Monitor help desk ticket volume

Your IT help desk can provide an invaluable signal on how well your deployment is progressing, so Microsoft recommends tracking your help desk ticket volume when executing a phishing-resistant passwordless deployment.

As your help desk ticket volume increases you should slow down the pace of your deployments, user communications, and enforcement actions. As the ticket volume decreases you can ramp these activities back up. Using this approach requires that you maintain flexibility in your rollout plan.

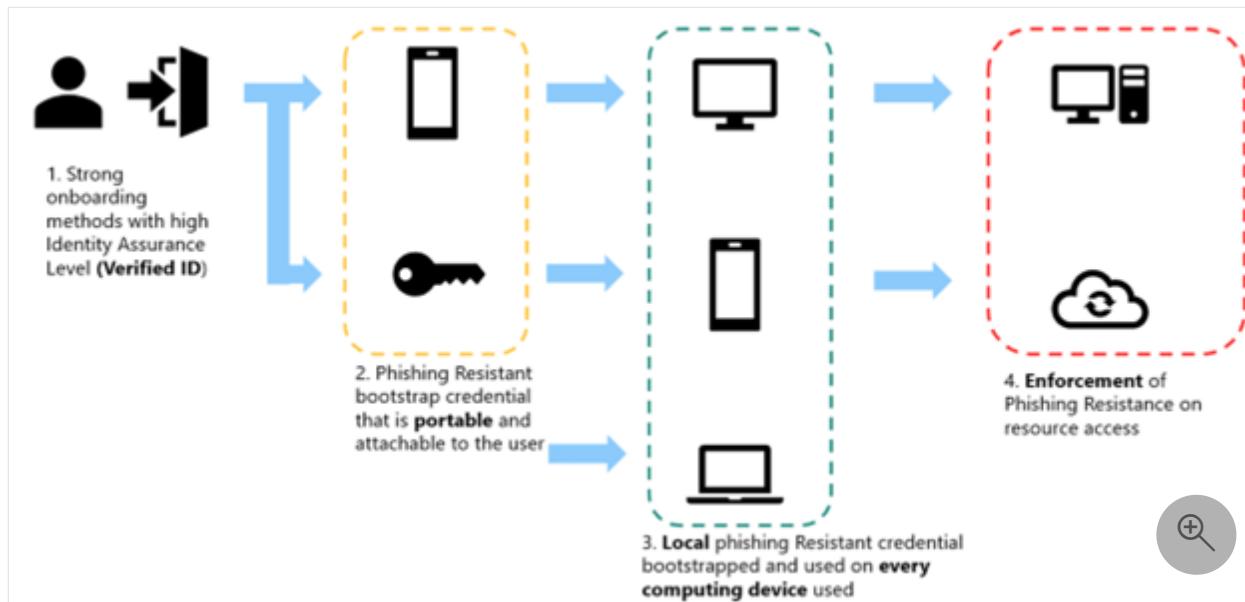
For example, you could execute your deployments and then enforcements in waves that have ranges of dates rather than specific dates:

1. June 1st-15th: Wave 1 cohort registration deployment and campaigns
2. June 16th-30th: Wave 2 cohort registration deployment and campaigns
3. July 1st-15th: Wave 3 cohort registration deployment and campaigns
4. July 16th-31st: Wave 1 cohort enforcement enabled
5. August 1st-15th: Wave 2 cohort enforcement enabled
6. August 16th-31st: Wave 3 cohort enforcement enabled

As you execute these different phases, you may need to slow down depending on the volume of help desk tickets opened and then resume when the volume has subsided. To execute on this strategy, Microsoft recommends that you create a Microsoft Entra ID security group for each wave, and add each group to your policies one at a time. This approach helps to avoid overwhelming your support teams.

Enforce phishing-resistant methods for sign-in

This section focuses on phase 4.



The final phase of a phishing-resistant passwordless deployment is enforcing the use of phishing-resistant credentials. The primary mechanism for doing this in Microsoft Entra ID is [Conditional Access authentication strengths](#). Microsoft recommends you approach enforcement for each persona based on a user/device pair methodology. For example, an enforcement rollout could follow this pattern:

1. Information workers on Windows and iOS
2. Information workers on macOS and Android
3. IT Pros on iOS and Android
4. FLWs on iOS and Android
5. FLWs on Windows and macOS
6. IT Pros on Windows and macOS

Microsoft recommends that you build a report of all your user/device pairs by using sign-in data from your tenant. You can use querying tools like [Azure Monitor](#) and [Workbooks](#). At minimum, try to identify all user/device pairs that match these categories.

Use the previously covered [Phishing-Resistant Passwordless Workbook](#) to assist with the enforcement phase, if possible.

For each user, create a list of which operating systems they regularly use for work. Map the list to the readiness for phishing-resistant sign-in enforcement for that user/device pair.

OS type	Ready for Enforcement	Not Ready for Enforcement
Windows	10+	8.1 and earlier, Windows Server
iOS	17+	16 and earlier
Android	14+	13 and earlier
macOS	13+ (Ventura)	12 and earlier
VDI	Depends ¹	Depends ¹
Other	Depends ¹	Depends ¹

¹For each user/device pair where the device version isn't immediately ready for enforcement, determine how to address the need to enforce phishing-resistance. Consider the following options for older operating systems, virtual desktop infrastructure (VDI), and other operating systems such as Linux:

- Enforce phishing-resistance using external hardware – FIDO2 security keys
- Enforce phishing-resistance using external hardware – smart cards
- Enforce phishing-resistance using remote credentials, such as passkeys in the cross-device authentication flow
- Enforce phishing-resistance using remote credentials inside of RDP tunnels (especially for VDI)

The key task is to measure through data which users and personas are ready for enforcement on particular platforms. Begin your enforcement actions on user/device pairs that are ready for enforcement to "stop the bleeding," and reduce the amount of phishable authentication occurring in your environment.

Then move on to other scenarios where the user/device pairs require readiness efforts. Work your way through the list of user/device pairs until you enforce phishing-resistant authentication across the board.

Create a set of Microsoft Entra ID groups to roll out enforcement gradually. Reuse the groups from the [previous step](#) if you used the wave-based rollout approach.

Recommended enforcement Conditional Access policies

Target each group with a specific Conditional Access policy. This approach helps you roll out your enforcement controls gradually by user/device pair.

Policy	Group name targeted in the policy	Policy – Device platform condition	Policy – Grant control
1	Windows phishing-resistant passwordless ready users	Windows	Require authentication strength – Phishing-resistant MFA
2	macOS phishing-resistant passwordless ready users	macOS	Require authentication strength – Phishing-resistant MFA
3	iOS phishing-resistant passwordless ready users	iOS	Require authentication strength – Phishing-resistant MFA
4	Android phishing-resistant passwordless ready users	Android	Require authentication strength – Phishing-resistant MFA
5	Other phishing-resistant passwordless ready users	Any except Windows, macOS, iOS, or Android	Require authentication strength – Phishing-resistant MFA

Add each user to each group as you determine whether their device and operating system is ready, or they don't have a device of that type. At the end of the rollout, each user should be in one of the groups.

Respond to risk for passwordless users

Microsoft Entra ID Protection helps organizations detect, investigate, and remediate identity-based risks. Microsoft Entra ID Protection provides important and useful detections for your users even after they switch to using phishing-resistant passwordless credentials. For example, some relevant detections for phishing-resistant users include:

- Activity from anonymous IP address
- Admin confirmed user compromised
- Anomalous Token
- Malicious IP address
- Microsoft Entra threat intelligence
- Suspicious browser
- Attacker in the middle
- Possible attempt to access Primary Refresh Token (PRT)
- And others: [Risk detections mapped to riskEventType](#)

Microsoft recommends that Microsoft Entra ID Protection customers take the following actions to best protect their phishing-resistant passwordless users:

1. Review the Microsoft Entra ID Protection deployment guidance: [Plan an ID Protection deployment](#)
2. Configure your risk logs to export to a SIEM
3. Investigate and act on any medium **user** risk
4. Configure a Conditional Access policy to block high risk **users**

After you deploy Microsoft Entra ID Protection, consider using [Conditional Access token protection](#). As users sign in with phishing-resistant passwordless credentials, attacks and detections continue to evolve. For example, when user credentials can no longer be easily phished, attackers may move on to try to exfiltrate tokens from user devices. Token protection helps mitigate this risk by binding tokens to the hardware of the device they were issued to.

Next steps

[Considerations for specific personas in a phishing-resistant passwordless authentication deployment in Microsoft Entra ID](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

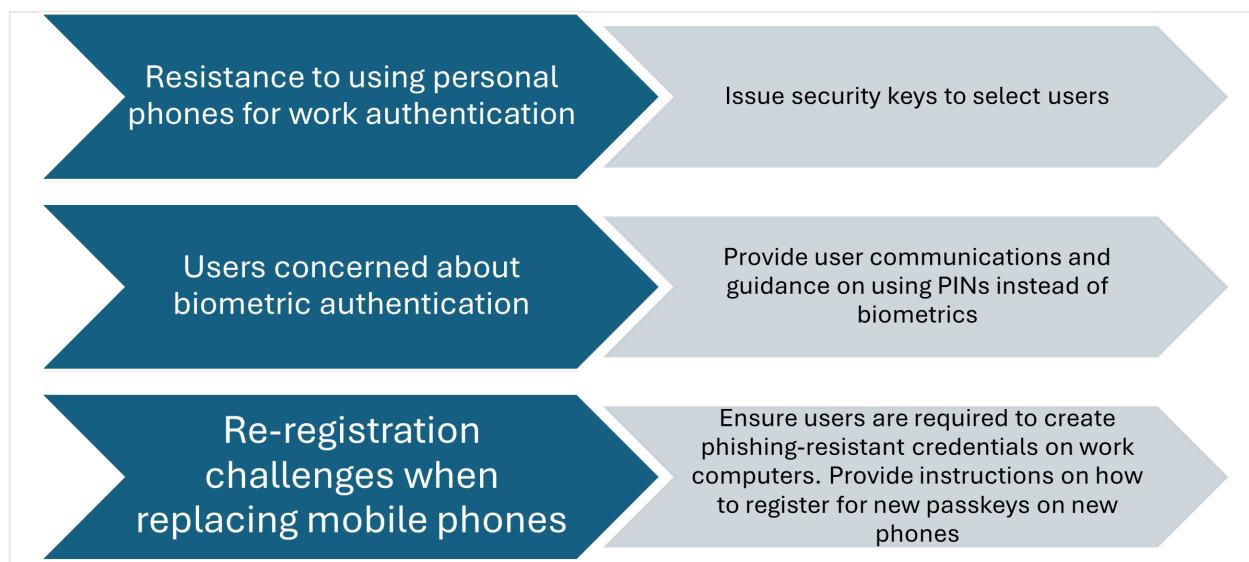
Considerations for specific personas in a phishing-resistant passwordless authentication deployment in Microsoft Entra ID

Article • 03/04/2025

Each persona has its own challenges and considerations that commonly come up during phishing-resistant passwordless deployments. As you identify which personas you need to accommodate, you should factor these considerations into your deployment project planning. The next sections provide specific guidance for each persona.

Information workers

Information workers typically have the simplest requirements and are the easiest to begin your phishing-resistant passwordless deployment with. However, there are still some issues that frequently arise when deploying for these users. Common examples include:



Information worker deployments, just like any other user persona, require proper communication and support. This commonly involves convincing users to install certain apps on their phones, distributing security keys where users won't use apps, addressing concerns about biometrics, and developing processes for helping users recover from partial or total loss of their credentials.

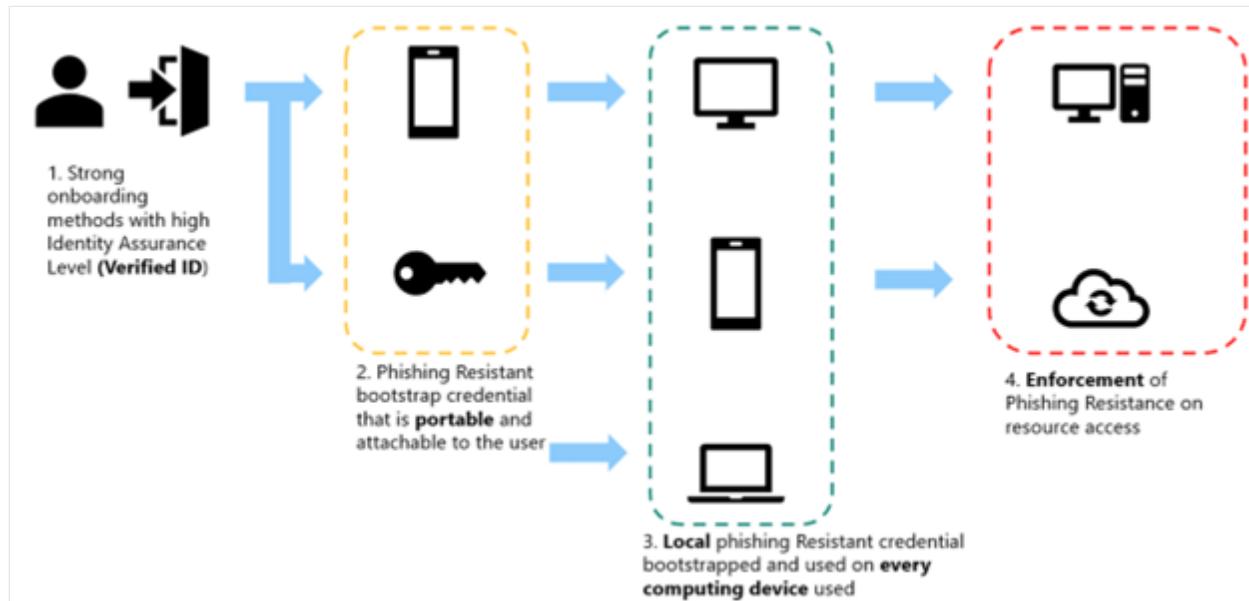
When dealing with concerns about biometrics, make sure that you understand how technologies like Windows Hello for Business handle biometrics. The biometric data is

stored only locally on the device and can't be converted back into raw biometric data even if stolen:

- Windows Hello for Business Biometric data storage

Information worker deployment flow

Phases 1-3 of the deployment flow for information workers should typically follow the standard deployment flow, as shown in the following image. Adjust the methods used at each step as needed in your environment:

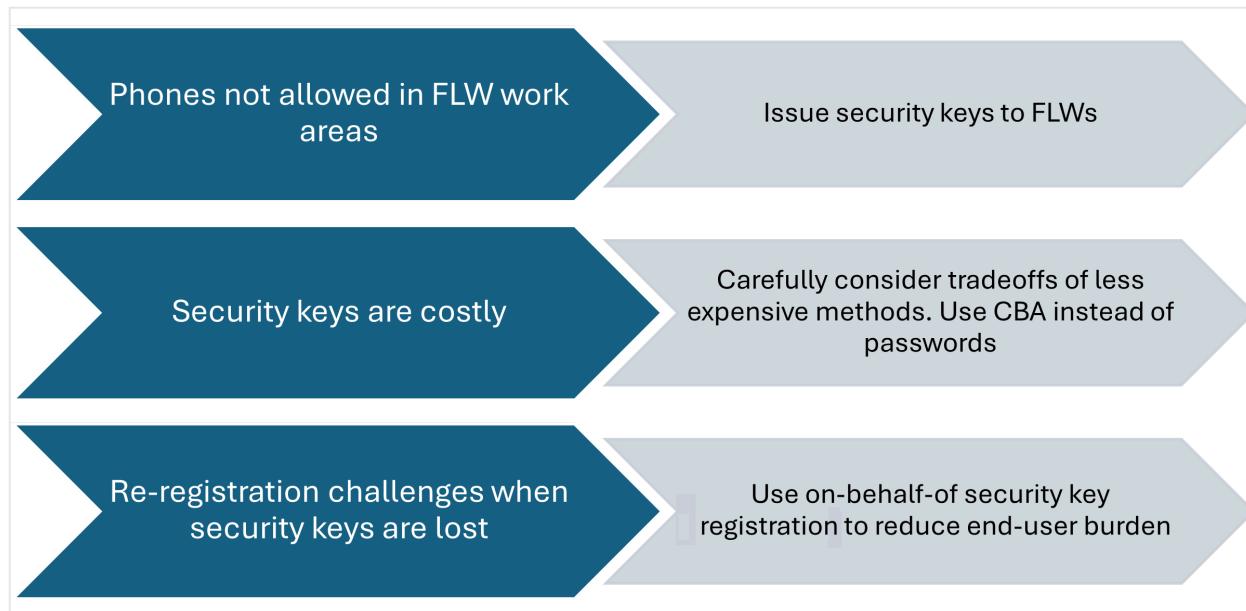


1. Phase 1: Onboarding
 - a. Microsoft Entra Verified ID service used to acquire a Temporary Access Pass
2. Phase 2: Portable credential registration
 - a. Microsoft Authenticator app passkey (preferred)
 - b. FIDO2 security key
3. Phase 3: Local credential registration
 - a. Windows Hello for Business
 - b. Platform SSO Secure Enclave Key

Frontline workers

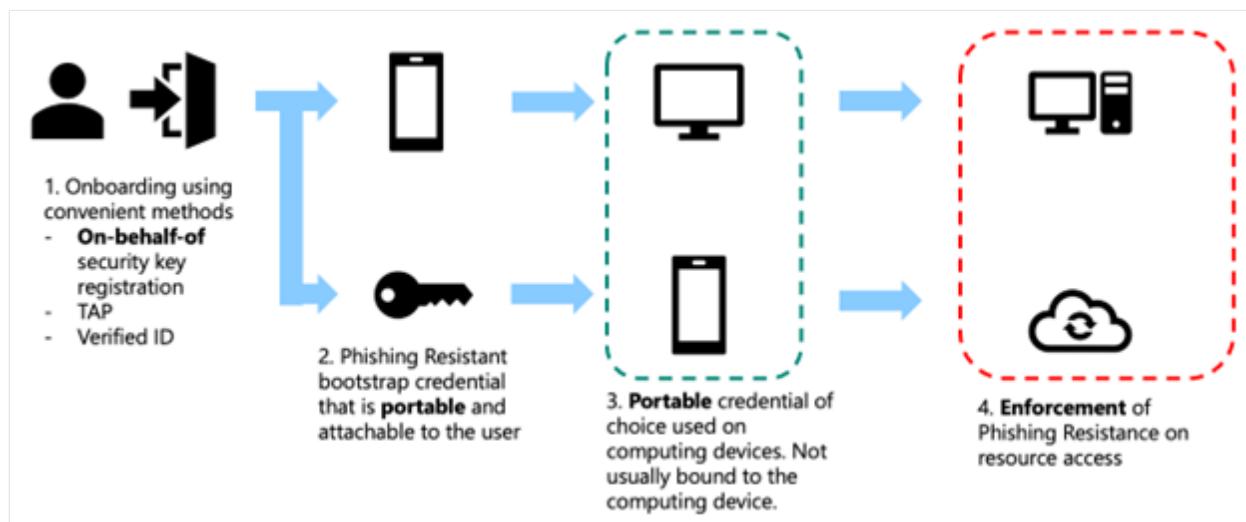
Frontline workers often have more complicated requirements due to increased needs for the portability of their credentials and limitations on which devices they can carry in retail or manufacturing settings. Security keys are a great option for frontline workers, but have a cost that must be considered. In order to achieve phishing-resistance, be sure to balance the cost challenges of security keys against the added deployment burden of smart cards and certificate-based authentication. Consider if there may be

different frontline worker user personas in your environment. It's possible security keys are better for some frontline workers, where smart cards are better for others.



Frontline worker deployment flow

Phases 1-3 of the deployment flow for frontline workers should typically follow a modified flow that emphasizes portable credentials. Many frontline workers may not have a permanent computing device, and they never need a local credential on a Windows or Mac workstation. Instead, they largely rely on portable credentials that they can take with them from device to device. Adjust the methods used at each step as needed in your environment:

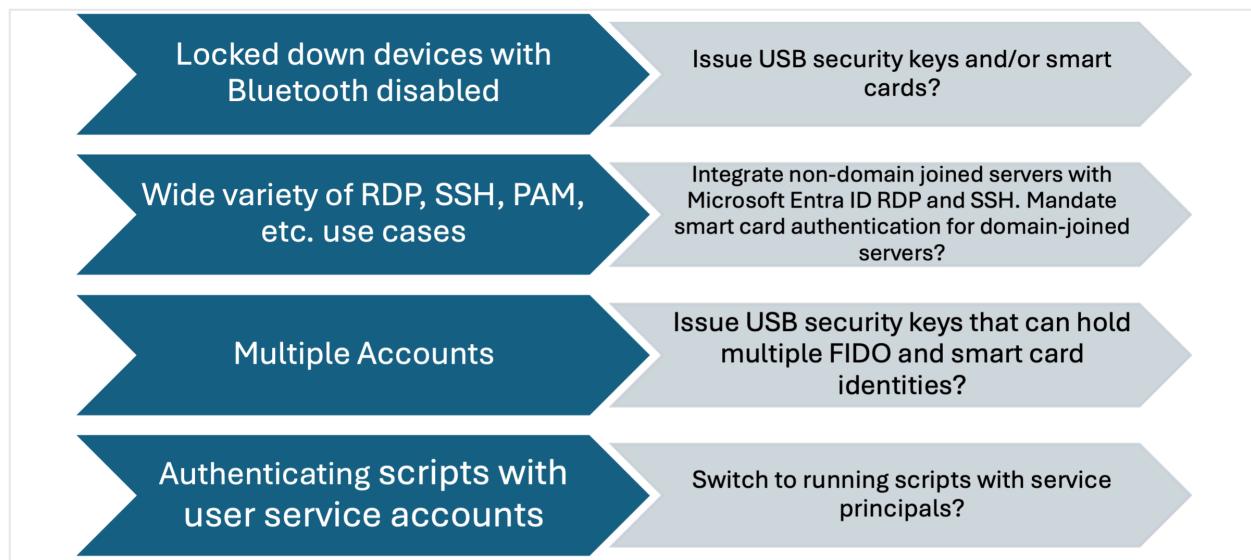


1. Phase 1: Onboarding
 - a. FIDO2 security key on-behalf-of registration (preferred)
 - b. Microsoft Entra Verified ID service used to acquire a Temporary Access Pass
2. Phase 2: Portable credential registration
 - a. FIDO2 security key (preferred)

- b. Smart card
 - c. Microsoft Authenticator app passkey
3. Phase 3 (Optional): Local credential registration
- a. Optional: Windows Hello for Business
 - b. Optional: Platform SSO Secure Enclave Key

IT pros/DevOps workers

IT pros and DevOps workers are especially reliant on remote access and multiple user accounts, which is why they're considered different from information workers. Many of the challenges posed by phishing-resistant passwordless for IT pros are caused by their increased need for remote access to systems and ability to run automations.



Understand the supported options for phishing-resistant with RDP especially for this persona.

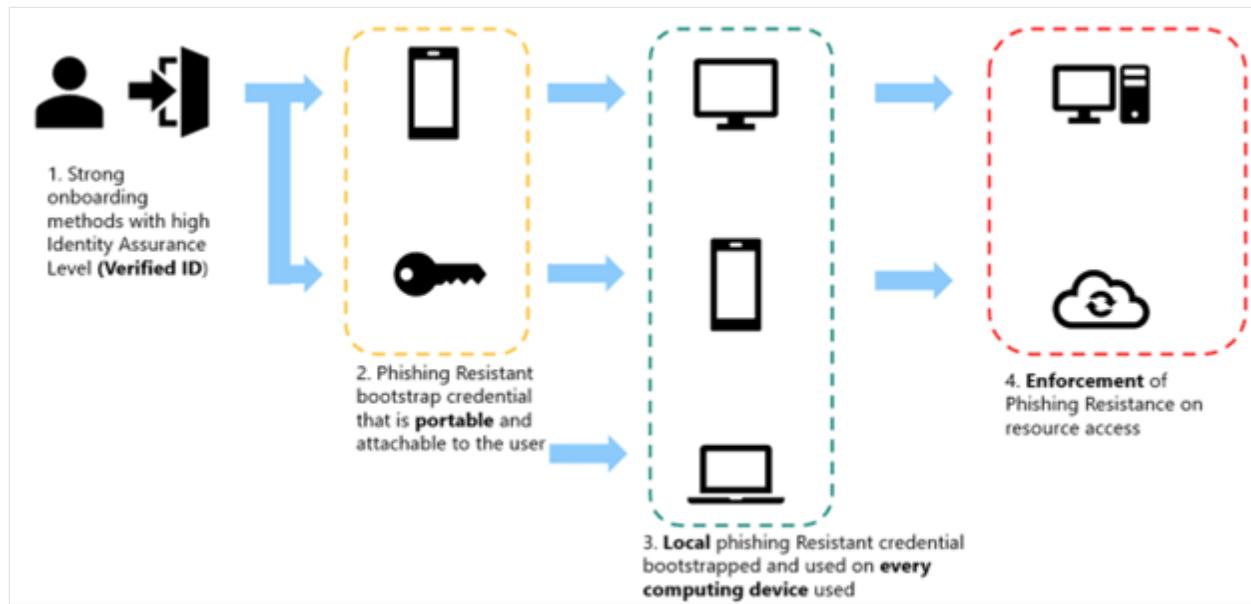
Make sure to understand where users are using scripts that run in the user context and are therefore not using MFA today. Instruct your IT pros on the proper way to run automations using service principals and managed identities. You should also consider processes to allow IT pros and other professionals to request new service principals and get the proper permissions assigned to them.

- [What are managed identities for Azure resources?](#)
- [Securing service principals in Microsoft Entra ID](#)

IT pros/DevOps worker deployment flow

Phases 1-3 of the deployment flow for IT pro/DevOps workers should typically follow the standard deployment flow as previously pictured for the user's primary account. IT

pros/DevOps workers often have secondary accounts that require different considerations. Adjust the methods used at each step as needed in your environment for the primary accounts:



1. Phase 1: Onboarding

- a. Microsoft Entra Verified ID service used to acquire a Temporary Access Pass

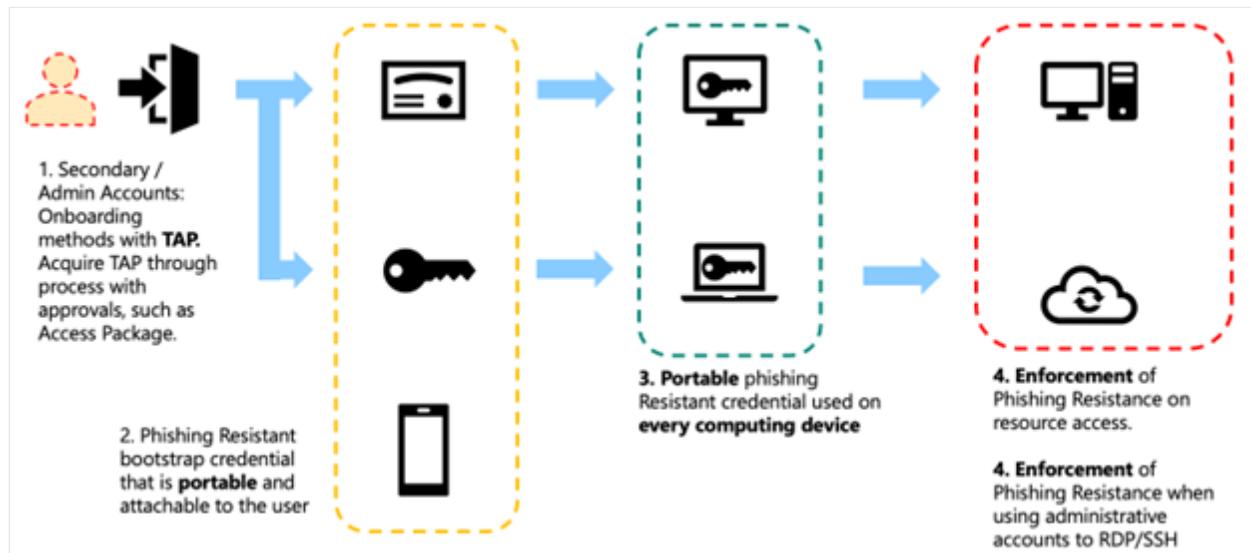
2. Phase 2: Portable credential registration

- a. Microsoft Authenticator app passkey (preferred)
- b. FIDO2 security key

3. Phase 3: Local credential registration

- a. Windows Hello for Business
- b. Platform SSO Secure Enclave Key

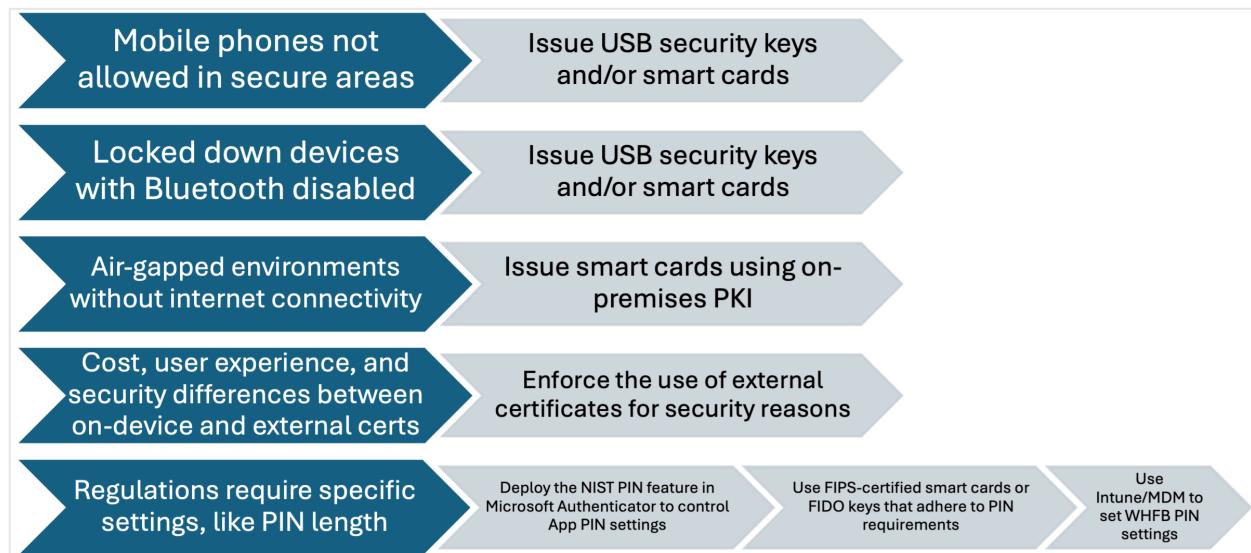
If your IT pro/DevOps workers have secondary accounts, you may need to handle those accounts differently. For example, for secondary accounts you may choose to use alternative portable credentials and forego local credentials on your computing devices entirely:



1. Phase 1: Onboarding
 - a. Microsoft Entra Verified ID service used to acquire a Temporary Access Pass (preferred)
 - b. Alternate process to provide TAPs for secondary accounts to the IT pro/DevOps worker
2. Phase 2: Portable credential registration
 - a. Microsoft Authenticator app passkey (preferred)
 - b. FIDO2 security key
 - c. Smart card
3. Phase 3: Portable credentials used rather than local credentials

Highly regulated workers

Highly regulated workers pose more challenges than the average information worker because they may work on locked down devices, work in locked down environments, or have special regulatory requirements they must satisfy.



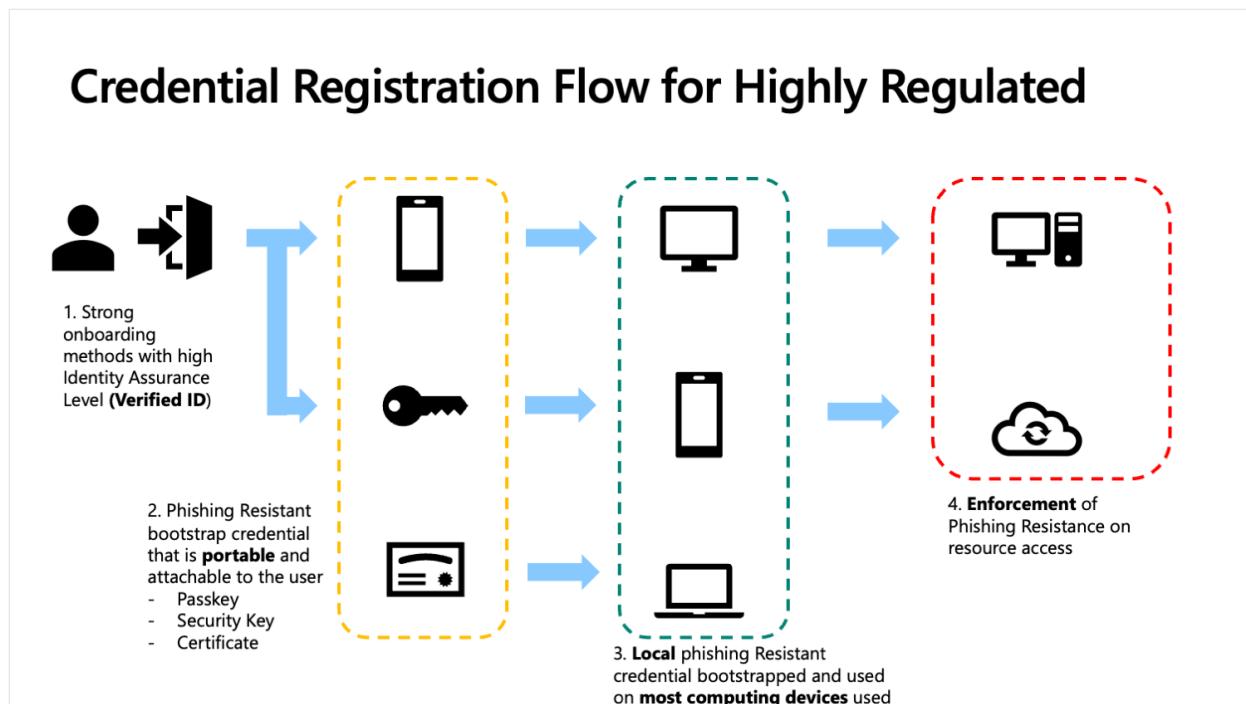
Highly regulated workers often use smart cards due to regulated environments already having heavy adoption of PKI and smart card infrastructure. However, consider when smart cards are desirable and required and when they can be balanced with more user-friendly options, such as Windows Hello for Business.

Highly regulated worker deployment flow without PKI

If you don't plan to use certificates, smart cards, and PKI, then the highly regulated worker deployment closely mirrors the information worker deployment. For more information, see [Information workers](#).

Highly regulated worker deployment flow with PKI

If you plan to use certificates, smart cards, and PKI, then the deployment flow for highly regulated workers typically differs from the information worker setup flow in key places. There's an increased need to identify if local authentication methods are viable for some users. Similarly, you need to identify if there are some users who need portable-only credentials, such as smart cards, that can work without internet connections. Depending on your needs, you may adjust the deployment flow further, and tailor it to the various user personas identified in your environment. Adjust the methods used at each step as needed in your environment:



1. Phase 1: Onboarding
 - a. Microsoft Entra Verified ID service used to acquire a Temporary Access Pass (preferred)
 - b. Smart card registration on behalf of the user, following an identity proofing process
2. Phase 2: Portable credential registration
 - a. Smart card (preferred)
 - b. FIDO2 security key
 - c. Microsoft Authenticator app passkey
3. Phase 3 (Optional): Local credential registration
 - a. Optional: Windows Hello for Business
 - b. Optional: Platform SSO Secure Enclave Key

① Note

It's always recommended that users have at least two credentials registered. This ensures the user has a backup credential available if something happens to their other credentials. For highly regulated workers, it's recommended that you deploy passkeys or Windows Hello for Business in addition to any smart cards you deploy.

Next steps

[Deploy a phishing-resistant passwordless authentication deployment in Microsoft Entra ID](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Considerations for Remote Desktop Connections in a phishing-resistant passwordless authentication deployment in Microsoft Entra ID

Article • 05/06/2025

Organizations deploying phishing-resistant passwordless typically have a need for some of their personas to use remote desktop technology to facilitate productivity, security, or administration. The two basic use cases are:

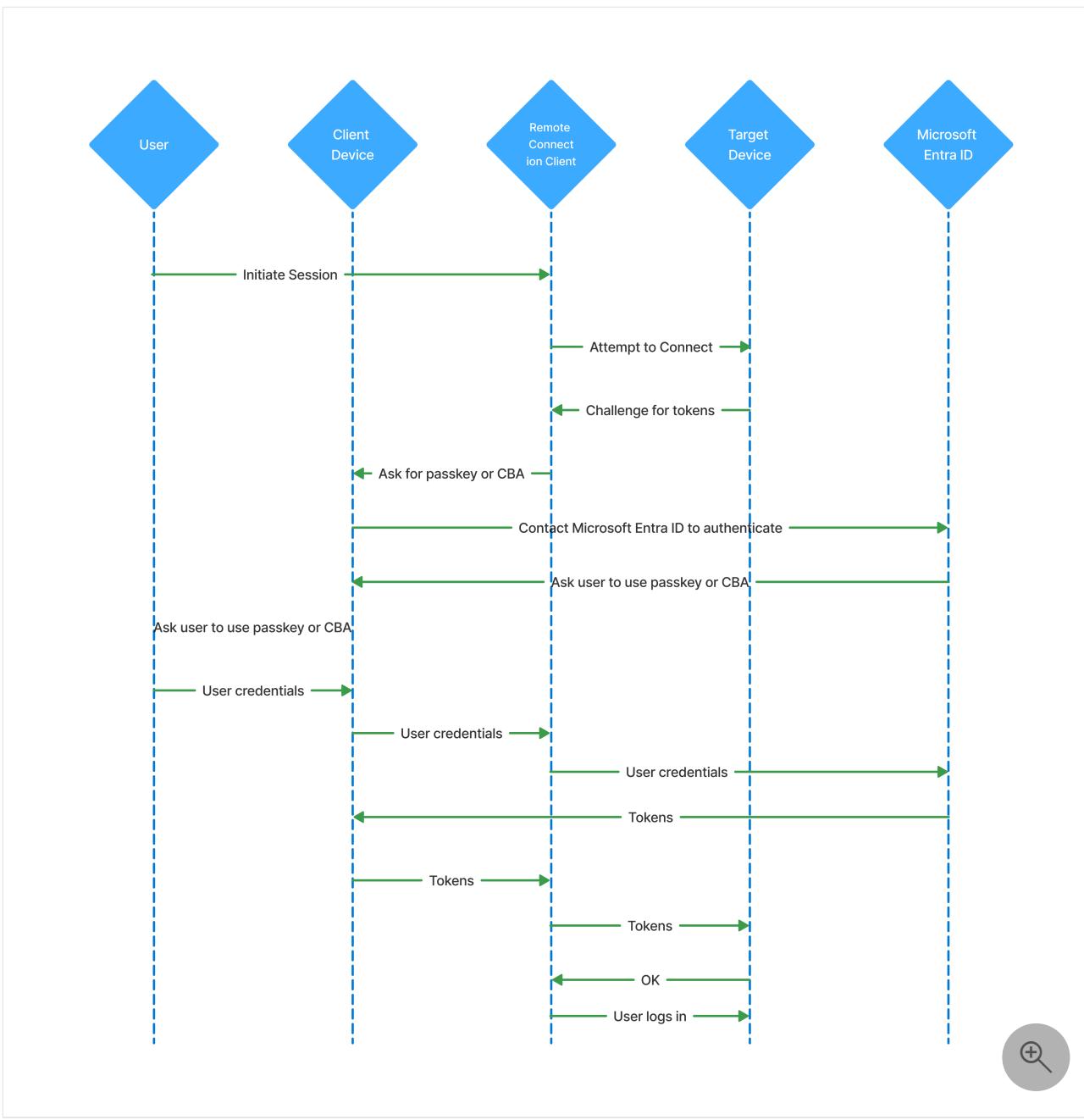
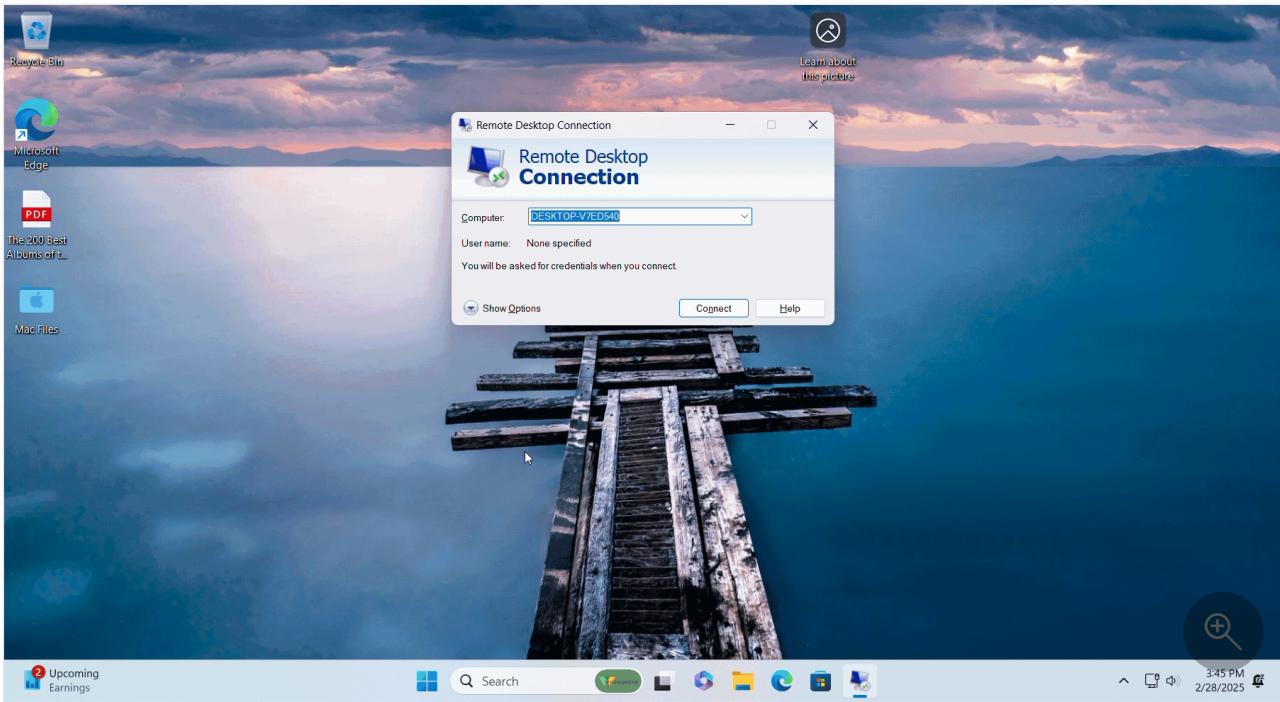
- Initializing and authenticating a remote desktop connection session from a local client to a remote machine using phishing-resistant passwordless credentials
- Utilizing phishing-resistant passwordless credentials inside of an established remote desktop connection session

Review the specific considerations for each use case.

Passwordless Remote Desktop Connection Session Initiation

Remote Desktop connection components

Windows remote desktop protocol involves three primary components, all of which need to properly support phishing-resistant passwordless credentials for initiating a remote desktop connection session using these credentials. If any of these components isn't able to properly function or lacks support for certain passwordless credentials, then one or both scenarios outlined won't function. This guide focuses on passkey/FIDO2 support and Cert-Based Authentication (CBA) support.



Step through the following sections to determine if support for phishing-resistant passwordless is expected across all three components you're utilizing. Repeat this process if you have multiple scenarios that require evaluation.

Client platform

There are several different commonly used operating systems for local clients that are used to instantiate remote desktop sessions. Commonly used options include:

- Windows 10+
- Windows Server
- macOS
- iOS
- Android
- Linux

Support for phishing-resistant passwordless and remote desktop connection depends on the client platform having support for passkey protocols, most notably [Client To Authenticator Protocol \(CTAP\)](#) and [WebAuthn](#). CTAP is a communication layer between roaming authenticators, such as FIDO2 security keys or passkeys on a mobile device, and a client platform. Most client platforms support these protocols, but there are certain platforms that don't. In some cases, such as with dedicated thin client devices running specialized OSes, you should contact the vendor to confirm support.

[Microsoft Entra certificate-based authentication \(CBA\)](#) requires configuration in Microsoft Entra ID so that users can utilize certificates from your Public Key Infrastructure (PKI) for authentication. This article does not address on-premises only certificate-based authentication implementations.

[+] [Expand table](#)

Client Platform	FIDO Support	Microsoft Entra CBA	Notes
Windows 10+	Yes	Yes	
Windows Server	Partial	Yes	Windows Server isn't recommended for client computing devices. Windows Server jump servers may impede FIDO-based phishing-resistant passwordless. If you use jump servers, then CBA is recommended instead of FIDO

Client Platform	FIDO Support	Microsoft Entra CBA	Notes
macOS	Yes	Yes	Not all Apple web frameworks support FIDO
iOS	Yes	Yes	Not all Apple web frameworks support FIDO
Android	Yes	Yes	
Linux	Maybe	Yes	Confirm FIDO support with the Linux distro vendor

Target platform

The target platform is critical for determining if phishing-resistant passwordless authentication is supported for establishing the remote desktop connection session itself.

[\[+\] Expand table](#)

Target Platform	Remote Desktop Connection Session Initialization FIDO Support	Remote Desktop Connection Session Initialization Microsoft Entra CBA
Windows 10+ Microsoft Entra joined	Yes	Yes
Windows Server Microsoft Entra joined	Yes ¹	Yes
Windows 10+ Microsoft Entra hybrid joined	Yes	Yes
Windows Server Microsoft Entra hybrid joined	Yes ¹	Yes
Windows 10+ Microsoft Entra registered	No	No
Windows 10+ on-premises domain joined only	No	No
Windows Server on-premises domain joined only	No	No
Windows 10+ Workgroup	No	No
Azure Arc-managed Windows Server standalone/workgroup ²	Yes	Yes

1. Only applies to Microsoft Entra joined or Hybrid Joined servers running Windows Server 2022 or later

2. Only applies to Microsoft Entra joined servers running Windows Server 2025 or later

Remote Desktop connection client

Client platform support for phishing-resistant authentication alone isn't sufficient to support phishing-resistant authentication for remote desktop connection sessions. The remote desktop connection client used must also support the necessary components for these credentials to work properly. Review many of the commonly used remote desktop connection clients and their various supported options:

 Expand table

Remote Desktop Connection Client	Remote Desktop Connection Session Initialization FIDO Support	Remote Desktop Connection Session Initialization Microsoft Entra CBA
MSTSC.exe for Windows Client	Yes	Yes
MSTSC.exe for Windows Server 2022+	Yes	Yes
MSTSC.exe for Windows Server 2019 or earlier	No	No
Windows App for Windows	Yes	Yes
Windows App for macOS	Yes	Yes
Windows App for iOS	Yes	Yes
Windows App for Android	Yes	Yes
Windows 365 Web App	No	No
Third Party Remote Desktop Connection Client	Maybe	Maybe

Important

Client and target devices must be Microsoft Entra joined, Microsoft Entra hybrid joined, or Microsoft Entra registered to the same tenant. Cross-tenant authentication

will not work, the client device will not be able to authenticate to the target device if they are joined to different tenants.

Evaluate support for your scenarios

If any one of the three components outlined in this document don't support your scenario, then your scenario isn't expected to work. To evaluate, consider each component for remote desktop connection session auth and in-session credential usage. Repeat this process for every scenario in your environment to understand which scenarios are expected to work and which aren't.

Example 1

For example, here's how you might evaluate if your scenario is "my Information Workers need to use their Windows devices to access Azure Virtual Desktop, need to authenticate the remote desktop connection session using a Microsoft Authenticator passkey, and use the passkey inside the remote desktop connection session in the Microsoft Edge browser":

[+] Expand table

Scenario	Client Platform	Target Platform	Remote Desktop Connection Client	Supported?
Remote Desktop Connection Session Initialization using Auth App Passkey	Windows 11 Microsoft Entra joined/Hybrid Joined/Standalone	Azure Virtual Desktop Microsoft Entra joined	Windows App	Yes + Yes + Yes = Yes
Remote Desktop Connection In-Session Auth using Auth App Passkey	Windows 11 Microsoft Entra joined/Hybrid Joined/Standalone	Azure Virtual Desktop Microsoft Entra joined	Windows App	Yes + Yes + Yes = Yes

In this example, both the remote desktop connection session itself and in-session apps can take advantage of the user's passkey. Phishing-resistant passwordless should work broadly.

Example 2

Here's how you might evaluate if your scenario is "my Information Workers need to use their macOS devices to access Azure Virtual Desktop, need to authenticate the remote desktop

connection session using a Microsoft Authenticator passkey, and use the passkey inside the remote desktop connection session":

[Expand table](#)

Scenario	Client Platform	Target Platform	Remote Desktop Connection Client	Supported?
Remote Desktop Connection Session Initialization using Auth App Passkey	macOS 15	Azure Virtual Desktop Microsoft Entra joined	Windows App	Yes+Yes+Yes = Yes
Remote Desktop Connection In-Session Auth using Auth App Passkey	macOS 15	Azure Virtual Desktop Microsoft Entra joined	Windows App	Yes+Yes+No = No

In this example, users can use their passkey to establish the remote desktop connection session, but can't use it inside of the remote desktop connection session because the Windows App on macOS doesn't support this functionality yet. You can wait for better passkey support in the remote desktop connection client or you can switch to another credential, such as certificates with CBA.

Example 3

Here's how you might evaluate if your scenario is "my admins need to use their Windows devices to access on-premises Windows Servers, need to authenticate the remote desktop connection session using a certificate, and use the certificate inside the remote desktop connection session":

[Expand table](#)

Scenario	Client Platform	Target Platform	Remote Desktop Connection Client	Supported?
Remote Desktop Connection Session Initialization using Certificate	Windows 11	Domain-Joined Windows Server	MSTSC.exe	Yes+Yes+Yes = Yes
Remote Desktop Connection In-Session Auth using Certificate	Windows 11	Domain-Joined Windows Server	MSTSC.exe	Yes+Yes+Yes = Yes

In this example, users can use their certificate to establish the remote desktop connection session and also use the certificate inside the remote desktop connection session. This scenario

won't work with a passkey however, since the domain-joined Windows server can't use a passkey to set up a remote desktop connection session or inside the session.

Example 4

Here's how you might evaluate if your scenario is "my frontline workers need to use a Linux-based thin client to access on-premises domain-joined Windows Virtual Desktop Infrastructure (VDI) clients that are NOT Microsoft Entra hybrid joined, need to authenticate the remote desktop connection session using a FIDO2 security key, and use the FIDO2 security key inside the remote desktop connection session":

 Expand table

Scenario	Client Platform	Target Platform	Remote Desktop Connection Client	Supported?
Remote Desktop Connection Session Initialization using FIDO2 Security Key	Linux Embedded Distro	Domain-Joined Windows 11	Vendor-Provided Client	Maybe+No+No = No
Remote Desktop Connection In-Session Auth using FIDO2 Security Key	Linux Embedded Distro	Domain-Joined Windows 11	Vendor-Provided Client	Maybe+Yes+Maybe = Maybe

In this example, users likely can't use their FIDO2 security keys for remote desktop connection at all because the thin client OS and remote desktop connection client don't support FIDO2/passkeys in every scenario required. Work with your thin client vendor to understand their roadmap for support. Additionally, plan on Microsoft Entra hybrid joining or Microsoft Entra joining the Target Platform virtual machines so that passkeys can be better supported.

Next steps

[Deploy a phishing-resistant passwordless authentication deployment in Microsoft Entra ID](#)

Enable passkeys (FIDO2) for your organization

Article • 05/05/2025

For enterprises that use passwords today, passkeys (FIDO2) provide a seamless way for workers to authenticate without entering a username or password. Passkeys (FIDO2) provide improved productivity for workers, and have better security.

This article lists requirements and steps to enable passkeys in your organization. After you complete these steps, users in your organization can then register and sign in to their Microsoft Entra account using a passkey stored on a FIDO2 security key or in Microsoft Authenticator.

For more information about enabling passkeys in Microsoft Authenticator, see [How to enable passkeys in Microsoft Authenticator](#).

For more information about passkey authentication, see [Support for FIDO2 authentication with Microsoft Entra ID](#).

! Note

Microsoft Entra ID currently supports device-bound passkeys stored on FIDO2 security keys and in Microsoft Authenticator. Microsoft is committed to securing customers and users with passkeys. We're investing in both synced and device-bound passkeys for work accounts.

Requirements

- Users must complete multifactor authentication (MFA) within the past five minutes before they can register a passkey (FIDO2).
- Users need a [FIDO2 security key eligible for attestation with Microsoft Entra ID](#) or Microsoft Authenticator.
- Devices must support passkey (FIDO2) authentication. For Windows devices that are joined to Microsoft Entra ID, the best experience is on Windows 10 version 1903 or higher. Hybrid-joined devices must run Windows 10 version 2004 or higher.

Passkeys (FIDO2) are supported across major scenarios on Windows, macOS, Android, and iOS. For more information on supported scenarios, see [Support for FIDO2 authentication in Microsoft Entra ID](#).

(!) Note

Support for same-device registration in Edge on Android is coming soon.

Passkey (FIDO2) Authenticator Attestation GUID (AAGUID)

The FIDO2 specification requires each security key vendor to provide an Authenticator Attestation GUID (AAGUID) during registration. An AAGUID is a 128-bit identifier indicating the key type, such as the make and model. Passkey (FIDO2) providers on desktop and mobile devices are also expected to provide an AAGUID during registration.

(!) Note

The vendor must ensure that the AAGUID is identical across all substantially identical security keys or passkey (FIDO2) providers made by that vendor, and different (with high probability) from the AAGUIDs of all other types of security keys or passkey (FIDO2) providers. To ensure this, the AAGUID for a given security key model or passkey (FIDO2) provider should be randomly generated. For more information, see [Web Authentication: An API for accessing Public Key Credentials - Level 2 \(w3.org\)](#) ↗.

You can work with your security key vendor to determine the AAGUID of the passkey (FIDO2), or see [FIDO2 security keys eligible for attestation with Microsoft Entra ID](#). If the passkey (FIDO2) is already registered, you can find the AAGUID by viewing the authentication method details of the passkey (FIDO2) for the user.

The screenshot shows the Microsoft Entra ID portal's 'User' dashboard for a user named 'Test User10-Test'. The left sidebar shows navigation options like 'Profile', 'Assigned roles', 'Administrative units', 'Groups (Preview)', 'Applications', 'Licenses', 'Devices', 'Azure role assignments', and 'Authentication methods'. The 'Authentication methods' option is selected. On the right, a modal window titled 'FIDO2 security key details' displays the following information:

ID	Value
ID	SQCb3QvqPRFUNbutG2dtbOVPKsChPZDUF-dja9r1dj-QuqOOVKCY1cyrqiaD87jkPUAuRWDab4wxoxS5olwr8AMhkWYgG6v9KhYgED2734eTyzVdaVFVEYzQ
Display name	E10-thinc-bio
Created	5/17/2021, 12:44:44 PM
Model	Ensuri ThinC FIDO2 Biometric Security Key
AA Guid	454e5346-4944-4ffd-6c93-8e9267193e9a
Attestation Level	Attested
Attestation Certificates	b22279838ecf2d3fc5193b38aefb3389b2de7198

Enable passkey (FIDO2) authentication method

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Policy Administrator**.
2. Browse to **Entra ID > Authentication methods > Policies**.
3. Under the method **Passkey (FIDO2)**, set the toggle to **Enable**. Select **All users** or **Add groups** to select specific groups. *Only security groups are supported.*
4. On the **Configure** tab:
 - Set **Allow self-service set up** to **Yes**. If set to **No**, users can't register a passkey by using [Security info](#), even if passkeys (FIDO2) are enabled by the Authentication methods policy.
 - Set **Enforce attestation** to **Yes** if your organization wants to be assured that a FIDO2 security key model or passkey provider is genuine and comes from the legitimate vendor.
 - For FIDO2 security keys, we require security key metadata to be published and verified with the FIDO Alliance Metadata Service, and also pass Microsoft's another set of validation testing. For more information, see [Become a Microsoft-compatible FIDO2 security key vendor](#).
 - Passkeys in Microsoft Authenticator also support attestation. For more information, see [How passkey attestation works with Authenticator](#).

 **Warning**

Attestation enforcement governs whether a passkey (FIDO2) is allowed only during registration. Users who register a passkey (FIDO2) without attestation aren't blocked from sign-in if **Enforce attestation** is set to **Yes** later.

Key Restriction Policy

- **Enforce key restrictions** should be set to **Yes** only if your organization wants to only allow or disallow certain security key models or passkey providers, which are identified by their AAGUID. You can work with your security key vendor to determine the AAGUID of the passkey. If the passkey is already registered, you can find the AAGUID by viewing the authentication method details of the passkey for the user.

 **Warning**

Key restrictions set the usability of specific models or providers for both registration and authentication. If you change key restrictions and remove an AAGUID that you

previously allowed, users who previously registered an allowed method can no longer use it for sign-in.

The screenshot shows the 'Passkey (FIDO2) settings' page in Microsoft Azure. At the top, there's a navigation bar with 'Microsoft Azure' and a breadcrumb trail: Home > Security | Authentication methods > Authentication methods | Policies > Passkey (FIDO2) settings. Below the title, a note says: 'Passkeys are a phishing-resistant, standards-based passwordless authentication method available from a variety of vendors. [Learn more](#). Passkeys are not usable in the Self-Service Password Reset flow.' There are two tabs: 'Enable and Target' and 'Configure', with 'Configure' being the active tab. Under 'GENERAL', there are two buttons: 'Allow self-service set up' (Yes selected) and 'Enforce attestation' (Yes selected). Under 'KEY RESTRICTION POLICY', there are two buttons: 'Enforce key restrictions' (Yes selected) and 'Restrict specific keys' (Allow selected). A checkbox for 'Microsoft Authenticator' is checked. On the right side, there's a search icon.

5. After you finish the configuration, select **Save**.

! Note

If you see an error when you try to save, replace multiple groups with a single group in one operation, and then click **Save** again.

Provision FIDO2 security keys using Microsoft Graph API (preview)

Currently in preview, administrators can use [Microsoft Graph](#) and custom clients to provision FIDO2 security keys on behalf of users [↗](#). Provisioning requires the [Authentication Administrator role](#) or a client application with UserAuthenticationMethod.ReadWrite.All permission. The provisioning improvements include:

- The ability to request WebAuthn creation Options from Microsoft Entra ID
- The ability to register the provisioned security key directly with Microsoft Entra ID

With these new APIs, organizations can build their own clients to provision passkey (FIDO2) credentials on security keys on behalf of a user. To simplify this process, three main steps are required.

1. **Request** creationOptions for a user: Microsoft Entra ID returns the necessary data for your client to provision a passkey (FIDO2) credential. This includes information such as user information, relying party ID, credential policy requirements, algorithms, registration challenge and more.
2. **Provision** the passkey (FIDO2) credential with the creation Options: Use the `creationOptions` and a client that supports the Client to Authenticator Protocol (CTAP) to provision the credential. During this step, you need to insert the security key and set a PIN.
3. **Register** the provisioned credential with Microsoft Entra ID: Use the formatted output from the provisioning process to provide Microsoft Entra ID the necessary data to register the passkey (FIDO2) credential for the targeted user.

Request - Get FIDO2 Credential options from Microsoft Entra ID

Provision – Client App invokes CTAP and creates cred on device

Register - Provide registration details to Microsoft Entra ID

Enable passkeys (FIDO2) using Microsoft Graph API

In addition to using the Microsoft Entra admin center, you can also enable passkeys (FIDO2) by using the Microsoft Graph API. To enable passkeys (FIDO2), you need to update the Authentication methods policy as at least an [Authentication Policy Administrator](#).

To configure the policy using Graph Explorer:

1. Sign in to [Graph Explorer](#) and consent to the **Policy.Read.All** and **Policy.ReadWrite.AuthenticationMethod** permissions.

2. Retrieve the Authentication methods policy:

JSON

GET

```
https://graph.microsoft.com/v1.0/authenticationMethodsPolicy/authenticationMethodConfigurations/FIDO2
```

3. To disable attestation enforcement and enforce key restrictions to only allow the AAGUID for RSA DS100 for example, perform a PATCH operation using the following request body:

JSON

PATCH

```
https://graph.microsoft.com/v1.0/authenticationMethodsPolicy/authenticationMethodConfigurations/FIDO2
```

Request Body:

{

```
    "@odata.type": "#microsoft.graph.fido2AuthenticationMethodConfiguration",
    "isAttestationEnforced": false,
    "keyRestrictions": {
        "isEnforced": true,
        "enforcementType": "allow",
        "aaGuids": [
            "7e3f3d30-3557-4442-bdae-139312178b39",
            <insert previous AAGUIDs here to keep them stored in policy>
        ]
    }
}
```

4. Make sure that the passkey (FIDO2) policy is updated properly.

JSON

GET

```
https://graph.microsoft.com/v1.0/authenticationMethodsPolicy/authenticationMethodConfigurations/FIDO2
```

Delete a passkey (FIDO2)

To remove a passkey (FIDO2) associated with a user account, delete it from the user's authentication method.

1. Sign in to the [Microsoft Entra admin center](#) and search for the user whose passkey (FIDO2) needs to be removed.
2. Select **Authentication methods** > right-click **Passkey (device-bound)** and select **Delete**.

Enforce passkey (FIDO2) sign-in

To make users sign in with a passkey (FIDO2) when they access a sensitive resource, you can:

- Use a built-in phishing-resistant authentication strength

Or

- Create a custom authentication strength

The following steps show how to create a custom authentication strength. It's a Conditional Access policy that allows passkey (FIDO2) sign-in for only a specific security key model or passkey (FIDO2) provider. For a list of FIDO2 providers, see [FIDO2 security keys eligible for attestation with Microsoft Entra ID](#).

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Conditional Access Administrator**.
2. Browse to **Entra ID** > **Authentication methods** > **Authentication strengths**.
3. Select **New authentication strength**.
4. Provide a **Name** for your new authentication strength.
5. Optionally provide a **Description**.
6. Select **Passkeys (FIDO2)**.
7. Optionally, if you want to restrict a specific AAGUID, select **Advanced options** > **Add AAGUID**. Enter the AAGUID, and select **Save**.
8. Choose **Next** and review the policy configuration.

Known issues

Security key provisioning

Administrator provisioning of security keys is in preview. See [Microsoft Graph and custom clients to provision FIDO2 security keys on behalf of users](#).

Guest users

Registration of passkey (FIDO2) credentials isn't supported for internal or external guest users, including B2B collaboration users in the resource tenant.

UPN changes

If a user's UPN changes, you can no longer modify passkeys (FIDO2) to account for the change. If the user has a passkey (FIDO2), they need to sign in to [Security info](#), delete the old passkey (FIDO2), and add a new one.

Next steps

[Native app and browser support of passkey \(FIDO2\) passwordless authentication](#)

[FIDO2 security key Windows 10 sign in](#)

[Enable FIDO2 authentication to on-premises resources](#)

[Register security keys on behalf of users](#)

[Learn more about device registration](#)

[Learn more about Microsoft Entra multifactor authentication](#)

Register a passkey

Article • 03/04/2025

This article shows how users can register a security key using the **Passkey** flow. For registration on a mobile device, see [Register a passkey using a mobile device](#).

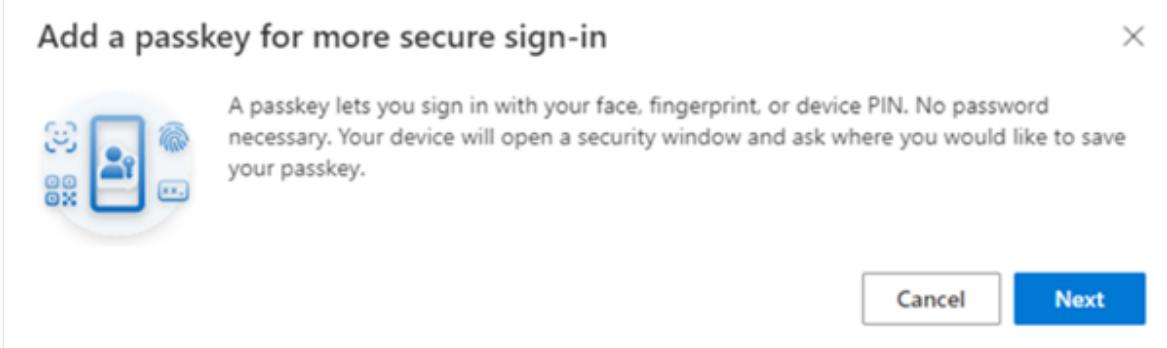
ⓘ Note

Looking to provide passkeys (FIDO2) on behalf of users? Use our [APIs](#).

For more information about enabling passkeys in Microsoft Authenticator, see [How to enable passkeys in Microsoft Authenticator](#).

Manual registration

1. Users can register a passkey (FIDO2) as an authentication method by navigating and completing the process from a browser at [Security info](#).
2. Tap **Add sign-in method** > **Choose a method** > **Passkey** > **Add**.
3. Sign in with multifactor authentication (MFA) before adding a passkey, then tap **Next**.
 - a. If you don't have at least one MFA method registered, you must add one.
 - b. An Authentication Policy Administrator can also issue a [Temporary Access Pass](#) to allow a user to strongly authenticate and register a passkey.

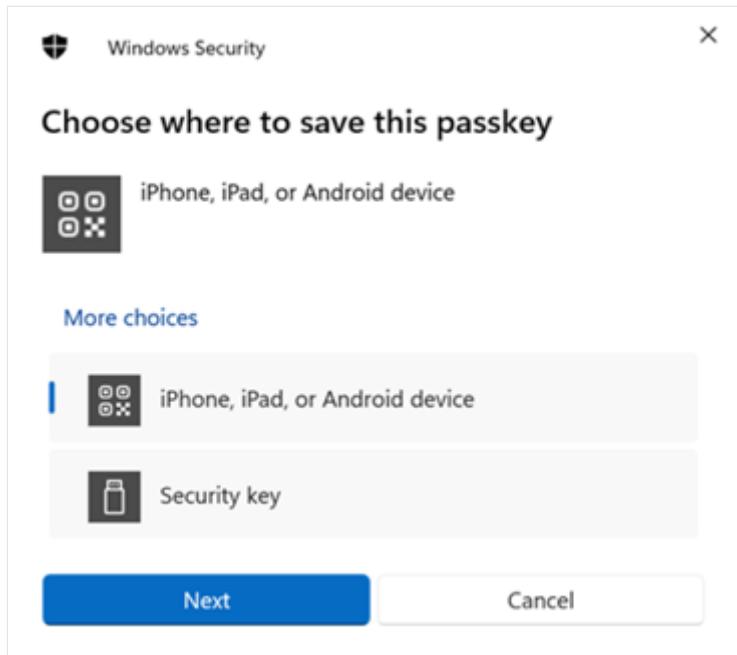


4. A security dialog opens on your device and asks where to save your passkey.

ⓘ Note

Options displayed vary depending on your browser and device operating system. If the device where you started the registration process supports

passkeys, you'll be asked to save the passkey to that device. Select **Use another device** or **More options** to display additional ways for you to save the passkey.



5. If your organization allows saving a passkey to a security key:
 - a. Choose **Security Key**.
 - b. Follow the guidance and insert or connect your security key when requested.
 - c. You're prompted to create or enter a PIN for your security key, then perform the required gesture for the key.
 - d. Upon completion, review any additional information from the security dialog, then tap Ok or Continue.
6. After you're redirected to Security info, you can change the default name for the new sign-in method.
7. Tap **Done** to finish registering the new method.

Next steps

- [Choosing authentication methods for your organization](#)

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Register a passkey using a mobile device

Article • 03/04/2025

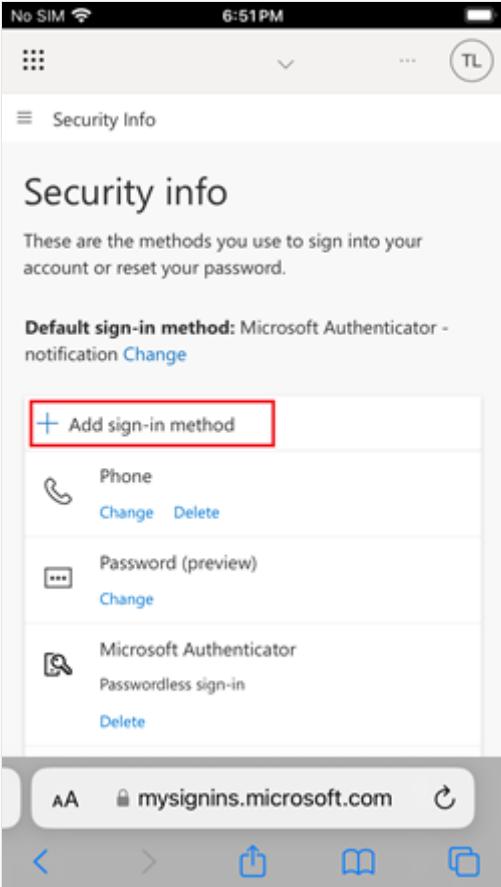
This article shows how to register a security key with your iOS or Android device.

You can also register passkeys in Microsoft Authenticator on your mobile device. With an Authenticator passkey, you can have seamless single sign-on (SSO) to other Microsoft native apps, like Teams or Outlook. For more information, see [How to enable passkeys in Microsoft Authenticator](#).

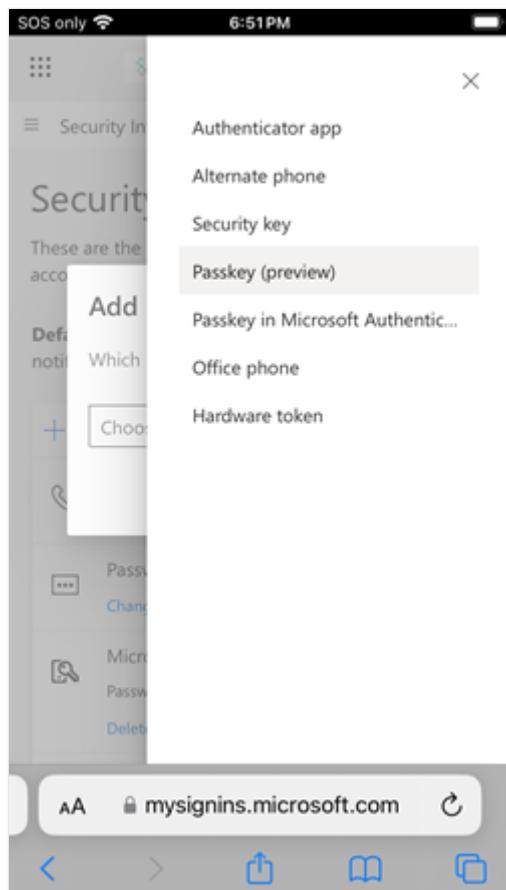
iOS

Register a passkey with iOS

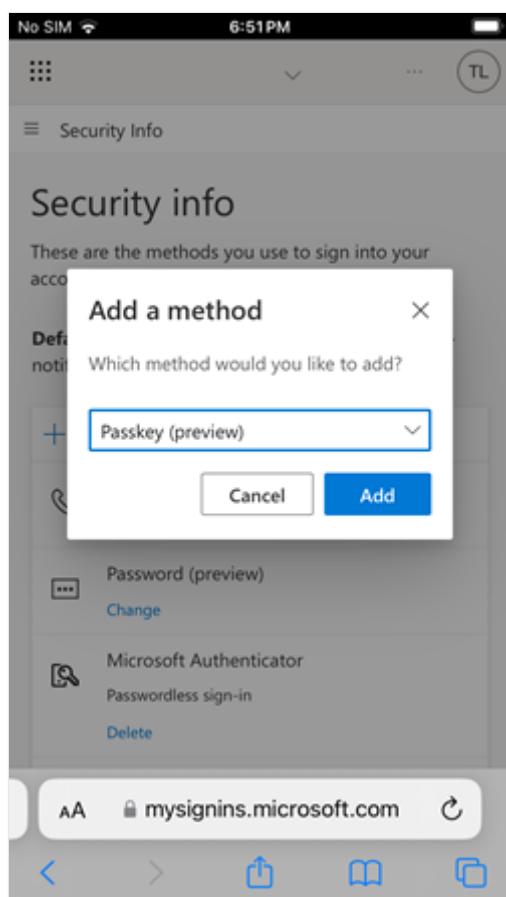
1. Using your iOS device, open a web browser and sign-in to [Security info](#).
2. Tap **+ Add sign-in method**.



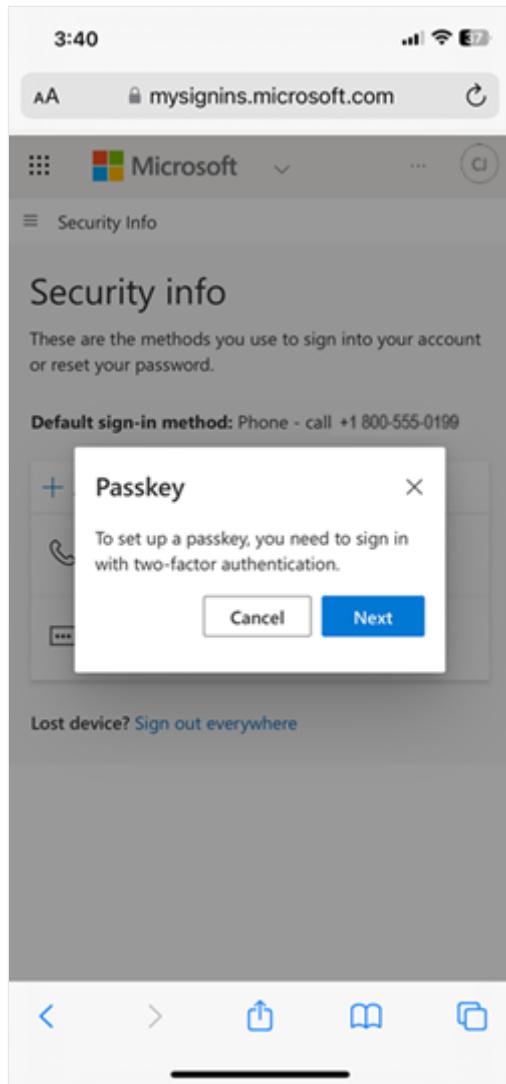
3. Select **Passkey**.



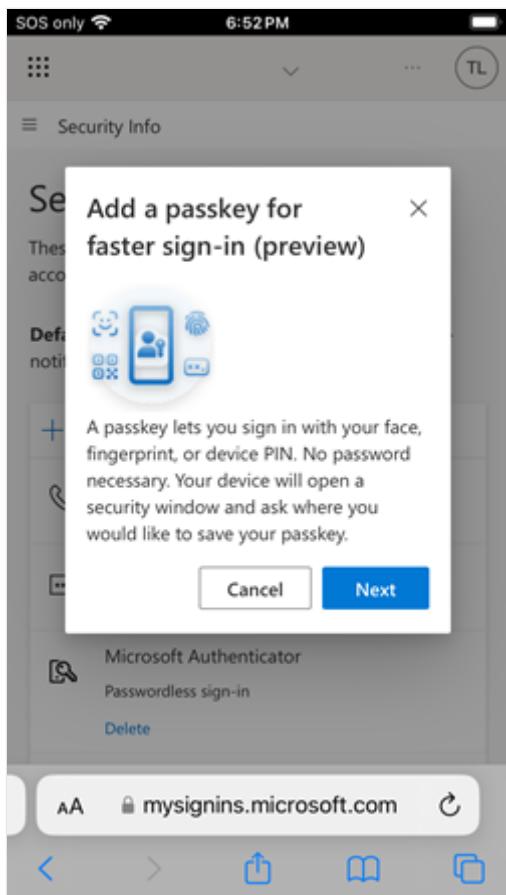
4. Tap Add.



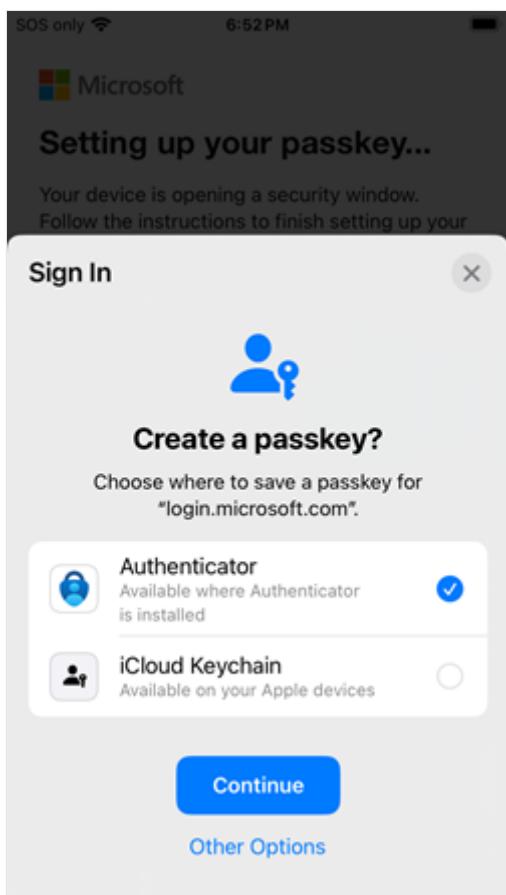
5. Sign in with multifactor authentication (MFA) before adding a passkey.



6. Select **Next**.



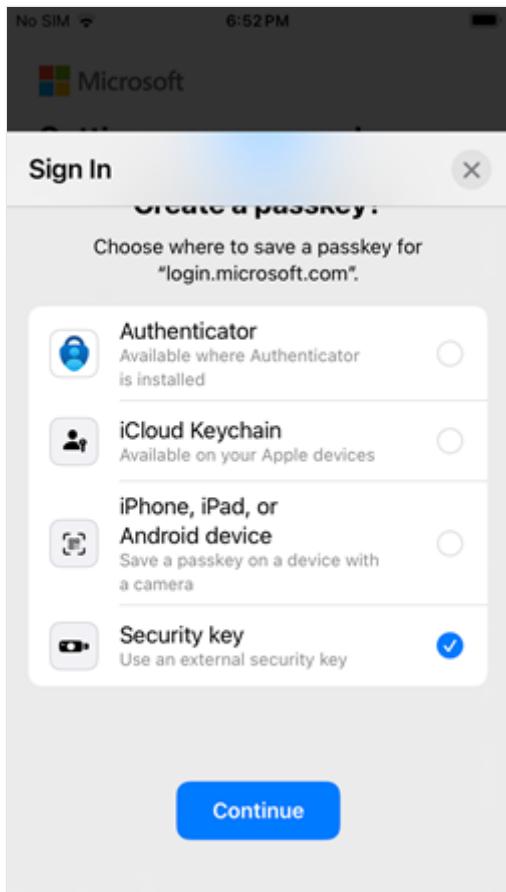
7. Your device opens a security window. Choose **Other Options**.



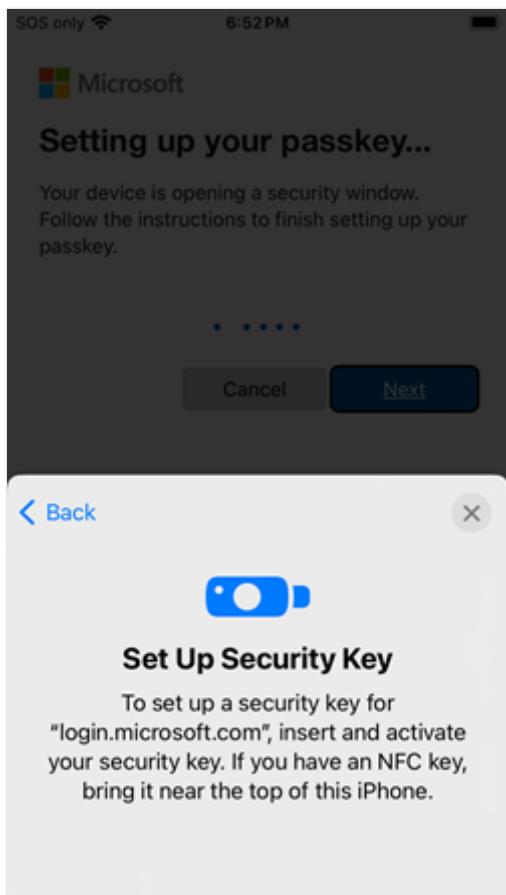
8. Select **Security key**.

! Note

Depending on the screen size and orientation of your iOS device, you may need to scroll down to see this option.



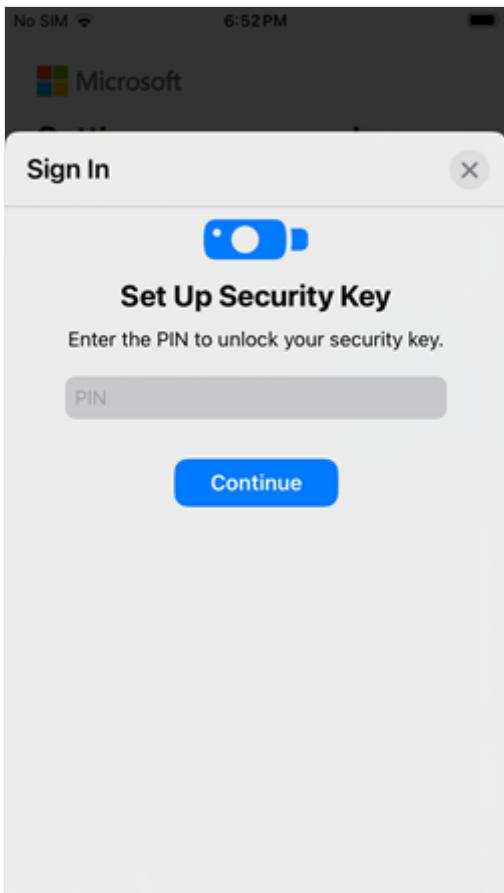
9. Connect your security key to your iOS device.



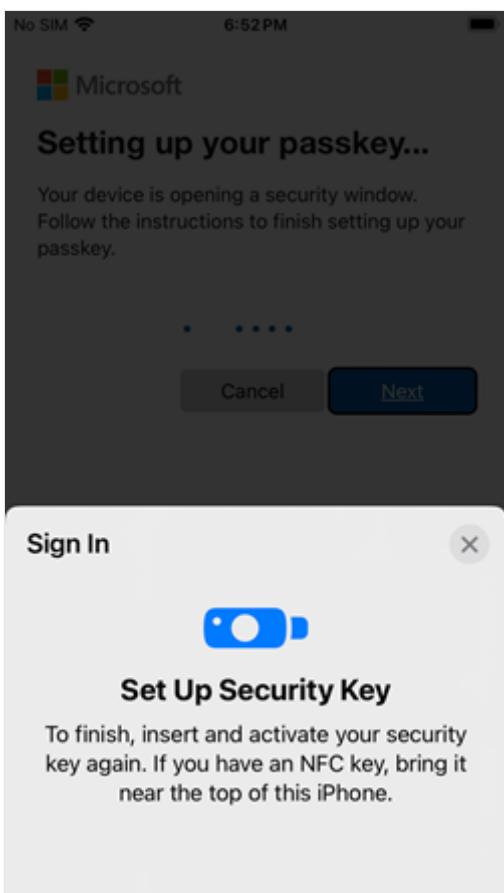
10. Provide your PIN or biometric.

(!) Note

If a PIN isn't configured for this security key, you need to first enroll a PIN before you continue registration.

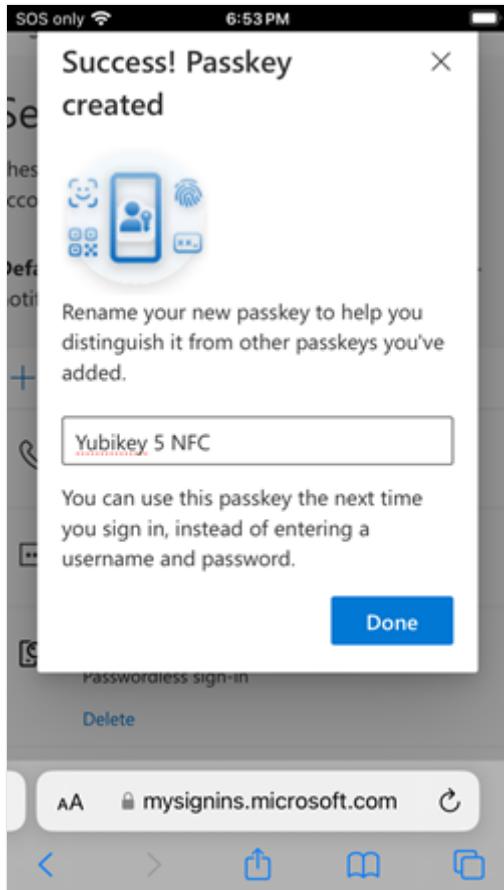


11. Reinsert or reconnect your security key to your iOS device.



12. Upon completion, you're redirected back to [Security info](#) and asked to rename your passkey. Name the passkey something memorable to you and

select Done.



Related content

- Choosing authentication methods for your organization

Feedback

Was this page helpful?

Yes

No

Provide product feedback ↗

Sign in with a passkey (FIDO2)

Article • 03/04/2025

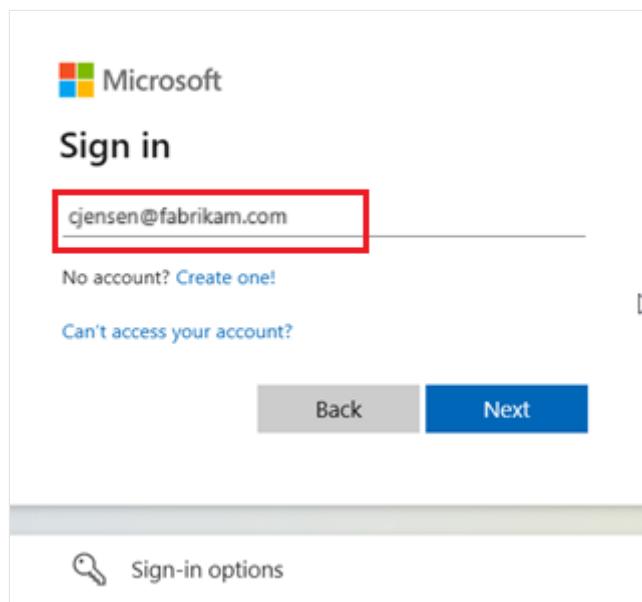
This article covers how users can sign in to Microsoft Entra ID with a passkey stored on a FIDO2 security key. For how to sign in with a passkey in Microsoft Authenticator, see [Sign in with passkeys in Authenticator for Android and iOS devices](#).

For more information on the availability of Microsoft Entra ID passkey (FIDO2) authentication across native apps, web browsers, and operating systems, see [Support for FIDO2 authentication with Microsoft Entra ID](#).

Sign in with a security key

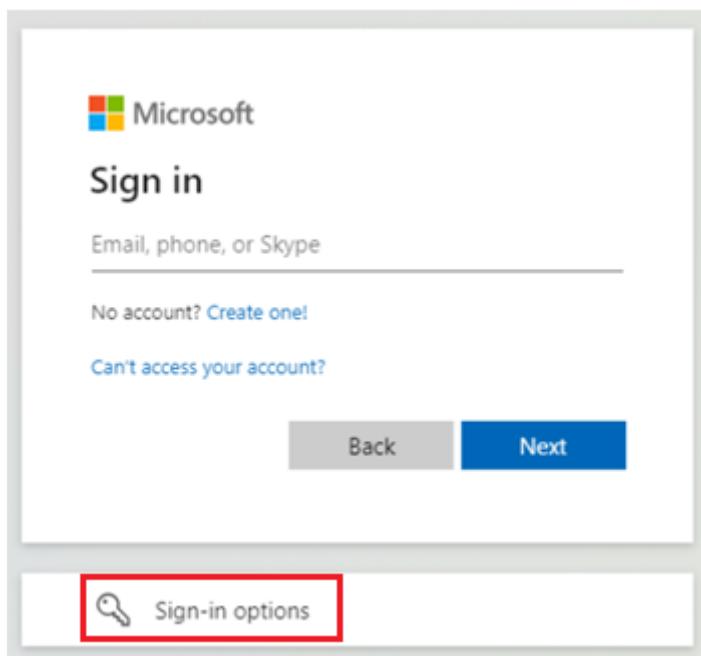
To sign into your work or school account with a security key, follow these steps on your device:

1. Go to [Office](#).
2. You can enter your username to sign in:

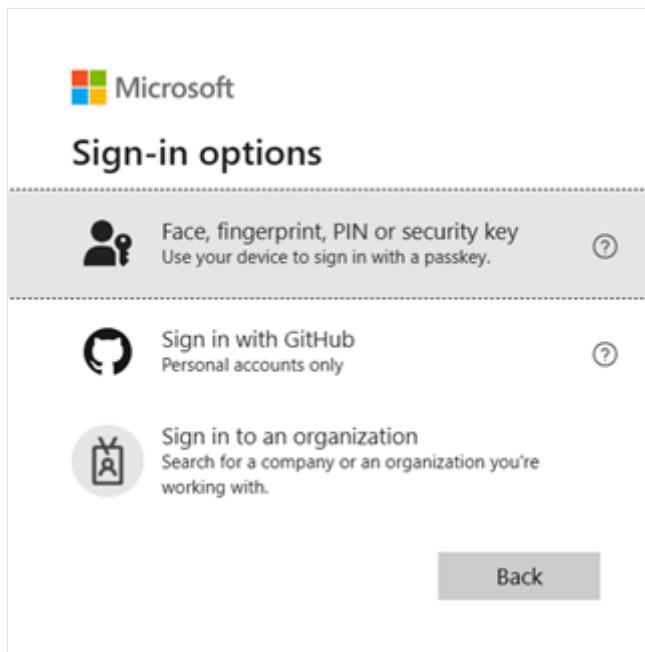


If you most recently used a passkey to sign in, you're automatically prompted to sign in with a passkey. Otherwise, select **Other ways to sign in**, and then select **Face, fingerprint, PIN, or security key**.

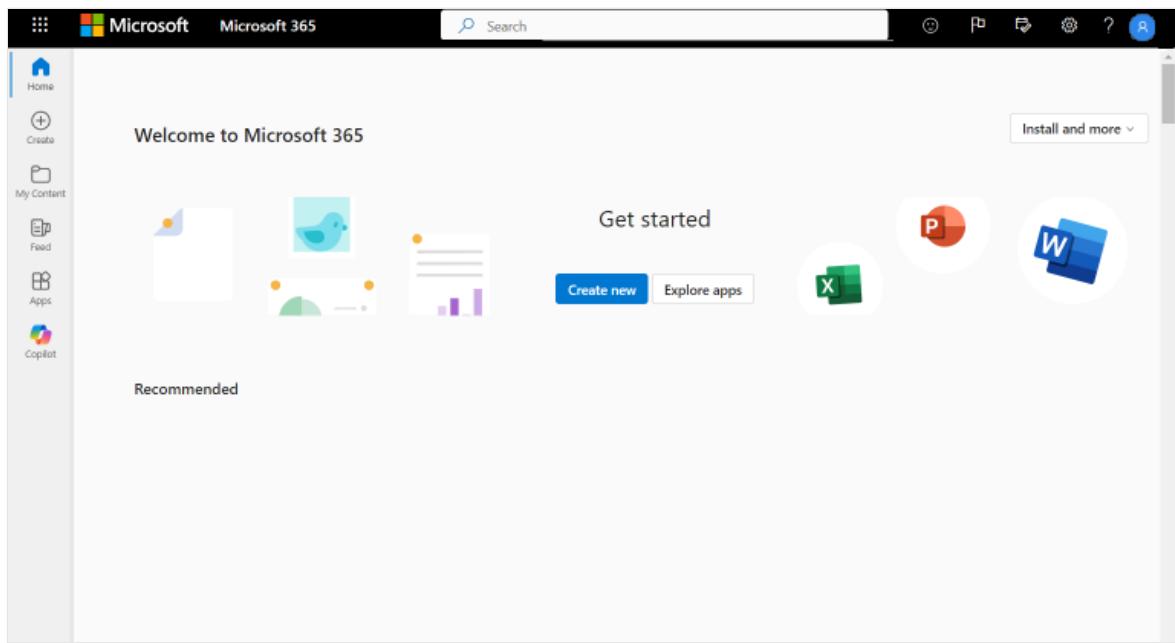
Alternatively, click **Sign-in options** to sign in more conveniently without having to enter a username.



If you chose **Sign-in options**, select Face, fingerprint, PIN, or security key. Otherwise, skip to next step.



3. Your device opens a security window. To use your security key, follow the steps in the operating system or browser dialog. Verify that it's you by scanning your fingerprint or entering your PIN.
4. Once you're signed in, your device displays a screen similar to this one:

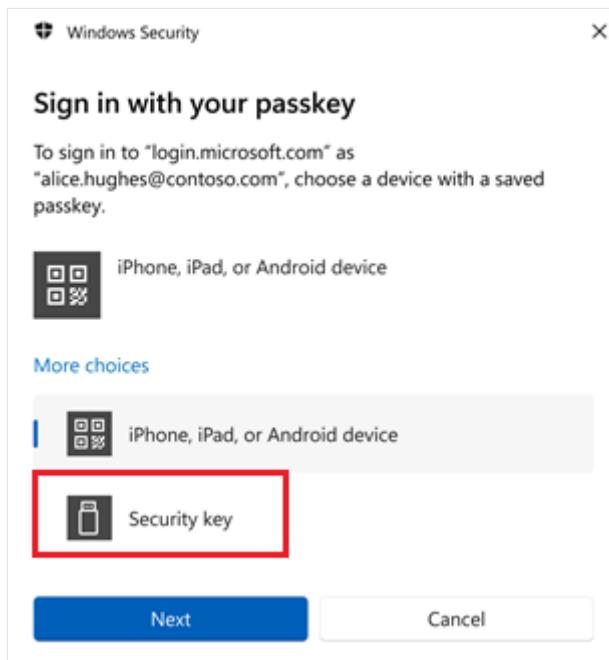


Known issues

Mobile device might be prioritized over security key

If you're using Chrome or Edge, the browser may prioritize using a passkey stored on a mobile device over a passkey stored on a security key.

- Beginning with Windows 11 version 23H2, the operating system shows the following prompt during sign-in. Below **More choices**, choose **Security key** and select **Next**.



- On earlier versions of Windows, the browser may show the QR pairing screen to continue with using a passkey stored on a mobile device. To use a passkey stored

on a security key instead, insert your security key and touch it to continue.

Use a passkey from another device?

Scan this QR code with the device that has the passkey you want to use for google.com



 If your passkey for google.com is on a USB security key, insert and touch it now

[Cancel](#)

Next steps

- Support for FIDO2 authentication with Microsoft Entra ID
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Passkey (FIDO2) authentication matrix with Microsoft Entra ID

Article • 05/12/2025

This article provides a comprehensive overview of passkey (FIDO2) authentication support in Microsoft Entra ID. It outlines compatibility across web browsers, native apps, and operating systems, enabling passwordless multifactor authentication. You'll also find platform-specific considerations, known issues, and guidance for third-party app and identity provider (IdP) support. Use this information to ensure seamless integration and optimal user experiences with passkeys in your environment.

For more information about how to sign in with FIDO2 security keys on a Windows device, see [Enable FIDO2 security key sign-in to Windows 10 and 11 devices with Microsoft Entra ID](#).

ⓘ Note

Microsoft Entra ID currently supports only device-bound passkeys stored on FIDO2 security keys or in Microsoft Authenticator. Microsoft is committed to securing customers and users with passkeys, and plans to support synced passkeys for Microsoft Entra ID.

Web browsers

The following section covers support for passkey (FIDO2) authentication in web browsers with Microsoft Entra ID.

⋮ Expand table

OS	Chrome	Edge	Firefox	Safari
Windows	✓	✓	✓	N/A
macOS	✓	✓	✓	✓
ChromeOS	✓	N/A	N/A	N/A
Linux	✓	✓	✓	N/A
iOS	✓	✓	✓	✓
Android	✓	✓	✗	N/A

Considerations for each platform

Windows

- Sign-in with security key requires one of the following items:
 - Windows 10 version 1903 or later
 - Chromium-based Microsoft Edge
 - Chrome 76 or later
 - Firefox 66 or later

macOS

- Sign-in with passkey requires macOS Catalina 11.1 or later with Safari 14 or later because Microsoft Entra ID requires user verification for multifactor authentication.
- Near-field communication (NFC) and Bluetooth Low Energy (BLE) security keys aren't supported on macOS by Apple.
- New security key registration doesn't work on these macOS browsers because they don't prompt to set up biometrics or PIN.
- See [Sign in when more than three passkeys are registered](#) for Safari on macOS.

ChromeOS

- NFC and BLE security keys aren't supported on ChromeOS by Google.
- Security key registration isn't supported on ChromeOS or Chrome browser.

Linux

- Sign-in with passkey in Microsoft Authenticator isn't supported in Firefox on Linux.

iOS

- Sign-in with passkey requires iOS 14.3 or later because Microsoft Entra ID requires user verification for multifactor authentication.
- BLE security keys aren't supported on iOS by Apple.
- NFC with FIPS 140-3 certified security keys isn't supported on iOS by Apple.
- New security key registration doesn't work on iOS browsers because they don't prompt to set up biometrics or PIN.
- See [Sign in when more than three passkeys are registered](#).

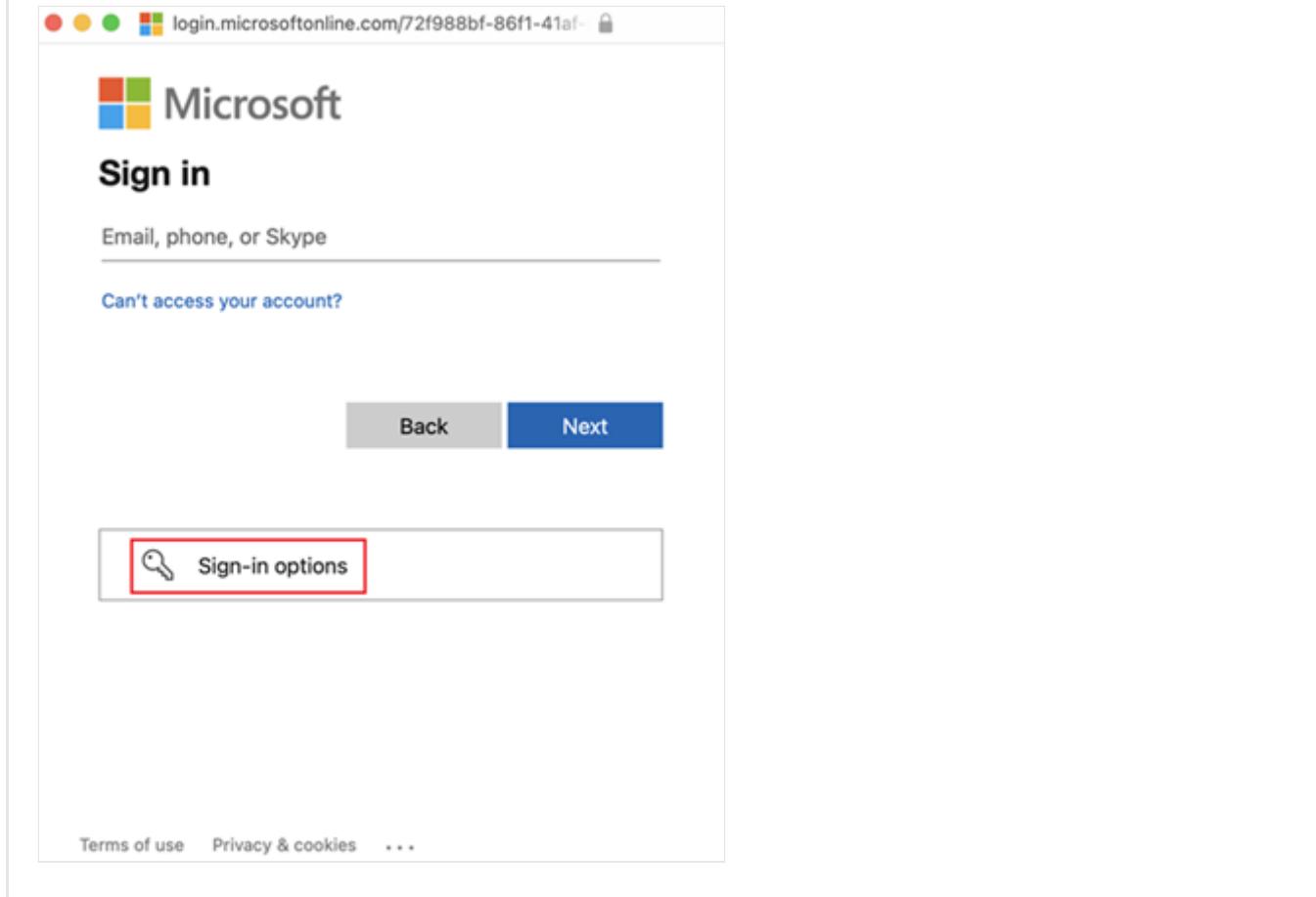
Android

- Sign-in with passkey requires Google Play Services 21 or later because Microsoft Entra ID requires user verification for multifactor authentication.
- BLE security keys aren't supported on Android by Google.
- Security key registration with Microsoft Entra ID isn't yet supported on Android.
- Sign-in with passkey isn't supported in Firefox on Android.

Known issues

Sign in when more than three passkeys are registered

If you registered more than three passkeys, sign in with a passkey might not work on iOS or Safari on macOS. If you have more than three passkeys, as a workaround, click **Sign-in options** and sign in without entering a username.



Next steps

[Enable passwordless security key sign-in](#)

Microsoft Entra ID attestation for FIDO2 security key vendors

Article • 05/13/2025

FIDO2 security keys enable phishing-resistant authentication. They can replace weak credentials with strong hardware-backed public/private-key credentials that can't be reused, replayed, or shared across services. Security keys support shared device scenarios, allowing you to carry your credential with you and safely authenticate on any supported device.

In Microsoft Entra ID Authentication methods policy, administrators can enforce attestation for FIDO2 security keys. When **Enforce attestation** is set to **Yes**, Microsoft requires extra metadata from FIDO2 security keys that are registered with the tenant. As a vendor, your FIDO2 security key is usable when attestation is enforced, if the following requirements are met.

(!) Note

Microsoft Entra ID currently supports device-bound passkeys stored on FIDO2 security keys and in Microsoft Authenticator. Microsoft is committed to securing customers and users with passkeys. We're investing in both synced and device-bound passkeys for work accounts.

Attestation requirements

Microsoft relies on the [FIDO Alliance Metadata Service \(MDS\)](#) to determine security key compatibility with Windows, Microsoft Edge browser, and online Microsoft accounts. Vendors report data to the FIDO MDS.

During FIDO2 registration, Microsoft Entra ID requires security keys to provide an attestation statement. For vendors, the expected attestation format is *packed*, as defined by [the FIDO standard](#).

The specific requirements vary based on how an administrator configures the FIDO2 authentication methods policy.

[+] Expand table

Enforce attestation set to Yes	Enforce attestation set to No
It must provide a valid <i>packed</i> attestation statement and a complete certificate that chains back to the attestation roots extracted from the FIDO Alliance	It must provide a valid <i>packed</i> attestation statement (but Microsoft will ignore attestation verification results) and a complete certificate

Enforce attestation set to Yes	Enforce attestation set to No
MDS, so that Microsoft can validate the key's metadata.	(which doesn't need to be associated with a particular certificate chain).

 **Note**

Vendors are responsible to publish all root attestation certificates to the FIDO Alliance MDS; otherwise, attestation verification can fail.

Additionally, if attestation is enforced, the following requirements apply:

- Your authenticator needs to have a FIDO2 certification. This can be at *any* level. To learn more about the certification, visit the FIDO Alliance Certification Overview [website](#).
- Your product metadata needs to be uploaded to the FIDO Alliance MDS, and you need to verify your metadata is in the MDS. The metadata must indicate that your authenticator supports:
 - FIDO 2.0 or higher.
 - User verification or client PIN - Microsoft Entra ID requires user verification with biometrics or PIN for all FIDO2 authentication attempts.
 - Resident keys (or discoverable credentials) - Resident keys are required for using a security key to sign in to Microsoft Entra ID without entering a username.
 - Hash-Based Message Authenticator Codes (HMAC) secret extension or Pseudo-Random Function (PRF) extension - An HMAC secret extension or PRF extension is required for using a security key to unlock Windows in offline scenarios.

Timelines

Microsoft ingests the latest version of the FIDO Alliance MDS every month. There may be a maximum four-week delay from the time that your FIDO2 security key appears in FIDO Alliance MDS to when Microsoft recognizes the key model. If your key meets the Microsoft attestation requirements, it automatically appears on the Microsoft FIDO2 partner page.

FIDO2 security keys eligible for attestation with Microsoft Entra ID

The following table includes each FIDO2 security key model listed in MDS version 156 that's eligible for attestation with Microsoft Entra ID. For each model, the table shows its Authenticator Attestation Globally Unique Identifier (AAGUID) and feature capabilities.

[\[+\] Expand table](#)

Description	AAGUID	Bio	USB	NFC	BLE
ACS FIDO Authenticator	50a45b0c-80e7-f944-bf29-f552bfa2e048	✗	✓	✗	✗
ACS FIDO Authenticator Card	973446ca-e21c-9a9b-99f5-9b985a67af0f	✗	✗	✓	✗
ACS FIDO Authenticator NFC	c89e6a38-6c00-5426-5aa5-c9cbf48f0382	✗	✓	✓	✗
Allthenticator Android App: roaming BLE FIDO2 Allthenticator for Windows, Mac, Linux, and Allthenticate door readers	5ca1ab1e-fa57-1337-f1d0-a117371ca702	✓	✓	✗	✗
Allthenticator iOS App: roaming BLE FIDO2 Allthenticator for Windows, Mac, Linux, and Allthenticate door readers	5ca1ab1e-1337-fa57-f1d0-a117e71ca702	✓	✓	✗	✗
Android Authenticator with SafetyNet Attestation	b93fd961-f2e6-462f-b122-82002247de78	✓	✗	✗	✗
Arculus FIDO 2.1 Key Card [P71]	3f59672f-20aa-4afe-b6f4-7e5e916b6d98	✗	✓	✗	✗
Arculus FIDO2/U2F Key Card	9d3df6ba-282f-11ed-a261-0242ac120002	✗	✓	✗	✗
ATKey.Card CTAP2.0	d41f5a69-b817-4144-a13c-9ebd6d9254d6	✓	✗	✗	✗
ATKey.Card NFC	da1fa263-8b25-42b6-a820-c0036f21ba7f	✓	✓	✓	✗
ATKey.Pro CTAP2.0	e1a96183-5016-4f24-b55b-e3ae23614cc6	✓	✗	✗	✗
ATKey.Pro CTAP2.1	e416201b-afeb-41ca-a03d-2281c28322aa	✓	✓	✗	✗
ATKey.ProS	ba76a271-6eb6-4171-874d-b6428dbe3437	✓	✓	✗	✗
Atos CardOS FIDO2	1c086528-58d5-f211-823c-356786e36140	✗	✓	✓	✗
authenton1 - CTAP2.1	b267239b-954f-4041-a01b-ee4f33c145b6	✗	✓	✓	✗

Description	AAGUID	Bio	USB	NFC	BLE
CardOS FIDO2 Token	8da0e4dc-164b-454e-972e-88f362b23d59	✗	✓	✓	✗
Chunghwa Telecom FIDO2 Smart Card Authenticator	175cd298-83d2-4a26-b637-313c07a6434e	✗	✗	✓	✗
Crayonic KeyVault K1 (USB-NFC-BLE FIDO2 Authenticator)	be727034-574a-f799-5c76-0929e0430973	✓	✓	✓	✓
Cryptnox FIDO2	9c835346-796b-4c27-8898-d6032f515cc5	✗	✗	✓	✗
Cryptnox FIDO2.1	1d1b4e33-76a1-47fb-97a0-14b10d0933f1	✗	✗	✓	✗
Dapple Authenticator from Dapple Security Inc.	6dae43be-af9c-417b-8b9f-1b611168ec60	✗	✗	✗	✗
Deepnet SafeKey/Classic (USB)	b9f6b7b6-f929-4189-bca9-dd951240c132	✗	✗	✗	✗
Egomet FIDO2 Authenticator for Android	1105e4ed-af1d-02ff-ffff-ffffffffffff	✓	✗	✗	✗
ellipticSecure MIRkey USB Authenticator	eb3b131e-59dc-536a-d176-cb7306da10f5	✗	✓	✗	✗
Ensurity AUTH BioPro	454e5346-4944-4ffd-6c93-8e9267193e9b	✓	✓	✗	✗
Ensurity ThinC	454e5346-4944-4ffd-6c93-8e9267193e9a	✓	✓	✗	✗
ESS Smart Card Inc. Authenticator	5343502d-5343-5343-6172-644649444f32	✗	✗	✓	✗
eToken Fusion FIPS	050dd0bc-ff20-4265-8d5d-305c4b215192	✗	✓	✗	✗
eToken Fusion NFC FIPS	10c70715-2a9a-4de1-b0aa-3cff6d496d39	✗	✗	✓	✗
eToken Fusion NFC PIV	146e77ef-11eb-4423-b847-ce77864e9411	✗	✗	✓	✗
eToken Fusion NFC PIV Enterprise	c3f47802-de73-4dfc-ba22-671fe3304f90	✗	✗	✓	✗
eWBM eFA310 FIDO2 Authenticator	95442b2e-f15e-4def-b270-efb106facb4e	✓	✗	✗	✗

Description	AAGUID	Bio	USB	NFC	BLE
eWBM eFA320 FIDO2 Authenticator	87dbc5a1-4c94-4dc8-8a47-97d800fd1f3c	✓	✗	✗	✗
eWBM eFA500 FIDO2 Authenticator	361a3082-0278-4583-a16f-72a527f973e4	✓	✗	✗	✗
eWBM eFPA FIDO2 Authenticator	61250591-b2bc-4456-b719-0b17be90bb30	✓	✗	✗	✗
Excelsecu eSecu FIDO2 Fingerprint Key	6002f033-3c07-ce3e-d0f7-0ffe5ed42543	✓	✓	✗	✗
Excelsecu eSecu FIDO2 Fingerprint Security Key	d384db22-4d50-ebde-2eac-5765cf1e2a44	✓	✓	✗	✗
Excelsecu eSecu FIDO2 Fingerprint Security Key	20f0be98-9af9-986a-4b42-8eca4acb28e4	✓	✓	✗	✗
Excelsecu eSecu FIDO2 NFC Security Key	fbefdf68-fe86-0106-213e-4d5fa24cbe2e	✗	✓	✓	✗
Excelsecu eSecu FIDO2 NFC Security Key	a3975549-b191-fd67-b8fb-017e2917fdb3	✗	✓	✓	✗
Excelsecu eSecu FIDO2 Pro Security Key	0d9b2e56-566b-c393-2940-f821b7f15d6d	✗	✓	✓	✓
Excelsecu eSecu FIDO2 PRO Security Key	bbf4b6a7-679d-f6fc-c4f2-8ac0ddf9015a	✗	✓	✓	✓
Excelsecu eSecu FIDO2 Security Key	cdbdaea2-c415-5073-50f7-c04e968640b6	✗	✓	✗	✗
Feitian AllinOne FIDO2 Authenticator	12ded745-4bed-47d4-abaa-e713f51d6393	✓	✓	✓	✓
Feitian BioPass FIDO2 Authenticator	77010bd7-212a-4fc9-b236-d2ca5e9d4084	✓	✓	✗	✗
Feitian BioPass FIDO2 Plus (Enterprise Profile)	a02140b7-0cbd-42e1-a9b5-a39da2545114	✓	✓	✗	✗
Feitian BioPass FIDO2 Plus Authenticator	b6ede29c-3772-412c-8a78-539c1f4c62d2	✓	✓	✗	✗
Feitian BioPass FIDO2 Plus Authenticator	42df17de-06ba-4177-a2bb-6701be1380d6	✓	✓	✗	✗
Feitian BioPass FIDO2 Pro (Enterprise Profile)	2bfff89f2-323a-48fc-b7c8-9ff7fe87c07e	✓	✓	✗	✗

Description	AAGUID	Bio	USB	NFC	BLE
Feitian BioPass FIDO2 Pro Authenticator	4c0cf95d-2f40-43b5-ba42-4c83a11c04ba	✓	✓	✗	✗
Feitian ePass FIDO Authenticator (CTAP2.1, CTAP2.0, U2F)	12755c32-8ad1-46eb-881c-e0b38d848b09	✗	✓	✗	✗
Feitian ePass FIDO2 Authenticator	833b721a-ff5f-4d00-bb2e-bdda3ec01e29	✗	✓	✗	✗
Feitian ePass FIDO2-NFC Authenticator	ee041bce-25e5-4cdb-8f86-897fd6418464	✗	✓	✓	✗
Feitian ePass FIDO2-NFC Plus Authenticator	260e3021-482d-442d-838c-7edfbe153b7e	✗	✓	✓	✗
Feitian ePass FIDO2-NFC Series (CTAP2.1, CTAP2.0, U2F)	234cd403-35a2-4cc2-8015-77ea280c77f5	✗	✓	✓	✗
Feitian ePass FIDO-NFC (Enterprise Profile) (CTAP2.1, CTAP2.0, U2F)	39589099-9a75-49fc-afaa-801ca211c62a	✗	✓	✓	✗
Feitian ePass FIDO-NFC(CTAP2.1, CTAP2.0, U2F)	78ba3993-d784-4f44-8d6e-cc0a8ad5230e	✗	✓	✓	✗
Feitian FIDO Smart Card	2c0df832-92de-4be1-8412-88a8f074df4a	✗	✗	✓	✗
Feitian iePass FIDO Authenticator	3e22415d-7fdf-4ea4-8a0c-dd60c4249b9d	✗	✓	✗	✗
FIDO Alliance TruU Sample FIDO2 Authenticator	ca87cb70-4c1b-4579-a8e8-4efdd7c007e0	✓	✓	✗	✗
FIDO KeyPass S3	f4c63eff-d26c-4248-801c-3736c7eaa93a	✗	✓	✗	✗
Foongtone FIDO Authenticator	46544d5d-8f5d-4db4-89ac-ea8977073fff	✗	✗	✓	✗
FT-JCOS FIDO Fingerprint Card	8c97a730-3f7b-41a6-87d6-1e9b62bda6f0	✗	✗	✓	✗
GoldKey Security Token	0db01cd6-5618-455b-bb46-1ec203d3213e	✗	✓	✓	✗
Google Titan Security Key v2	42b4fb4a-2866-43b2-9bf7-6c6669c2e5d3	✗	✓	✓	✗
GoTrust Idem Card FIDO2 Authenticator	9f0d8150-baa5-4c00-9299-ad62c8bb4e87	✗	✗	✗	✗

Description	AAGUID	Bio	USB	NFC	BLE
GoTrust Idem Key FIDO2 Authenticator	3b1adb99-0dfe-46fd-90b8-7f7614a4de2a	✗	✗	✗	✗
GSTAG OAK FIDO2 Authenticator	773c30d9-5919-4e96-a4f5-db65e95cf890	✗	✗	✓	✗
HID Crescendo 4000	2a55aee6-27cb-42c0-bc6e-04efe999e88a	✗	✗	✓	✗
HID Crescendo C2300	aeb6569c-f8fb-4950-ac60-24ca2bbe2e52	✗	✗	✓	✗
HID Crescendo C3000	c80dbd9a-533f-4a17-b941-1a2f1c7cedff	✗	✗	✓	✗
HID Crescendo Enabled	54d9fee8-e621-4291-8b18-7157b99c5bec	✗	✗	✓	✗
HID Crescendo Fusion	c4ddaf11-3032-4e77-b3b9-3a340369b9ad	✗	✗	✓	✗
HID Crescendo Key	692db549-7ae5-44d5-a1e5-dd20a493b723	✗	✓	✓	✗
HID Crescendo Key V2	2d3bec26-15ee-4f5d-88b2-53622490270b	✗	✓	✓	✗
HID Crescendo Key V3	7991798a-a7f3-487f-98c0-3faf7a458a04	✗	✓	✓	✗
Hideez Key 3 FIDO2	3e078ffd-4c54-4586-8baa-a77da113aec5	✗	✗	✗	✓
Hideez Key 4 FIDO2 SDK	4e768f2c-5fab-48b3-b300-220eb487752b	✗	✓	✓	✓
Hyper FIDO Bio Security Key	d821a7d4-e97c-4cb6-bd82-4237731fd4be	✓	✗	✗	✗
Hyper FIDO Pro	9f77e279-a6e2-4d58-b700-31e5943c6a98	✗	✗	✗	✗
HYPR FIDO2 Authenticator	0076631b-d4a0-427f-5773-0ec71c9e0279	✓	✗	✗	✗
IDCore 3121 Fido	e86addcd-7711-47e5-b42a-c18257b0bf61	✗	✗	✓	✗
IDEMEDIA ID-ONE Card	8d1b1fcf-3c76-49a9-9129-5515b346aa02	✗	✓	✓	✗

Description	AAGUID	Bio	USB	NFC	BLE
IDmelon Android Authenticator	39a5647e-1853-446ca1f6-a79bae9f5bc7	✓	✗	✗	✗
IDmelon iOS Authenticator	820d89ed-d65a-409e-85cb-f73f0578f82a	✓	✗	✗	✗
ID-One Card	bb405265-40cf-4115-93e5-a332c1968d8c	✗	✗	✓	✗
ID-One Key	82b0a720-127a-4788-b56d-d1d4b2d82eac	✗	✓	✓	✗
ID-One Key	f2145e86-211e-4931-b874-e22bba7d01cc	✗	✓	✓	✗
IDPrime 3930 FIDO	ca4cff1b-5a81-4404-8194-59aabcf1660b	✗	✗	✓	✗
IDPrime 3940 FIDO	b50d5e0a-7f81-4959-9b12-f45407407503	✗	✗	✓	✗
IDPrime 931 Fido	2194b428-9397-4046-8f39-007a1605a482	✗	✗	✓	✗
IDPrime 941 Fido	2ffd6452-01da-471f-821b-ea4bf6c8676a	✗	✗	✓	✗
IIST FIDO2 Authenticator	4b89f401-464e-4745-a520-486ddfc5d80e	✗	✓	✗	✗
ImprovID Authenticator	4c50ff10-1057-4fc6-b8ed-43a529530c3c	✗	✓	✓	✗
KEY-ID FIDO2 Authenticator	d91c5288-0ef0-49b7-b8ae-21ca0aa6b3f3	✗	✓	✗	✗
KeyVault Secp256R1 FIDO2 CTAP2 Authenticator	d61d3b87-3e7c-4aea-9c50-441c371903ad	✓	✗	✗	✗
KeyXentic FIDO2 Secp256R1 FIDO2 CTAP2 Authenticator	4b3f8944-d4f2-4d21-bb19-764a986ec160	✓	✓	✗	✗
KeyXentic FIDO2 Secp256R1 FIDO2 CTAP2 Authenticator	ec31b4cc-2acc-4b8e-9c01-bade00ccb26	✓	✓	✗	✗
KONAI Secp256R1 FIDO2 Conformance Testing CTAP2 Authenticator	f7c558a0-f465-11e8-b568-0800200c9a66	✓	✓	✓	✗
KX701 SmartToken FIDO	fec067a1-f1d0-4c5eb4c0-cc3237475461	✗	✓	✓	✗

Description	AAGUID	Bio	USB	NFC	BLE
Ledger Nano S FIDO2 Authenticator	341e4da9-3c2e-8103-5a9f-aad887135200	✗	✓	✗	✗
Ledger Nano S Plus FIDO2 Authenticator	58b44d0b-0a7c-f33a-fd48-f7153c871352	✗	✓	✗	✗
Ledger Nano X FIDO2 Authenticator	fcb1bcb4-f370-078c-6993-bc24d0ae3fbe	✗	✓	✗	✗
NEOWAVE Badgeo FIDO2	c5703116-972b-4851-a3e7-ae1259843399	✗	✓	✓	✗
NEOWAVE Winkeo FIDO2	3789da91-f943-46bc-95c3-50ea2012f03a	✗	✓	✗	✗
Nitrokey 3 AM	2cd2f727-f6ca-44da-8f48-5c2e5da000a2	✗	✓	✗	✗
NXP Semiconductros FIDO2 Conformance Testing CTAP2 Authenticator	07a9f89c-6407-4594-9d56-621d5f1e358b	✗	✗	✗	✗
Nymi FIDO2 Authenticator	0acf3011-bc60-f375-fb53-6f05f43154e0	✓	✗	✓	✗
OCTATCO EzFinger2 FIDO2 AUTHENTICATOR	a1f52be5-dfab-4364-b51c-2bd496b14a56	✓	✗	✗	✗
OCTATCO EzQuant FIDO2 AUTHENTICATOR	bc2fe499-0d8e-4ffe-96f3-94a82840cf8c	✓	✓	✗	✗
OneKey FIDO2 Authenticator	69e7c36f-f2f6-9e0d-07a6-bcc243262e6b	✗	✓	✗	✗
OneKey FIDO2 Bluetooth Authenticator	70e7c36f-f2f6-9e0d-07a6-bcc243262e6b	✗	✓	✗	✓
OneSpan DIGIPASS FX1 BIO	30b5035e-d297-4ff1-b00b-addc96ba6a98	✓	✓	✗	✓
OneSpan DIGIPASS FX1a	30b5035e-d297-4ff1-010b-addc96ba6a98	✓	✓	✓	✗
OneSpan DIGIPASS FX7	30b5035e-d297-4ff7-b00b-addc96ba6a98	✗	✓	✗	✗
OneSpan FIDO Touch	30b5035e-d297-4fc1-b00b-addc96ba6a97	✗	✓	✗	✓
OnlyKey Secp256R1 FIDO2 CTAP2 Authenticator	998f358b-2dd2-4cbe-a43a-e8107438dfb3	✗	✗	✗	✗

Description	AAGUID	Bio	USB	NFC	BLE
OpenSK authenticator	664d9f67-84a2-412a-9ff7-b4f7d8ee6d05	✗	✓	✗	✗
Pone Biometrics OFFPAD Authenticator	69700f79-d1fb-472e-bd9b-a3a3b9a9eda0	✓	✗	✗	✓
Pone Biometrics OFFPAD Authenticator	09591fc6-9811-48f7-8f57-b9f23df6413f	✓	✗	✓	✓
Precision InnALT Key FIDO 2 Level 2 certified	88bbd2f0-342a-42e7-9729-dd158be5407a	✓	✓	✗	✗
RSA Authenticator 4 for Android	59f85fe7-faa5-4c92-9f52-697b9d4d5473	✓	✗	✗	✗
RSA Authenticator 4 for iOS	8681a073-5f50-4d52-bce4-e21658d207b3	✓	✗	✗	✗
RSA DS100	7e3f3d30-3557-4442-bdae-139312178b39	✗	✓	✗	✗
SafeKey/Classic (FP)	e41b42a3-60ac-4afb-8757-a98f2d7f6c9f	✓	✗	✗	✗
SafeKey/Classic (NFC)	b12eac35-586c-4809-a4b1-d81af6c305cf	✗	✗	✗	✗
Safenet eToken FIDO	efb96b10-a9ee-4b6c-a4a9-d32125ccd4a4	✗	✓	✗	✗
SafeNet eToken Fusion	74820b05-a6c9-40f9-8fb0-9f86aca93998	✗	✓	✗	✗
SafeNet eToken Fusion CC	23786452-f02d-4344-87ed-aaf703726881	✗	✓	✗	✗
Samsung Pass	53414d53-554e-4700-0000-000000000000	✓	✗	✗	✗
Security Key by Yubico	f8a011f3-8c0a-4d15-8006-17111f9edc7d	✗	✓	✗	✗
Security Key by Yubico	b92c3f9a-c014-4056-887f-140a2501163b	✗	✓	✗	✗
Security Key by Yubico with NFC	149a2021-8ef6-4133-96b8-81f8d5b7f1f5	✗	✓	✓	✗
Security Key by Yubico with NFC	6d44ba9b-f6ec-2e49-b930-0c8fe920cb73	✗	✓	✓	✗

Description	AAGUID	Bio	USB	NFC	BLE
Security Key NFC by Yubico	b7d3f68e-88a6-471e-9ecf-2df26d041ede	✗	✓	✓	✗
Security Key NFC by Yubico	a4e9fc6d-4cbe-4758-b8ba-37598bb5bbaa	✗	✓	✓	✗
Security Key NFC by Yubico	e77e3c64-05e3-428b-8824-0cbeb04b829d	✗	✓	✗	✗
Security Key NFC by Yubico - Enterprise Edition	47ab2fb4-66ac-4184-9ae1-86be814012d5	✗	✓	✗	✗
Security Key NFC by Yubico - Enterprise Edition	ed042a3a-4b22-4455-bb69-a267b652ae7e	✗	✓	✓	✗
Security Key NFC by Yubico - Enterprise Edition	0bb43545-fd2c-4185-87dd-feb0b2916ace	✗	✓	✓	✗
Security Key NFC by Yubico - Enterprise Edition (Enterprise Profile)	9ff4cc65-6154-4fff-ba09-9e2af7882ad2	✗	✓	✗	✗
Security Key NFC by Yubico - Enterprise Edition (Enterprise Profile)	72c6b72d-8512-4c66-8359-9d3d10d9222f	✗	✓	✓	✗
Security Key NFC by Yubico - Enterprise Edition Preview	2772ce93-eb4b-4090-8b73-330f48477d73	✗	✓	✗	✗
Security Key NFC by Yubico Preview	760eda36-00aa-4d29-855b-4012a182cdeb	✗	✓	✗	✗
Sentry Enterprises CTAP2 Authenticator	89b19028-256b-4025-8872-255358d950e4	✓	✓	✗	✓
SHALO AUTH	57235694-51a5-4a4d-a81a-f42185df6502	✗	✓	✗	✗
SmartDisplay BobeePass FIDO2 Authenticator	516d3969-5a57-5651-5958-4e7a49434167	✓	✓	✓	✓
Solo Secp256R1 FIDO2 CTAP2 Authenticator	8876631b-d4a0-427f-5773-0ec71c9e0279	✗	✗	✗	✗
Solo Tap Secp256R1 FIDO2 CTAP2 Authenticator	8976631b-d4a0-427f-5773-0ec71c9e0279	✗	✗	✓	✗
Somu Secp256R1 FIDO2 CTAP2 Authenticator	9876631b-d4a0-427f-5773-0ec71c9e0279	✗	✗	✗	✗
Swissbit iShield Key 2	7787a482-13e8-4784-8a06-c7ed49a7aaf4	✗	✓	✓	✗

Description	AAGUID	Bio	USB	NFC	BLE
Swissbit iShield Key 2 Enterprise	e400ef8c-711d-4692-af46-7f2cf7da23ad	✗	✓	✓	✗
Swissbit iShield Key 2 FIPS	817cdab8-0d51-4de1-a821-e25b88519cf3	✗	✓	✓	✗
Swissbit iShield Key 2 FIPS Enterprise	5eaff75a-dd43-451f-af9f-87c9eeae293e	✗	✓	✓	✗
Swissbit iShield Key FIDO2	931327dd-c89b-406c-a81e-ed7058ef36c6	✗	✓	✗	✗
Swissbit iShield Key Pro	5d629218-d3a5-11ed-afa1-0242ac120002	✗	✓	✓	✗
Taglio CTAP2.1 CS	092277e5-8437-46b5-b911-ea64b294acb7	✗	✗	✓	✗
Taglio CTAP2.1 EP	7d2afadd-bf6b-44a2-a66b-e831fce8eff	✗	✗	✓	✗
Thales IDPrime FIDO Bio	4d41190c-7beb-4a84-8018-adf265a6352d	✓	✗	✓	✗
Token Ring 3 FIDO2 Authenticator	c62100de-759b-4bf8-b22b-63b3e3a80401	✓	✗	✓	✗
Token Ring FIDO2 Authenticator	91ad6b93-264b-4987-8737-3a690cad6917	✓	✗	✓	✗
TOKEN2 FIDO2 Security Key	ab32f0c6-2239-afbb-c470-d2ef4e254db7	✗	✗	✗	✗
TOKEN2 PIN Plus Security Key Series	eabb46cc-e241-80bf-ae9e-96fa6d2975cf	✗	✓	✓	✗
TruU Windows Authenticator	95e4d58c-056e-4a65-866d-f5a69659e880	✗	✗	✗	✗
TruU Windows Authenticator	ba86dc56-635f-4141-aef6-00227b1b9af6	✗	✗	✗	✗
T-Shield TrustSec FIDO2 Bio and client PIN version	882adaf5-3aa9-4708-8e7d-3957103775b4	✓	✓	✓	✗
USB/NFC Passcode Authenticator	fcfb13a2-244f-4b36-9077-82b79d6a7de7	✗	✓	✓	✗
uTrust FIDO2 Security Key	73402251-f2a8-4f03-873e-3cb6db604b03	✗	✓	✓	✗

Description	AAGUID	Bio	USB	NFC	BLE
VALMIDO PRO FIDO	5626bed4-e756-430b-a7ff-ca78c8b12738	✓	✗	✗	✓
Veridium Android SDK	5ea308b2-7ac7-48b9-ac09-7e2da9015f8c	✓	✗	✗	✓
Veridium iOS SDK	6e8d1eae-8d40-4c25-bcf8-4633959afc71	✓	✗	✗	✓
VeridiumID Passkey Android SDK	8d4378b0-725d-4432-b3c2-01fcdaf46286	✓	✗	✗	✓
VeridiumID Passkey iOS SDK	1e906e14-77af-46bc-ae9f-fe6ef18257e4	✓	✗	✗	✓
VeriMark Guard Fingerprint Key	d94a29d9-52dd-4247-9c2d-8b818b610389	✓	✗	✗	✗
VeroCard FIDO2 Authenticator	99ed6c29-4573-4847-816d-78ad8f1c75ef	✗	✗	✗	✓
VinCSS FIDO2 Authenticator	5fdb81b8-53f0-4967-a881-f5ec26fe4d18	✗	✗	✗	✗
VivoKey Apex FIDO2	d7a423ad-3e19-4492-9200-78137dccc136	✗	✗	✓	✗
WinMagic FIDO Eazy - Phone	f56f58b3-d711-4afc-ba7d-6ac05f88cb19	✓	✗	✗	✗
WinMagic FIDO Eazy - Software	31c3f7ff-bf15-4327-83ec-9336abcbcd34	✗	✗	✗	✗
WinMagic FIDO Eazy - TPM	970c8d9c-19d2-46af-aa32-3f448db49e35	✗	✗	✗	✗
WiSECURE AuthTron USB FIDO2 Authenticator	504d7149-4e4c-3841-4555-55445a677357	✓	✓	✗	✗
WiSECURE Blentity FIDO2 Authenticator	5753362b-4e6b-6345-7b2f-255438404c75	✗	✓	✗	✗
YubiKey 5 FIPS Series	73bb0cd4-e502-49b8-9c6f-b59445bf720b	✗	✓	✗	✗
YubiKey 5 FIPS Series	57f7de54-c807-4eab-b1c6-1c9be7984e92	✗	✓	✗	✗
YubiKey 5 FIPS Series (Enterprise Profile)	905b4cb4-ed6f-4da9-92fc-45e0d4e9b5c7	✗	✓	✗	✗

Description	AAGUID	Bio	USB	NFC	BLE
YubiKey 5 FIPS Series (RC Preview)	d2fdbd093-ee62-488d-9dad-1e36389f8826	✗	✓	✗	✗
YubiKey 5 FIPS Series with Lightning	85203421-48f9-4355-9bc8-8a53846e5083	✗	✓	✗	✗
YubiKey 5 FIPS Series with Lightning	7b96457d-e3cd-432b-9ceb-c9fdd7ef7432	✗	✓	✗	✗
YubiKey 5 FIPS Series with Lightning (Enterprise Profile)	3a662962-c6d4-4023-bebb-98ae92e78e20	✗	✓	✗	✗
YubiKey 5 FIPS Series with Lightning (RC Preview)	9e66c661-e428-452a-a8fb-51f7ed088acf	✗	✓	✗	✗
YubiKey 5 FIPS Series with Lightning Preview	5b0e46ba-db02-44ac-b979-ca9b84f5e335	✗	✓	✗	✗
YubiKey 5 FIPS Series with NFC	fcc0118f-cd45-435b-8da1-9782b2da0715	✗	✓	✓	✗
YubiKey 5 FIPS Series with NFC	c1f9a0bc-1dd2-404a-b27f-8e29047a43fd	✗	✓	✓	✗
YubiKey 5 FIPS Series with NFC (Enterprise Profile)	79f3c8ba-9e35-484b-8f47-53a5a0f5c630	✗	✓	✓	✗
YubiKey 5 FIPS Series with NFC (RC Preview)	ce6bf97f-9f69-4ba7-9032-97adc6ca5cf1	✗	✓	✓	✗
YubiKey 5 FIPS Series with NFC Preview	62e54e98-c209-4df3-b692-de71bb6a8528	✗	✓	✗	✗
YubiKey 5 Series	19083c3d-8383-4b18-bc03-8f1c9ab2fd1b	✗	✓	✗	✗
YubiKey 5 Series	cb69481e-8ff7-4039-93ec-0a2729a154a8	✗	✓	✗	✗
YubiKey 5 Series	ee882879-721c-4913-9775-3dfcce97072a	✗	✓	✗	✗
YubiKey 5 Series	ff4dac45-ede8-4ec2-aced-cf66103f4335	✗	✓	✗	✗
YubiKey 5 Series (Enterprise Profile)	4599062e-6926-4fe7-9566-9e8fb1aedaa0	✗	✓	✗	✗
YubiKey 5 Series (Enterprise Profile)	20ac7a17-c814-4833-93fe-539f0d5e3389	✗	✓	✗	✗

Description	AAGUID	Bio	USB	NFC	BLE
YubiKey 5 Series with Lightning	c5ef55ff-ad9a-4b9f-b580-adebafe026d0	✗	✓	✗	✗
YubiKey 5 Series with Lightning	a02167b9-ae71-4ac7-9a07-06432ebb6f1c	✗	✓	✗	✗
YubiKey 5 Series with Lightning	24673149-6c86-42e7-98d9-433fb5b73296	✗	✓	✗	✗
YubiKey 5 Series with Lightning (Enterprise Profile)	b90e7dc1-316e-4fee-a25a-56a666a670fe	✗	✓	✗	✗
YubiKey 5 Series with Lightning (Enterprise Profile)	3b24bf49-1d45-4484-a917-13175df0867b	✗	✓	✗	✗
YubiKey 5 Series with Lightning Preview	3124e301-f14e-4e38-876d-fbeeb090e7bf	✗	✓	✗	✗
YubiKey 5 Series with NFC	fa2b99dc-9e39-4257-8f92-4a30d23c4118	✗	✓	✓	✗
YubiKey 5 Series with NFC	2fc0579f-8113-47ea-b116-bb5a8db9202a	✗	✓	✓	✗
YubiKey 5 Series with NFC	d7781e5d-e353-46aa-afe2-3ca49f13332a	✗	✓	✓	✗
YubiKey 5 Series with NFC	a25342c0-3cdc-4414-8e46-f4807fca511c	✗	✓	✓	✗
YubiKey 5 Series with NFC (Enterprise Profile)	1ac71f64-468d-4fe0-bef1-0e5f2f551f18	✗	✓	✗	✗
YubiKey 5 Series with NFC (Enterprise Profile)	6ab56fad-881f-4a43-acb2-0be065924522	✗	✓	✓	✗
YubiKey 5 Series with NFC Preview	34f5766d-1536-4a24-9033-0e294e510fb0	✓	✓	✗	✗
YubiKey Bio Series - FIDO Edition	d8522d9f-575b-4866-88a9-ba99fa02f35b	✓	✓	✗	✗
YubiKey Bio Series - FIDO Edition	dd86a2da-86a0-4cbe-b462-4bd31f57bc6f	✓	✓	✗	✗
YubiKey Bio Series - FIDO Edition	7409272d-1ff9-4e10-9fc9-ac0019c124fd	✓	✓	✗	✗
YubiKey Bio Series - FIDO Edition (Enterprise Profile)	ad08c78a-4e41-49b9-86a2-ac15b06899e2	✓	✓	✗	✗

Description	AAGUID	Bio	USB	NFC	BLE
YubiKey Bio Series - FIDO Edition (Enterprise Profile)	8c39ee86-7f9a-4a95-9ba3-f6b097e5c2ee	✓	✓	✗	✗
YubiKey Bio Series - FIDO Edition (Enterprise Profile)	83c47309-aabb-4108-8470-8be838b573cb	✓	✓	✗	✗
YubiKey Bio Series - Multi-protocol Edition	90636e1f-ef82-43bf-bdcf-5255f139d12f	✓	✓	✗	✗
YubiKey Bio Series - Multi-protocol Edition	34744913-4f57-4e6e-a527-e9ec3c4b94e6	✓	✓	✗	✗
YubiKey Bio Series - Multi-protocol Edition	7d1351a6-e097-4852-b8bf-c9ac5c9ce4a3	✓	✓	✗	✗
YubiKey Bio Series - Multi-protocol Edition (Enterprise Profile)	6ec5cff2-a0f9-4169-945b-f33b563f7b99	✓	✓	✗	✗
YubiKey Bio Series - Multi-protocol Edition (Enterprise Profile)	97e6a830-c952-4740-95fc-7c78dc97ce47	✓	✓	✗	✗
YubiKey Bio Series - Multi-protocol Edition 1VDJSN	58276709-bb4b-4bb3-baf1-60eea99282a7	✓	✓	✗	✗
ZTPass SmartAuth	b415094c-49d3-4c8b-b3fe-7d0ad28a6bc4	✗	✓	✓	✗
ZTPass SmartAuth	99bf4610-ec26-4252-b31f-7380ccd59db5	✗	✓	✓	✗

Next steps

For more information about Microsoft Entra ID support for phishing-resistant authentication with FIDO2 security keys in browsers and native apps, see [FIDO2 compatibility](#).

Enable passkeys in Authenticator

Article • 04/02/2025

This article lists steps to enable and enforce use of passkeys in Authenticator for Microsoft Entra ID. First, you update the Authentication methods policy to allow users to register and sign in with passkeys in Authenticator. Then you can use Conditional Access authentication strengths policies to enforce passkey sign-in when users access a sensitive resource.

Requirements

- [Microsoft Entra multifactor authentication \(MFA\)](#).
- Android 14 and later or iOS 17 and later.
- For cross-device registration and authentication:
 - Both devices must have Bluetooth enabled.
 - Internet connectivity to these two endpoints must be allowed in your organization:
 - <https://cable.ua5v.com>
 - <https://cable.auth.com>

 Note

Users can't use cross-device registration if you enable attestation.

To learn more about FIDO2 support, see [Support for FIDO2 authentication with Microsoft Entra ID](#).

Enable passkeys in Authenticator in the admin center

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Authentication method policy**.
3. Under the method **Passkey (FIDO2)**, select **All users** or **Add groups** to select specific groups. *Only security groups are supported.*

4. On the **Configure** tab:

- Set **Allow self-service set up** to **Yes**. If it's set to **No**, users can't register a passkey by using [Security info](#), even if passkeys (FIDO2) are enabled by the Authentication methods policy.
- Set **Enforce attestation** to **Yes** or **No**.

When attestation is enabled in the passkey (FIDO2) policy, Microsoft Entra ID tries to verify the legitimacy of the passkey being created. When the user is registering a passkey in the Authenticator, attestation verifies that the legitimate Authenticator app created the passkey by using Apple and Google services. Here are more details:

- **iOS:** Authenticator attestation uses the [iOS App Attest service](#) to ensure the legitimacy of the Authenticator app before registering the passkey.
- **Android:**
 - For Play Integrity attestation, Authenticator attestation uses the [Play Integrity API](#) to ensure the legitimacy of the Authenticator app before registering the passkey.
 - For Key attestation, Authenticator attestation uses [key attestation by Android](#) to verify that the passkey being registered is hardware-backed.

 **Note**

For both iOS and Android, Authenticator attestation relies upon Apple and Google services to verify the authenticity of the Authenticator app. Heavy service usage can make passkey registration fail, and users might need to try again. If Apple and Google services are down, Authenticator attestation blocks registration that requires attestation until services are restored. To monitor the status of Google Play Integrity service, see [Google Play Status Dashboard](#). To monitor the status of the iOS App Attest service, see [System Status](#).

 **Note**

Users can only register attested passkeys directly in the Authenticator app. Cross-device registration flows don't support registration of attested passkeys.

- **Key restrictions** set the usability of specific passkeys for both registration and authentication. You can set **Enforce key restrictions** to **No** to allow users to register any supported passkey, including passkey registration directly in the Authenticator app. If you set **Enforce key restrictions** to **Yes** and already have active passkey usage, you should collect and add the AAGUIDs of the passkeys being used today.

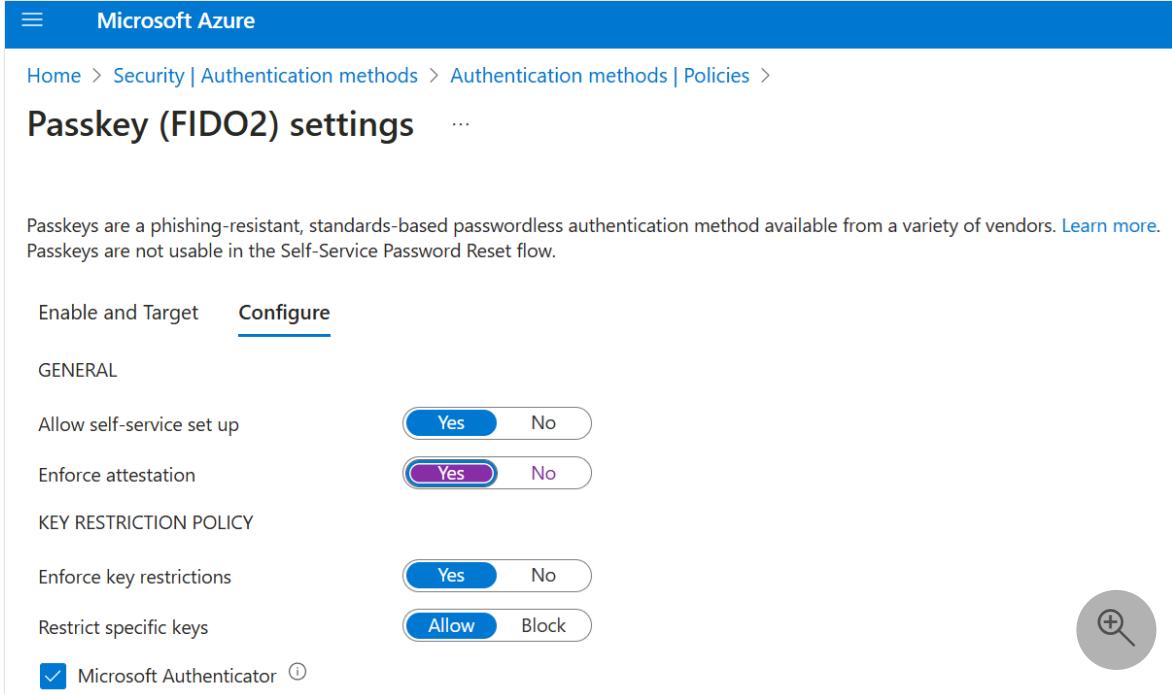
If you set **Restrict specific keys** to **Allow**, select **Microsoft Authenticator** to automatically add the Authenticator app AAGUIDs to the key restrictions list. You can also manually add the following AAGUIDs to allow users to register passkeys in Authenticator by signing in to the Authenticator app or by going through a guided flow on **Security info**:

- **Authenticator for Android:** de1e552d-db1d-4423-a619-566b625cdc84
- **Authenticator for iOS:** 90a3ccdf-635c-4729-a248-9b709135078f

If you change key restrictions and remove an AAGUID that you previously allowed, users who previously registered an allowed method can no longer use it for sign-in.

Note

If you turn off key restrictions, make sure you clear the **Microsoft Authenticator** checkbox so that users aren't prompted to set up a passkey in the Authenticator app on [Security info](#).



The screenshot shows the Microsoft Azure 'Passkey (FIDO2) settings' page. At the top, there's a navigation bar with 'Home > Security | Authentication methods > Authentication methods | Policies > Passkey (FIDO2) settings'. Below the navigation, a note states: 'Passkeys are a phishing-resistant, standards-based passwordless authentication method available from a variety of vendors. [Learn more](#). Passkeys are not usable in the Self-Service Password Reset flow.' The main area has tabs for 'Enable and Target' and 'Configure', with 'Configure' being active. Under 'GENERAL', there are two buttons: 'Allow self-service set up' (Yes) and 'Enforce attestation' (Yes). Under 'KEY RESTRICTION POLICY', there are two buttons: 'Enforce key restrictions' (Yes) and 'Restrict specific keys' (Allow). A 'Microsoft Authenticator' checkbox is checked. In the bottom right corner, there's a circular button with a plus sign and a magnifying glass icon.

5. After you finish the configuration, select **Save**.

If you see an error when you try to save, replace multiple groups with a single group in one operation, and then select **Save** again.

Enable passkeys in Authenticator by using Graph Explorer

In addition to using the Microsoft Entra admin center, you can also enable passkeys in Authenticator by using Graph Explorer. If you're assigned at least the [Authentication Policy Administrator](#) role, you can update the Authentication methods policy to allow the AAGUIDs for Authenticator.

To configure the policy by using Graph Explorer:

1. Sign in to [Graph Explorer](#) and consent to the **Policy.Read.All** and **Policy.ReadWrite.AuthenticationMethod** permissions.
2. Retrieve the Authentication methods policy:

JSON

GET

```
https://graph.microsoft.com/v1.0/authenticationMethodsPolicy/authenticationMethodConfigurations/FID02
```

3. To disable attestation enforcement and enforce key restrictions to allow only AAGUIDs for Authenticator, perform a **PATCH** operation by using the following request body:

JSON

PATCH

```
https://graph.microsoft.com/v1.0/authenticationMethodsPolicy/authenticationMethodConfigurations/FID02
```

Request Body:

```
{  
    "@odata.type": "#microsoft.graph.fido2AuthenticationMethodConfiguration",  
    "isAttestationEnforced": true,  
    "keyRestrictions": {  
        "isEnforced": true,  
        "enforcementType": "allow",  
        "aaGuids": [  
            "90a3ccdf-635c-4729-a248-9b709135078f",  
            "de1e552d-db1d-4423-a619-566b625cdc84"  
        ]  
    }  
}
```

<insert previous AAGUIDs here to keep them stored in

```
    policy>
        ]
    }
}
```

4. Make sure that the passkey (FIDO2) policy is updated properly.

JSON

GET

<https://graph.microsoft.com/v1.0/authenticationMethodsPolicy/authenticationMethodConfigurations/FIDO2>

Find AAGUIDs

Use the `GetRegisteredPasskeyAAGUIDsForAllUsers.ps1` Microsoft Graph PowerShell script to enumerate the AAGUIDs of all registered passkeys in the tenant.

Save the body of this script to a file called `GetRegisteredPasskeyAAGUIDsForAllUsers.ps1`.

PowerShell

```
# Disconnect from Microsoft Graph
Disconnect-MgGraph

# Connect to Microsoft Graph with required scopes
Connect-MgGraph -Scope
'User.Read,UserAuthenticationMethod.Read,UserAuthenticationMethod.Read.All'

# Define the output file [If the script is run more than once, delete the
# file to avoid appending to it.]
$file = ".\AAGUIDs.txt"

# Initialize the file with a header
Set-Content -Path $file -Value '---'

# Retrieve all users
$userArray = Get-MgBetaUser -All

# Iterate through each user
foreach ($user in $userArray) {
    # Retrieve Passkey authentication methods for the user
    $fidos = Get-MgBetaUserAuthenticationFido2Method -UserId $user.Id

    if ($fidos -eq $null) {
        # Log and write to file if no Passkey methods are found
        Write-Host "User object ID $($user.Id) has no Passkey"
        Add-Content -Path $file -Value "User object ID $($user.Id) has no
Passkey"
```

```

} else {
    # Iterate through each Passkey method
    foreach ($fido in $fidos) {
        # Log and write to file the Passkey details
        Write-Host "- User object ID $($user.Id) has a Passkey with
AAGUID $($fido.Aaguid) of Model type '$($fido.Model)'"
        Add-Content -Path $file -Value "- User object ID $($user.Id) has
a Passkey with AAGUID $($fido.Aaguid) of Model type '$($fido.Model)'"
    }
}

# Log and write a separator to file
Write-Host "===="
Add-Content -Path $file -Value "===="
}

```

Restrict Bluetooth usage to passkeys in Authenticator

Some organizations restrict Bluetooth usage, which includes the use of passkeys. In such cases, organizations can allow passkeys by permitting Bluetooth pairing exclusively with passkey-enabled FIDO2 authenticators. For more information about how to configure Bluetooth usage only for passkeys, see [Passkeys in Bluetooth-restricted environments](#).

Delete a passkey

If a user deletes a passkey in Authenticator, the passkey is also removed from the user's sign-in methods. An Authentication Policy Administrator can also follow these steps to delete a passkey from the user's authentication methods, but it won't remove the passkey from Authenticator.

1. Sign in to the [Microsoft Entra admin center](#), and search for the user whose passkey must be removed.
2. Select **Authentication methods**, right-click **Passkey**, and select **Delete**.

Unless the user initiated the passkey deletion themselves in Authenticator, they need to also remove the passkey in Authenticator on their device.

Enforce sign-in with passkeys in Authenticator

To make users sign in with a passkey when they access a sensitive resource, use the built-in phishing-resistant authentication strength, or create a custom authentication

strength by following these steps:

1. Sign in to the [Microsoft Entra admin center](#) as a Conditional Access Administrator.
2. Browse to **Protection > Authentication methods > Authentication strengths**.
3. Select **New authentication strength**.
4. Provide a descriptive name for your new authentication strength.
5. Optionally, provide a description.
6. Select **Passkeys (FIDO2)**, and then select **Advanced options**.
7. Select **Phishing-resistant MFA strength** or add AAGUIDs for passkeys in Authenticator:
 - **Authenticator for Android:** `de1e552d-db1d-4423-a619-566b625cdc84`
 - **Authenticator for iOS:** `90a3ccdf-635c-4729-a248-9b709135078f`
8. Select **Next**, and review the policy configuration.

Related content

- [Support for passkey in Windows](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Support passkeys in Authenticator in your Microsoft Entra ID tenant

Article • 03/04/2025

This article covers issues that users might see when they use passkeys in Authenticator and possible ways for administrators to resolve them.

Store passkeys in Android profiles

Passkeys on Android are used only from the profile where they're stored. If a passkey is stored in an Android Work profile, it's used from that profile. If a passkey is stored in an Android Personal profile, it's used from that profile. To make sure that users can access and use the passkey they need, users with both an Android Personal profile and an Android Work profile should create their passkeys in Authenticator for each profile.

Workarounds

Use the following workarounds for Authenticator passkey issues.

Workarounds for an authentication strength Conditional Access policy loop

Users can get in a loop when they try to add a passkey in Authenticator if a Conditional Access policy requires phishing-resistant authentication to access **All resources** (formerly 'All cloud apps'). For example:

- Condition: **All devices (Windows, Linux, macOS, Windows, Android)**
- Targeted resource: **All resources (formerly 'All cloud apps')**
- Grant control: **Authentication strength – Require passkey in Authenticator**

The policy forces targeted users to use a passkey to sign in to all cloud applications, which includes the Authenticator app. It requires users to use a passkey when they try to add a passkey in Authenticator on either Android or iOS.

Here are some workarounds:

- You can [filter for applications](#) and transition the policy target from **All resources** (formerly 'All cloud apps') to specific applications. Start with a review of

applications that are used in your tenant. Use filters to tag Authenticator and other applications.

- To further reduce support costs, you can run an internal campaign to help users adopt passkeys before you enforce them. When you're ready to enforce passkey usage, create two Conditional Access policies:
 - A policy for mobile operating system (OS) versions
 - A policy for desktop OS versions

Require a different authentication strength for each policy, and configure other policy settings listed in the following table. You can enable a [Temporary Access Pass \(TAP\)](#) for users or enable other authentication methods to help users register the passkey.

A TAP limits the time when users can register a passkey. You can accept it only on mobile platforms where you allow passkey registration.

[+] [Expand table](#)

Conditional Access policy	Desktop OS	Mobile OS
Name	Require a passkey in Authenticator to access a desktop OS.	Require a TAP, a phishing-resistant credential, or any other specified authentication method to access a mobile OS.
Condition	Specific devices (desktop operating systems).	Specific devices (mobile operating systems).
Devices	N/A.	Android, iOS.
Exclude devices	Android, iOS.	N/A.
Targeted resource	All resources.	All resources.
Grant control	Authentication strength.	Authentication strength. ¹
Methods	Passkey in Authenticator.	TAP, passkey in Authenticator.
Policy result	Users who can't sign in with a passkey in Authenticator are directed to the My Sign-ins wizard mode. After registration, they're asked to sign in to Authenticator on their mobile device.	Users who sign in to Authenticator with a TAP or another allowed method can register a passkey directly in Authenticator. No loop occurs because the user meets the authentication requirements.

¹For users to register new sign-in methods, your grant control for the mobile policy needs to match your Conditional Access policy to register [Security info](#).

Note

With either workaround, users must also satisfy any Conditional Access policy that targets **Register security info** or they can't register the passkey. If you have other conditions set up with the **All resources** policies, those conditions must be met when the passkey is registered.

Users who can't register passkeys because of **Require approved client app** or **Require app protection policy** Conditional Access grant controls

Users can't register passkeys in Authenticator if they're included in the following Conditional Access policy:

- Condition: **All devices (Windows, Linux, macOS, Windows, Android)**
- Targeted resource: **All resources (formerly 'All cloud apps')**
- Grant control: **Require approved client app** or **Require app protection policy**

The policy forces users to sign in to all cloud applications by using an app that supports [Microsoft Intune app protection policies](#). Authenticator doesn't support this policy on either Android or iOS.

Here are some workarounds:

- You can [filter for applications](#) and transition the policy target from **All resources (formerly 'All cloud apps')** to specific applications. Start with a review of applications that are used in your tenant. Use filters to tag appropriate applications.
- You can use mobile device management (MDM) and the **Require device to be marked as compliant** control. Authenticator can satisfy this grant control if MDM fully manages the device and it's compliant. For example:
 - Condition: **All devices (Windows, Linux, macOS, Windows, Android)**
 - Targeted resource: **All resources (formerly 'All cloud apps')**
 - Grant control: **Require approved client app**, or **Require app protection policy**, or **Require device to be marked as compliant**
- You can grant users a temporary exemption from the Conditional Access policy. We recommend that you use one or more compensating controls:

- Allow the exemption for only a limited period of time. Communicate to the user when they're allowed to register a passkey. Remove the exemption after the time period. Then direct users to call the help desk if they missed their time.
- Use another Conditional Access policy to require that users register only from a specific network location or a compliant device.

 Note

With any proposed workaround, users must also satisfy any Conditional Access policy that targets **Register security info** or they can't register the passkey. If you have other conditions set up with the **All resources** policies, they also must be met before users can register a passkey.

Restrict Bluetooth usage to passkeys in Authenticator

Some organizations restrict Bluetooth usage, which includes the use of passkeys. In such cases, organizations can allow passkeys by permitting Bluetooth pairing exclusively with passkey-enabled FIDO2 authenticators. For more information about how to configure Bluetooth usage only for passkeys, see [Passkeys in Bluetooth-restricted environments](#).

Related content

- For more information about passkeys in Authenticator, see [Microsoft Authenticator authentication method](#).
- To enable passkeys in Authenticator as a way for users to sign in, see [Enable passkeys in Microsoft Authenticator](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Register passkeys in Authenticator on Android or iOS devices

Article • 03/04/2025

This article shows how to register a passkey by using Authenticator on your iOS or Android device by directly signing in to the Authenticator app or by using [Security info](#). For more information about the availability of Microsoft Entra ID passkey (FIDO2) authentication across native apps, web browsers, and operating systems, see [Support for FIDO2 authentication with Microsoft Entra ID](#).

The easiest and fastest way to add a passkey is to add it directly in the Authenticator app.

Alternatively, you can add a passkey from your mobile device browser or through cross-device registration by using another device, such as a laptop. Your mobile device needs to run iOS version 17 or Android version 14, or later.

[+] Expand table

Scenario	iOS	Android
Same-device registration by signing into Authenticator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Same-device registration in a browser	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> ¹
Cross-device registration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

¹Support for same-device registration in Microsoft Edge on Android is coming soon.

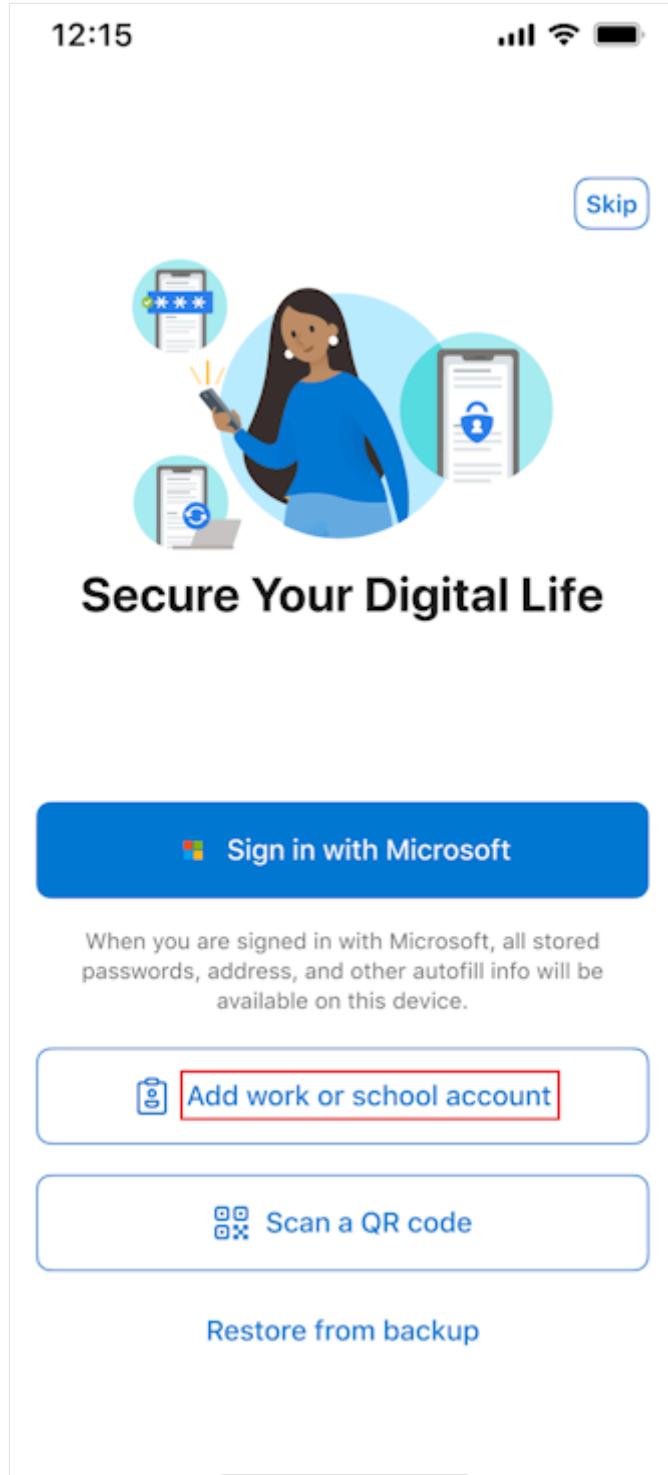
iOS

Registration by signing in to Authenticator (iOS)

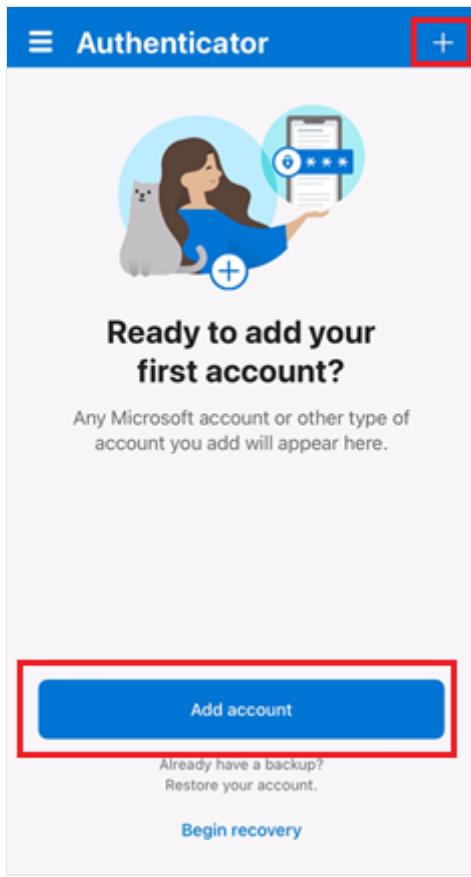
You can sign in to Authenticator to create a passkey in the app and get seamless single sign-on across Microsoft native apps. **We recommend this preferred flow to set up a passkey in Authenticator.** If you're signed in or already have an account in Authenticator, you still need to complete these steps to add a passkey in Authenticator.

1. Download Authenticator from the App Store, and go through the privacy screens.

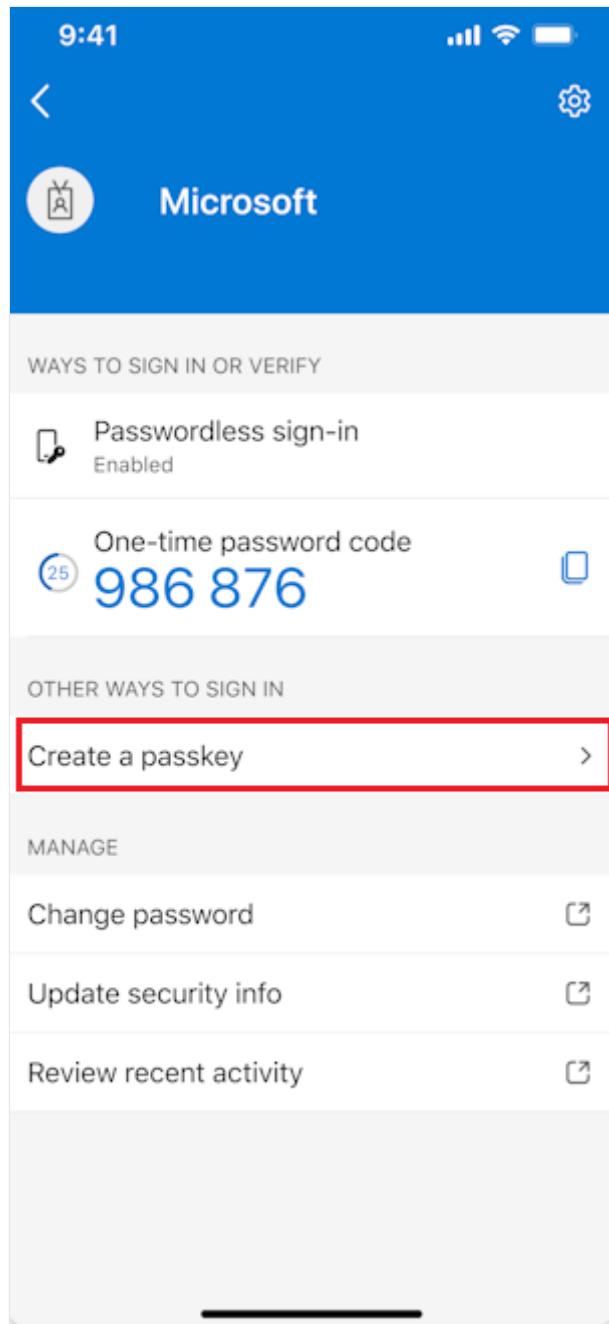
- If you installed Authenticator for the first time on your device, on the **Secure Your Digital Life** screen, tap **Add work or school account**.



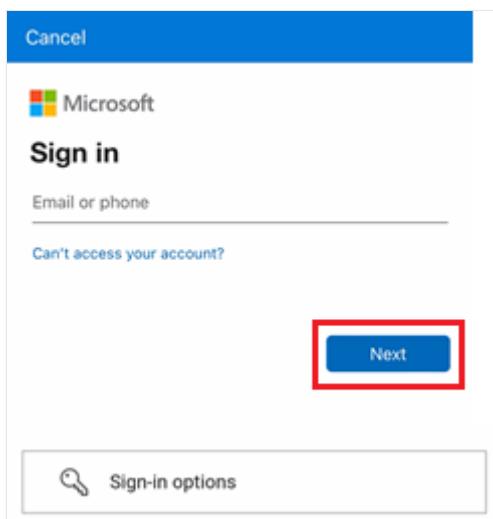
- If you installed Authenticator on your device but you didn't add an account, tap **Add account** or the **+** button, and select **Work or school account**. Then tap **Sign in**.



- If you already added an account in Authenticator, tap your account, and then tap **Create a passkey**.



2. You need to complete multifactor authentication (MFA).



3. If necessary, tap **Settings** and set up a screen lock.

Step 1 of 3

Screen lock required

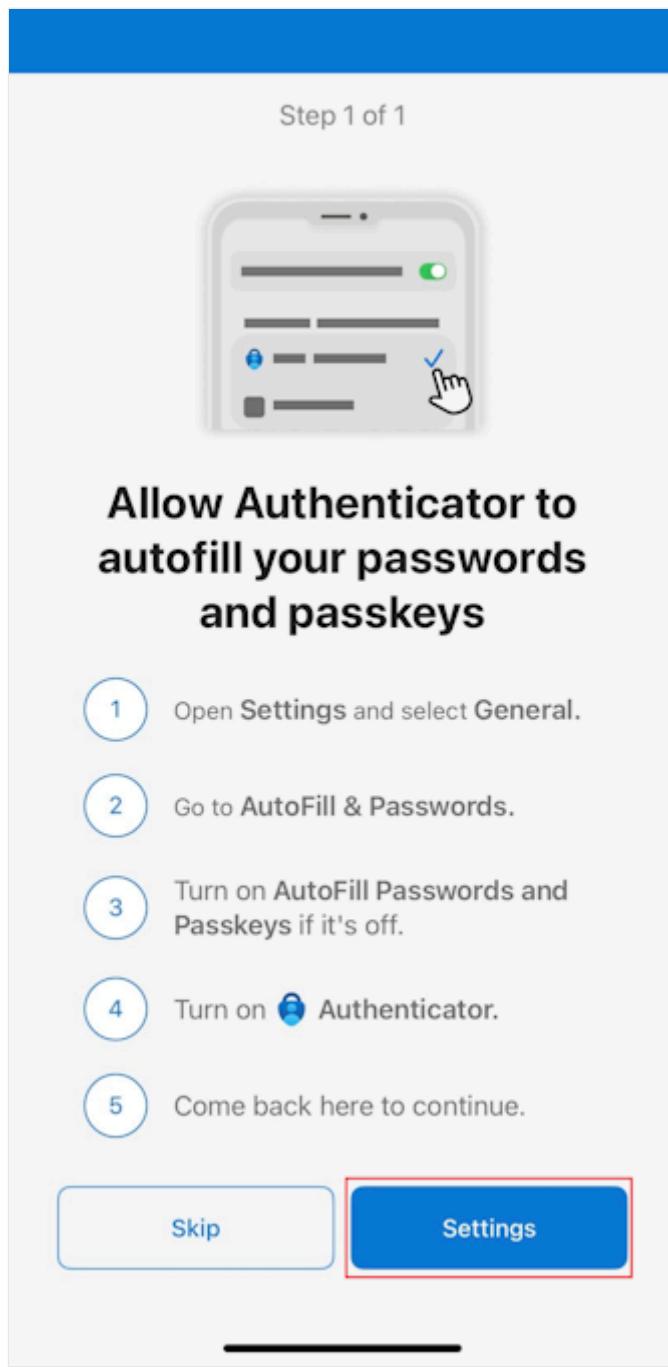


To use passkeys and the passwordless sign-in request methods, you need to set up a screen lock. Your screen lock is a PIN, fingerprint, or facial recognition used to unlock your device.

[Skip](#)

[Settings](#)

4. Tap **Settings** to enable Authenticator as a passkey provider.



5. On your iOS 18 device, go to **Settings** > **General** > **Autofill & Passwords**. On your iOS 17 device, go to **Settings** > **Passwords** > **Password Options**.

On both operating systems, make sure that **AutoFill Passwords and Passkeys** is turned on. Under **Autofill From**, make sure that **Authenticator** is selected.

[Back](#)

AutoFill Passwords and Passkeys



Automatically suggest passwords, passkeys, and verification codes when signing in to apps and websites.

AUTOFILL FROM:

Passwords

Passkeys, passwords, and codes



Authenticator

Passkeys and passwords



VERIFICATION CODES

Delete After Use

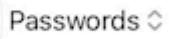


Automatically delete verification codes in Messages and Mail after they are used.

Set Up Codes In



Passwords



Open verification code setup links and QR codes with this app.

6. After you return to Authenticator, tap **Done** to confirm that you added Authenticator as a passkey provider. Then you can see **Passkey** added as a sign-in method for your account. Tap **Done** again to finish.

Account added



You can now sign in or verify using the following methods.

-  **Passkey**
Sign in with your face, fingerprint, or PIN.
[How to use your passkey](#)
-  **Passwordless sign-in requests**
Sign in without a password.
-  **Multifactor authentication**
Approve a sign-in request on your phone.
-  **One-time Password codes**
Use a code generated in Authenticator

Done

- Authenticator sets up passkey, passwordless, and MFA for sign-in according to your work or school account policies. Tap your account to see information, including your new passkey.

Passkey registration from Security info (iOS)

By default, **Security info** prompts users to sign in to the Authenticator app to register their passkey.

- On the same iOS device as the Authenticator or by using another device, such as a laptop, open a web browser and sign in with MFA to [Security info](#).



←

Enter Temporary Access Pass

.....

Show Temporary Access Pass

[Other ways to sign in](#)

Sign in

2. On **Security info**, tap + Add sign-in method and select **Passkey** in Microsoft Authenticator.

Add a sign-in method

X



Passkey in Microsoft Authenticator

Sign in with your face, fingerprint, PIN



Security key or passkey (preview)

Sign in with your face, fingerprint, PIN or security key



Security key

Sign in using a USB, Bluetooth, or NFC device



Microsoft Authenticator

Approve sign-in requests or use one-time codes



Phone

Get a call or text to sign in with a code

3. If you're asked to sign in with MFA, select **Next**.

4. If necessary, download Authenticator to your iOS device. You can select [Microsoft Authenticator](#) and scan a QR code to install Authenticator from the iOS App Store. After you download Authenticator, tap **Next**.

Create your passkey in Microsoft Authenticator

X



A passkey lets you sign in more easily and securely with your face, fingerprint, or PIN.

You will need to install [Microsoft Authenticator](#) on a mobile device with at least Android 14 or iOS 17.



Back

Next

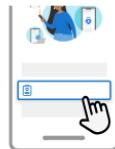
5. You're prompted to open the Authenticator app and create your passkey there. Open Authenticator and go through the privacy screens as needed.

Complete the setup in Microsoft Authenticator

X

To finish creating your passkey, add your work or school account. Then sign in with

Just installed Authenticator?



Select work or
school account and
sign in.

or

Already using Authenticator?



Tap the plus icon
select work or
school account and
sign in.

Back

Next

6. Add your account in Authenticator on your iOS device.

- If you installed Authenticator for the first time on your device, on the **Secure Your Digital Life** screen, tap **Add work or school account**.

12:15



Skip



Secure Your Digital Life

 Sign in with Microsoft

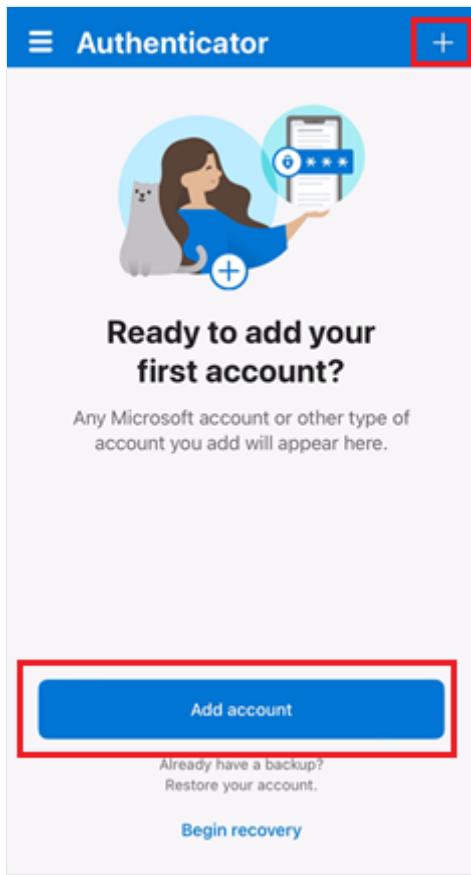
When you are signed in with Microsoft, all stored passwords, address, and other autofill info will be available on this device.

 Add work or school account

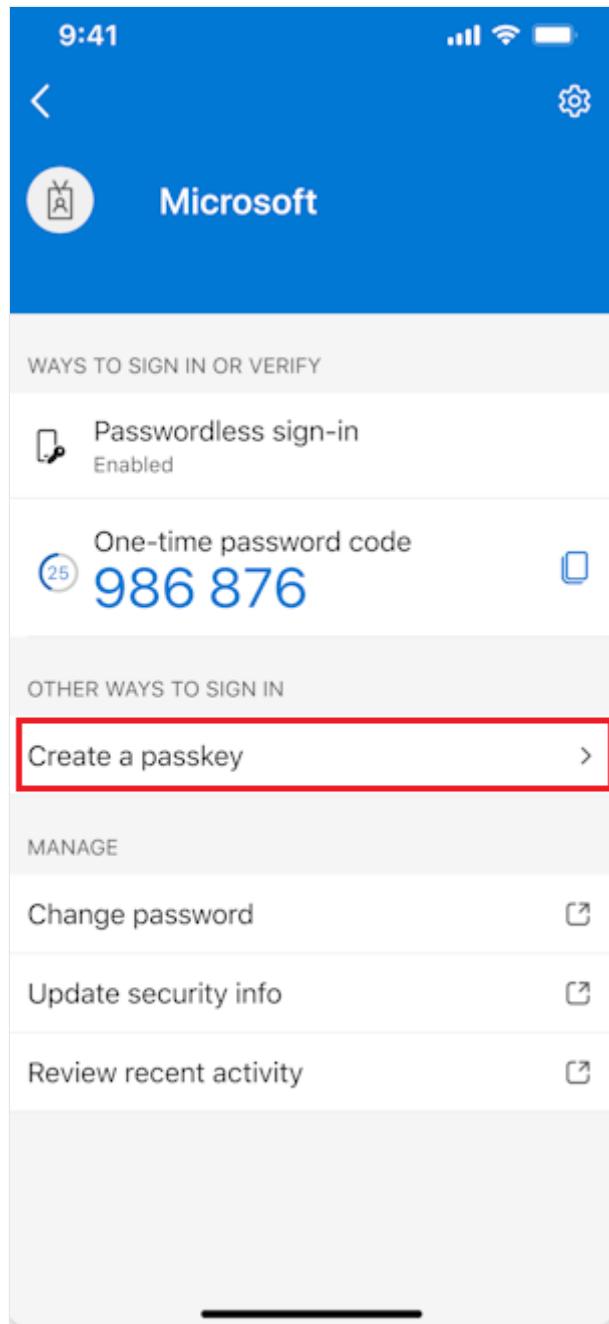
 Scan a QR code

[Restore from backup](#)

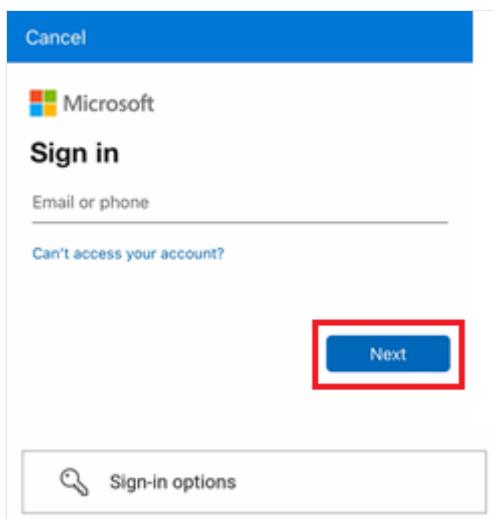
- If you installed Authenticator on your device before but didn't add an account, tap **Add account** or the + button, and select **Work or school account**. Then tap **Sign in**.



- If you already added an account in Authenticator, tap your account, and then tap **Create a passkey**.



7. You need to complete multifactor authentication (MFA).



8. If necessary, tap **Settings** and set up a screen lock.

Step 1 of 3

Screen lock required



To use passkeys and the passwordless sign-in request methods, you need to set up a screen lock. Your screen lock is a PIN, fingerprint, or facial recognition used to unlock your device.

Skip

Settings

9. Tap **Settings** to enable Authenticator as a passkey provider.
10. On your iOS 18 device, go to **Settings > General > Autofill & Passwords**. On your iOS 17 device, go to **Settings > Passwords > Password Options**.
On both operating systems, make sure that **AutoFill Passwords** and **Passkeys** is turned on. Under **Autofill From**, make sure that **Authenticator** is selected.

[Back](#)

AutoFill Passwords and Passkeys



Automatically suggest passwords, passkeys, and verification codes when signing in to apps and websites.

AUTOFILL FROM:



Passwords

Passkeys, passwords, and codes



Authenticator

Passkeys and passwords



VERIFICATION CODES

Delete After Use

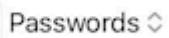


Automatically delete verification codes in Messages and Mail after they are used.

Set Up Codes In



Passwords



Open verification code setup links and QR codes with this app.

11. After you return to Authenticator, tap **Done** to confirm that you added Authenticator as a passkey provider. Then you can see **Passkey** added as a sign-in method for your account. Tap **Done** again to finish.

Account added



You can now sign in or verify using the following methods.

Passkey

Sign in with your face, fingerprint, or PIN.

[How to use your passkey](#)

[Additional sign-in methods](#)

Passwordless sign-in requests

Sign in without a password.

Multifactor authentication

Approve a sign-in request on your phone.

One-time Password codes

Use a code generated in Authenticator

[Done](#)

12. Authenticator sets up passkey, passwordless, and MFA for sign-in according to your work or school account policies.
13. Return to your browser after you finish the passkey setup in Authenticator, and select **Next**.

Complete the setup in Microsoft Authenticator

X

To finish creating your passkey, add your work or school account. Then sign in with

Just installed Authenticator?



Select work or
school account and
sign in.

or

Already using Authenticator?



Tap the plus icon
select work or
school account and
sign in.

Back

Next

14. The wizard verifies that the passkey was created in Authenticator.

Complete the setup in Microsoft Authenticator

X

To finish creating your passkey, add your work or school account. Then sign in with
passkeymanage6@authapp2112.onmicrosoft.com

Just installed Authenticator?



Select work or
school account and
sign in.

or

Already using Authenticator?



Tap the plus icon
select work or
school account and
sign in.

Back

Verifying passkey

15. After the passkey is created, select Done.

Passkey created

X



You can now sign in more easily and securely with your new passkey. Your passkey is
called: Authenticator - iOS

[How to use your passkey to sign in](#)

Done

16. On **Security info**, you can see the new passkey that was added.

The screenshot shows the 'My Sign-Ins' section of the Microsoft Security info page. On the left, there's a sidebar with links: Overview, Security info (which is selected and highlighted in blue), Devices, Password, Organizations, Settings & Privacy, and Recent activity. The main content area is titled 'Security info' and contains a message: 'For security reasons, we recommend that you delete any sign-in methods that you no longer use.' Below this, it says 'These are the methods you use to sign into your account or reset your password.' and 'You're using the most advisable sign-in method where it applies.' A note states 'Sign-in method when most advisable is unavailable: Phone - text' with a 'Change' link. There's a 'Add sign-in method' button. The list of sign-in methods includes:

- Phone (selected, with 'Change' and 'Delete' buttons)
- Password (Last updated: 5 months ago, with 'Change' and 'Delete' buttons)
- Microsoft Authenticator (in microsoftAuthenticator) iPhone (Push multi-factor authentication (MFA), with 'Delete' button)
- Passekey (preview) Microsoft Authenticator (Authenticator - iOS, with 'Delete' button)
- Temporary access pass (Expires 10/31/2024, 7:37:42 AM, with 'Delete' button)

At the bottom, there's a link 'Lost device? Sign out everywhere'.

Alternate registration flow from Security info if you have trouble (iOS)

If you can't sign in to Authenticator to register a passkey, you can register directly from **Security info** with WebAuthn.

(!) Note

You can't register a passkey in Authenticator this way if attestation is enabled by your administrator.

If you sign in to **Security info** on a different device, you need Bluetooth and an internet connection. Connectivity to the following two endpoints must be allowed in your organization:

- <https://cable.ua5v.com>
- <https://cable.auth.com>

If your organization restricts Bluetooth usage, you can permit Bluetooth pairing exclusively with passkey-enabled FIDO2 authenticators to allow cross-device registration of passkeys. For more information, see [Passkeys in Bluetooth-restricted environments](#).

1. On **Security info**, when you add a passkey in Authenticator, tap **Having trouble**.

Complete the setup in Microsoft Authenticator

X

To finish creating your passkey, add your work or school account. Then sign in with

Just installed Authenticator?



Select work or
school account and
sign in.

or

Already using Authenticator?



Tap the plus icon
select work or
school account and
sign in.

[Having trouble?](#)

[Back](#)

[Next](#)

2. Now, tap **create your passkey a different way**.

Having Trouble?

X

Can't sign in to Microsoft Authenticator? You can still [create your passkey a different way](#) using your browser and mobile device. This requires Bluetooth on both devices.

For more information, go to our [support page](#). If you still need help, contact your admin.

[Close](#)

3. Select **iPhone or iPad**, and go through the rest of the flow to register a passkey on the device.

Which device do you want to use?

X



Android

Passkeys require at least Android 14



iPhone or iPad

Passkeys require at least iOS 17 or iPad OS 17

If a user wants to revert to the original instructions and register a passkey in Authenticator through sign-in:

1. On **Security info**, when you add a passkey in Authenticator, tap **Having trouble**.

2. Now, tap **create your passkey a different way** by signing in to Authenticator.
3. Go through the rest of the flow to register a passkey on your device.

 **Note**

If you register your passkey with the Chrome browser on macOS, allow login.microsoft.com to access your security key or device when prompted.

Delete your passkey in Authenticator for iOS

To remove the passkey from Authenticator, tap the account name, and then tap **Settings > Delete passkey**. You also need to delete your passkey from [Security info](#).

Troubleshooting

In some cases when you try to register a passkey, it gets stored locally in the Authenticator app but isn't registered on the authentication server. For example, the passkey provider might not be permitted or the connection might time out. If you try to register a passkey and see an error that the passkey already exists, [delete the passkey](#) that was created locally in Authenticator and retry registration.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Sign in with passkeys in Authenticator for Android and iOS devices

Article • 03/04/2025

This article explains the sign-in experience when you use passkeys in Authenticator with Microsoft Entra ID. For more information about the availability of Microsoft Entra ID passkey (FIDO2) authentication across native applications, web browsers, and operating systems, see [Support for FIDO2 authentication with Microsoft Entra ID](#).

[+] Expand table

Scenario	iOS	Android
Same-device authentication in a browser	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> ¹
Same-device authentication in native Microsoft applications	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Cross-device authentication	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

¹Support for same-device registration in Microsoft Edge on Android is coming soon.

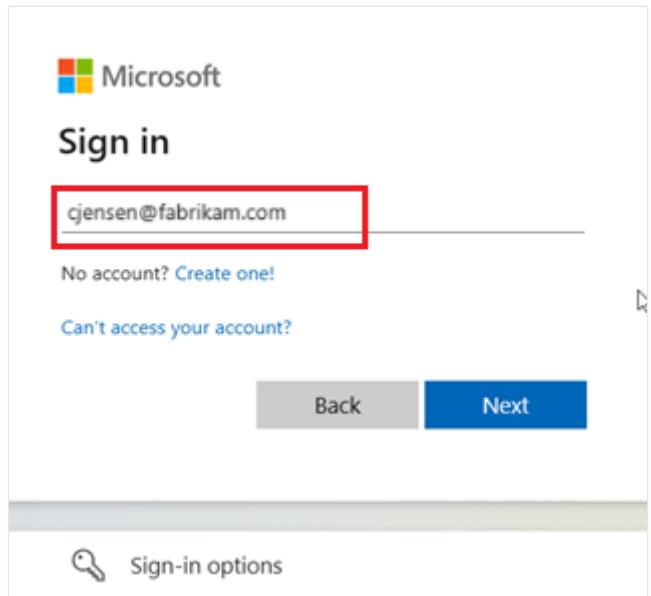
iOS

To sign in with a passkey in Authenticator, your iOS device needs to run iOS 17 or later.

Same-device authentication in a browser (iOS)

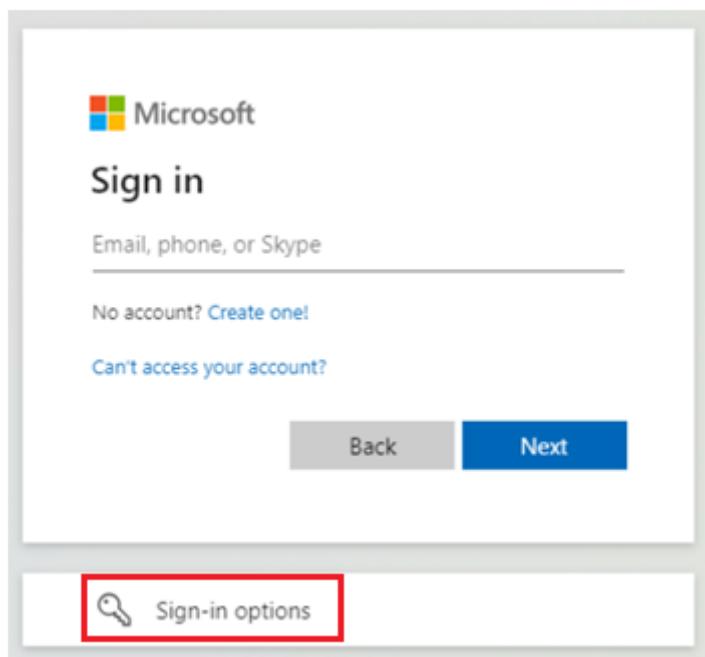
Follow these steps to sign in to Microsoft Entra ID with a passkey in Authenticator on your iOS device.

1. On your iOS device, open your browser and go to the resource you're trying to access, such as [Office](#).
2. Enter your username to sign in.

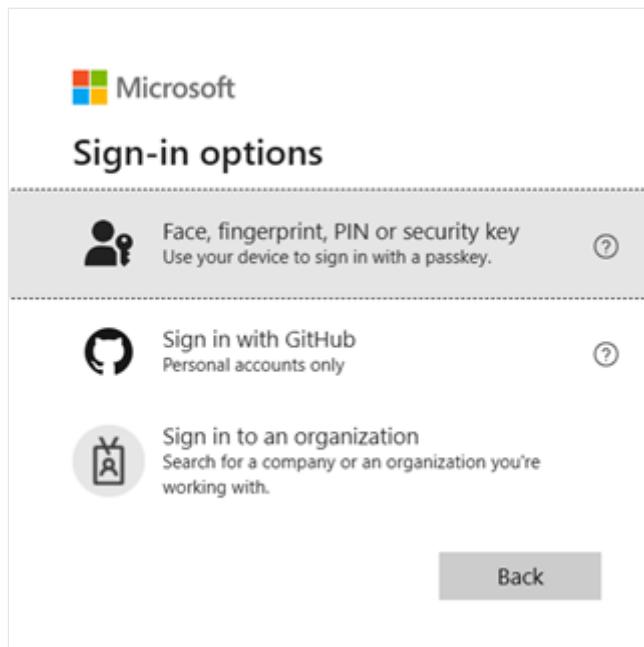


If you most recently used a passkey to sign in, you're prompted to sign in with a passkey. Otherwise, select **Other ways to sign in**, and then select **Face, fingerprint, PIN or security key**.

Alternatively, select **Sign-in options** to sign in without entering a username.



If you selected **Sign-in options**, then select **Face, fingerprint, PIN or security key**. Otherwise, skip to the next step.



ⓘ Note

If you try to sign in without a username and multiple passkeys are saved to your device, you're prompted to choose which passkey to use for sign-in.

3. To select your passkey, follow the steps in the iOS operating system dialog. Verify yourself by using Face ID or Touch ID, or by entering your device PIN.

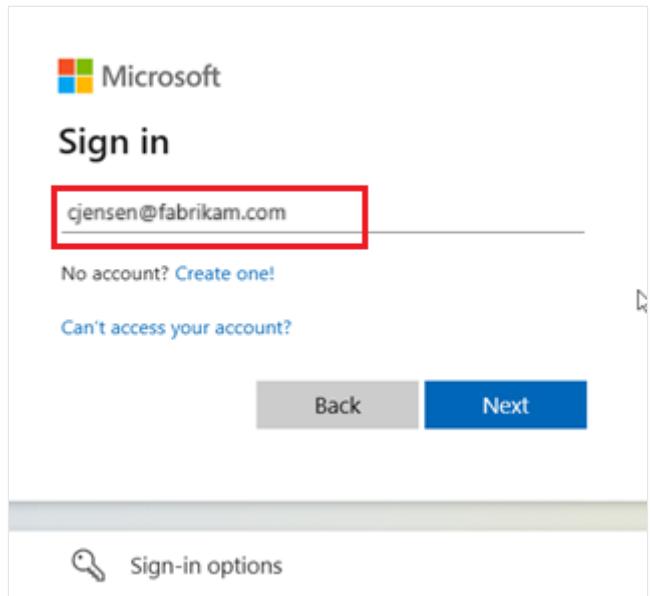
You're now signed in to Microsoft Entra ID.

Cross-device authentication (iOS)

Follow these steps to sign in to Microsoft Entra ID on another device with a passkey in Authenticator on your iOS device.

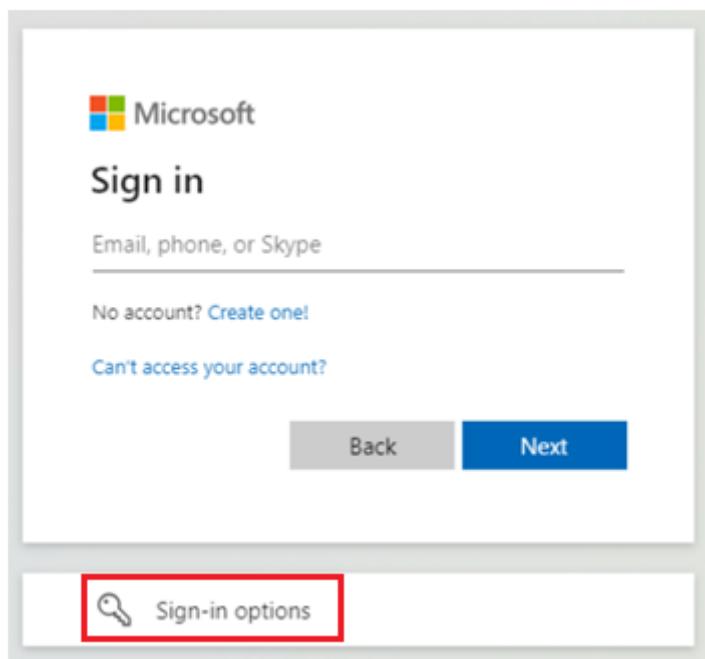
This sign-in option requires Bluetooth and an internet connection for both devices. If your organization restricts Bluetooth usage, an administrator can allow cross-device sign-in for passkeys by permitting Bluetooth pairing exclusively with passkey-enabled FIDO2 authenticators. For more information about how to configure Bluetooth usage only for passkeys, see [Passkeys in Bluetooth-restricted environments](#).

1. On the other device where you want to sign in to Microsoft Entra ID, go to the resource you're trying to access, such as [Office](#).
2. Enter your username to sign in.

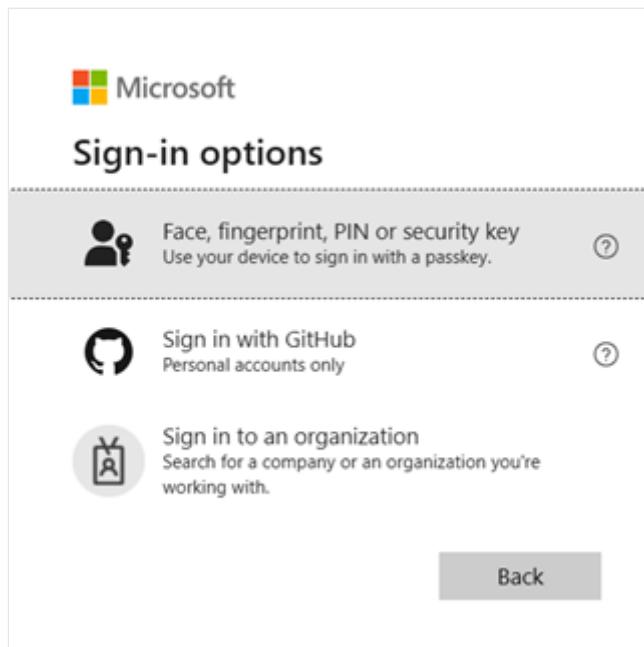


If you last used a passkey to authenticate, you're prompted to authenticate with a passkey. Otherwise, select **Other ways to sign in**, and then select **Face, fingerprint, PIN or security key**.

Alternatively, select **Sign-in options** to sign in without entering a username.



If you selected **Sign-in options**, then select **Face, fingerprint, PIN or security key**. Otherwise, skip to the next step.



ⓘ Note

If you try to sign in without a username and multiple passkeys are saved to your device, you're prompted to choose which passkey to use for sign-in.

3. To begin cross-device authentication, follow the steps in the operating system or browser prompt. On Windows 11 23H2 or later, select **iPhone, iPad, or Android device**.
4. A QR code should appear on the screen. Now, on your iOS device, open the camera app and scan the QR code.

The camera inside the iOS Authenticator app doesn't support scanning a WebAuthn QR code. You need to use the system camera app.

5. Select **Sign in with passkey** when the option appears.

Bluetooth and an internet connection are required for this step and must both be enabled on your mobile and remote device.

6. To select your passkey, follow the steps in the iOS operating system dialog. Verify yourself by using Face ID or Touch ID, or by entering your device PIN.

You're now signed in to Microsoft Entra ID on your other device.

Same-device authentication in native Microsoft applications (iOS)

You can use Authenticator on your iOS device to seamlessly sign in with a passkey to other Microsoft apps, such as OneDrive, SharePoint, and Outlook.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback !\[\]\(91d7db494ac368a91da1a247e3bce644_img.jpg\)](#)

Frequently asked questions about passkey support in Microsoft Authenticator

FAQ

This article addresses frequently asked questions about passkey support in Microsoft Authenticator. Keep checking back for updated content.

How does Microsoft Authenticator store passkeys on a device?

Authenticator passkeys are backed by hardware.

On iOS, Authenticator stores the private key in the [Secure Enclave](#).

On Android, Authenticator uses the [Android Keystore system API](#) to securely store device-bound passkeys. The Android Keystore system supports binding key material to the secure hardware of an Android device, in this order of preference:

- [Secure Element \(SE\)](#)
- [Trusted Execution Environment \(TEE\)](#)

On Android, Authenticator only stores a passkey (private key) if the Android device has one of these two secure hardware options. If neither hardware option exists, Authenticator passkey registration fails, even if attestation is disabled.

Can I restore or sync Authenticator passkeys to a new device?

Authenticator passkeys are only device-bound and can't be synced. For more information, see [Device-bound passkeys in Microsoft Authenticator](#).

Do I need to enable Bluetooth to perform cross-device authentication?

To complete cross-device authentication by using passkeys in Authenticator, you must enable Bluetooth on both the laptop and the mobile device. Both devices need internet access.

Can I have multiple passkeys in Authenticator?

You can have only one passkey for each account in Authenticator. At this time, Authenticator only supports passkeys for Microsoft Entra ID.

Can I use the Authenticator app camera to scan the WebAuthn QR code for registration and authentication?

You can use the Authenticator camera to register and authenticate with passkeys. This option is useful if your organization doesn't push the system camera app to Android Work Profile.

Can I use passkeys in Authenticator without an internet connection?

You can't use passkeys without an internet connection. For same-device scenarios, the mobile device that contains the passkey needs internet access. For cross-device scenarios, both the device with the passkey and the secondary device where you want to sign in need internet access.

Why is cross-device registration failing with "Device couldn't connect"?

Make sure that Bluetooth and an active internet connection are enabled on both devices. Also, allow connectivity to these two endpoints in your organization to enable cross-device registration and authentication:

- <https://cable.ua5v.com>
- <https://cable.auth.com>

I'm on an Android 14 device, and I followed all the steps. Why can't I register passkeys in the Authenticator app?

The Authenticator app uses [Android APIs](#) on Android 14 or higher to use passkeys. Manufacturers choose whether or not to implement these APIs for each device they make. If your device doesn't support these APIs, the Authenticator app might not work for your device on Android 14. For the best experience, we recommend that you upgrade to Android 15.

Why do I get prompted for PIN instead of biometric sign-in on my Android device?

If biometric sign-in fails on an Android device, the Authenticator app will prompt you to enter your PIN instead. The next time you sign in with the passkey, Authenticator continues to request the PIN rather than biometric sign-in. Authenticator periodically retries biometric sign-in. If biometric sign-in succeeds, it will be used for subsequent sign-ins.

What happens to my passkey after I change my PIN or biometric sign-in on my Android device?

Your passkey is invalidated if you change your PIN, or if you change your biometric sign in from thumbprint to face, or vice-versa. If your passkey is invalidated, you need to sign-in by using a different method, and then create a new passkey.

Can I sign in with a passkey in Authenticator in China?

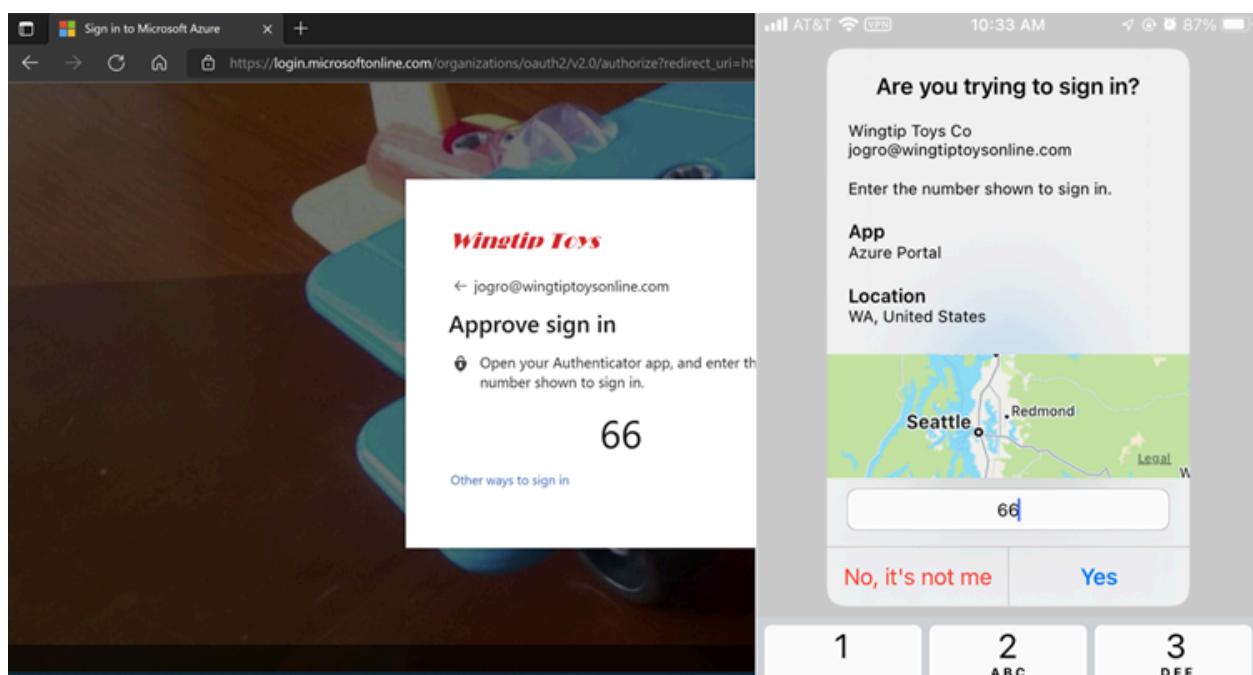
No. Only limited authentication methods are available for Authenticator in China. For more information, see [Download Microsoft Authenticator in China](#).

Enable passwordless sign-in with Authenticator

Article • 03/04/2025

Authenticator is used to sign in to any Microsoft Entra account without using a password. Authenticator uses key-based authentication to enable a user credential that's tied to a device, where the device uses a PIN or biometric. [Windows Hello for Business](#) uses a similar technology.

Authentication technology can be used on any device platform, including mobile. Authenticator can run on either iOS or Android.



Phone sign-in from Authenticator shows a message that asks the user to tap a number in the app. It doesn't ask for a username or password. To complete the sign-in process in the app, follow these steps:

1. In the Authenticator dialog, enter the number shown on the sign-in screen.
2. Select **Approve**.
3. Provide your PIN or biometric.

Multiple accounts

You can enable passwordless phone sign-in for multiple accounts in Authenticator on any supported Android or iOS device. Consultants, students, and other users with multiple accounts in Microsoft Entra ID can add each account to Authenticator and use passwordless phone sign-in for all of them from the same device.

The Microsoft Entra accounts can be in the same tenant or different tenants. Guest accounts aren't supported for multiple account sign-ins from one device.

Prerequisites

To use passwordless phone sign-in with Authenticator, you must meet the following prerequisites:

- Recommended: Microsoft Entra multifactor authentication (MFA), with push notifications allowed as a verification method. Push notifications to a user smartphone or tablet help the Authenticator app to prevent unauthorized access to accounts and stop fraudulent transactions. The Authenticator app automatically generates codes when set up to do push notifications. A user has a backup sign-in method even if their device doesn't have connectivity.
- The device must be registered with each tenant where it's used to sign in. For example, the following device must be registered with Contoso and Wingtip Toys to allow all accounts to sign in:
 - balas@contoso.com
 - balas@wingtiptoys.com and bsandhu@wingtiptoys

To use passwordless authentication in Microsoft Entra ID, first enable the combined registration experience, and then enable users for the passwordless method.

Enable passwordless phone sign-in authentication methods

Microsoft Entra ID lets [Authentication Policy Administrators](#) choose which authentication methods can be used to sign in. You can enable **Microsoft Authenticator** in the Authentication methods policy to manage both the traditional push MFA method and the passwordless authentication method.

After **Microsoft Authenticator** is enabled as an authentication method, users can go to [Security info](#) to register Authenticator as a way to sign in. **Microsoft Authenticator** is listed as a method on **Security info**. For example, **Microsoft Authenticator-Passwordless** or **Microsoft Authenticator-MFA Push** appears, depending on what's enabled and registered.

To enable the authentication method for passwordless phone sign-in, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).

2. Browse to Protection > Authentication methods > Policies.

Each group is enabled by default to use **Any** mode. **Any** mode allows group members to sign in with either a push notification or passwordless phone sign-in.

ⓘ Note

If you see an error when you try to save, it might be because of the number of users or groups being added. As a workaround, replace the users and groups that you're trying to add with a single group in the same operation. Then select **Save** again.

User registration

Users register for the passwordless authentication method of Microsoft Entra ID. Users who already registered the Authenticator app for [MFA](#) can skip to the next section and [enable phone sign-in](#).

Direct phone sign-in registration

Users can register for passwordless phone sign-in directly within the Authenticator app without the need to first register Authenticator with their account, all while never accruing a password. Here's how:

1. Acquire a [Temporary Access Pass](#) from your admin or organization.
2. Download and install the Authenticator app on your mobile device.
3. Open Authenticator and select **Add account**, and then select **Work or school account**.
4. Select **Sign in**.
5. Follow the instructions to sign in with your account by using the Temporary Access Pass provided by your admin or organization.
6. After sign-in, continue following the extra steps to set up phone sign-in.

Guided registration with My Sign-Ins

ⓘ Note

Users can register Authenticator via combined registration only if the Authenticator authentication mode is set to **Any** or **Push**.

To register the Authenticator app, follow these steps:

1. Browse to [Security info](#).
2. Sign in, and then select **Add method** > **Authenticator app** > **Add** to add Authenticator.
3. Follow the instructions to install and configure the Authenticator app on your device.
4. Select **Done** to finish the Authenticator configuration.

Enable phone sign-in

After users register for the Authenticator app, they need to enable phone sign-in:

1. In **Microsoft Authenticator**, select the account registered.
2. Select **Set up Passwordless sign-in requests**.
3. Follow the instructions in the app to finish registering the account for passwordless phone sign-in.

An organization can direct its users to sign in with their phones, without using a password. For further assistance configuring Authenticator and enabling phone sign-in, see [Sign in to your accounts by using the Authenticator app](#).

 **Note**

If a policy restricts the user from using phone sign-in, the user can't enable it within Authenticator.

Sign in with a passwordless credential

A user can start using passwordless sign-in after all the following actions are completed:

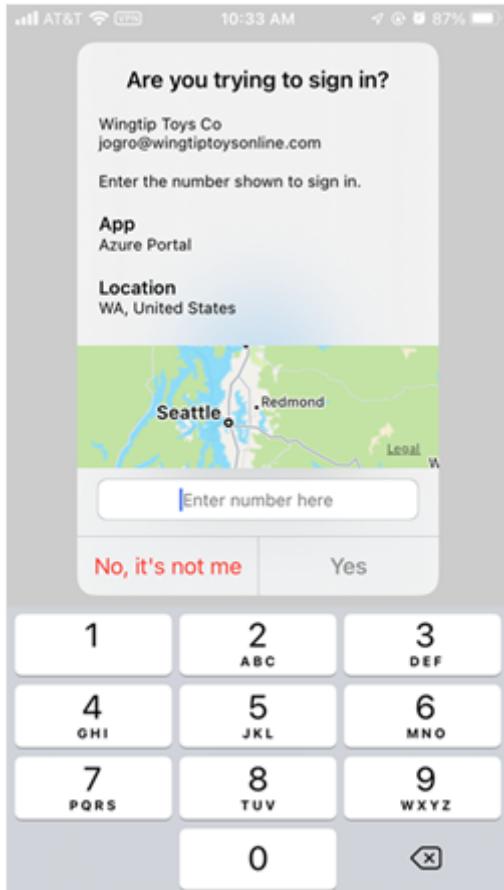
- An admin enabled the user's tenant.
- The user added Authenticator as a sign-in method.

To start the phone sign-in process for the first time, follow these steps:

1. Enter your name on the **Sign-in** pane.
2. Select **Next**.
3. If necessary, select **Other ways to sign in**.
4. Select **Approve a request on my Authenticator app**.

A number then appears. The app prompts the user to authenticate by entering the appropriate number instead of by entering a password.

After the user uses passwordless phone sign-in, the app continues to guide the user through this method. The user also sees the option to choose another method.



Temporary Access Pass

If the tenant administrator enabled self-service password reset for users to set up passwordless sign-in with the Authenticator app for the first time by using a Temporary Access Pass, follow these steps:

1. Open a browser on a mobile device or desktop, and go to [Security info](#).
2. Register the Authenticator app as your sign-in method. This action links your account to the app.
3. Return to your mobile device and activate passwordless sign-in through the Authenticator app.

Management

We recommend the Authentication methods policy as the best way to manage Authenticator. [Authentication Policy Administrators](#) can edit this policy to enable or

disable Authenticator. Admins can include or exclude specific users and groups from using it.

Admins can also configure parameters to better control how Authenticator is used. For example, they can add a location or the app name to the sign-in request so that users have greater context before they approve.

Known issues

The following known issues exist.

Not seeing the option for passwordless phone sign-in

In one scenario, a user might have an unanswered passwordless phone sign-in verification that's pending. If the user attempts to sign in again, the user sees only the option to enter a password.

To resolve this scenario, follow these steps:

1. Open Authenticator.
2. Respond to any notification prompts.

Then continue to use passwordless phone sign-in.

AuthenticatorAppSignInPolicy not supported

The legacy policy `AuthenticatorAppSignInPolicy` isn't supported with Authenticator. To enable users for push notifications or passwordless phone sign-in with the Authenticator app, use the [Authentication Methods policy](#).

Federated accounts

After a user enables any passwordless credential, the Microsoft Entra sign-in process stops using `login_hint`. The process no longer accelerates the user toward a federated sign-in location.

This logic generally prevents a user in a hybrid tenant from being directed to Active Directory Federation Services for sign-in verification. The option to select **Use your password instead** is still available.

On-premises users

Admins can enable users for MFA through an on-premises identity provider. Users can still create and use a single passwordless phone sign-in credential.

If a user attempts to upgrade multiple installations (5+) of Authenticator with the passwordless phone sign-in credential, this change might result in an error.

Related content

To learn about Microsoft Entra authentication and passwordless methods, see the following articles:

- [Learn how passwordless authentication works](#)
 - [Learn about device registration](#)
 - [Learn about Microsoft Entra multifactor authentication](#)
-

Feedback

Was this page helpful?



[Provide product feedback](#) ↗

How number matching works in MFA push notifications for Authenticator - Authentication methods policy

Article • 03/04/2025

This article explains how number matching in Authenticator push notifications improves user sign-in security. Number matching is a key security upgrade to traditional second-factor notifications in Authenticator.

Number matching is enabled for all Authenticator push notifications.

Number matching scenarios

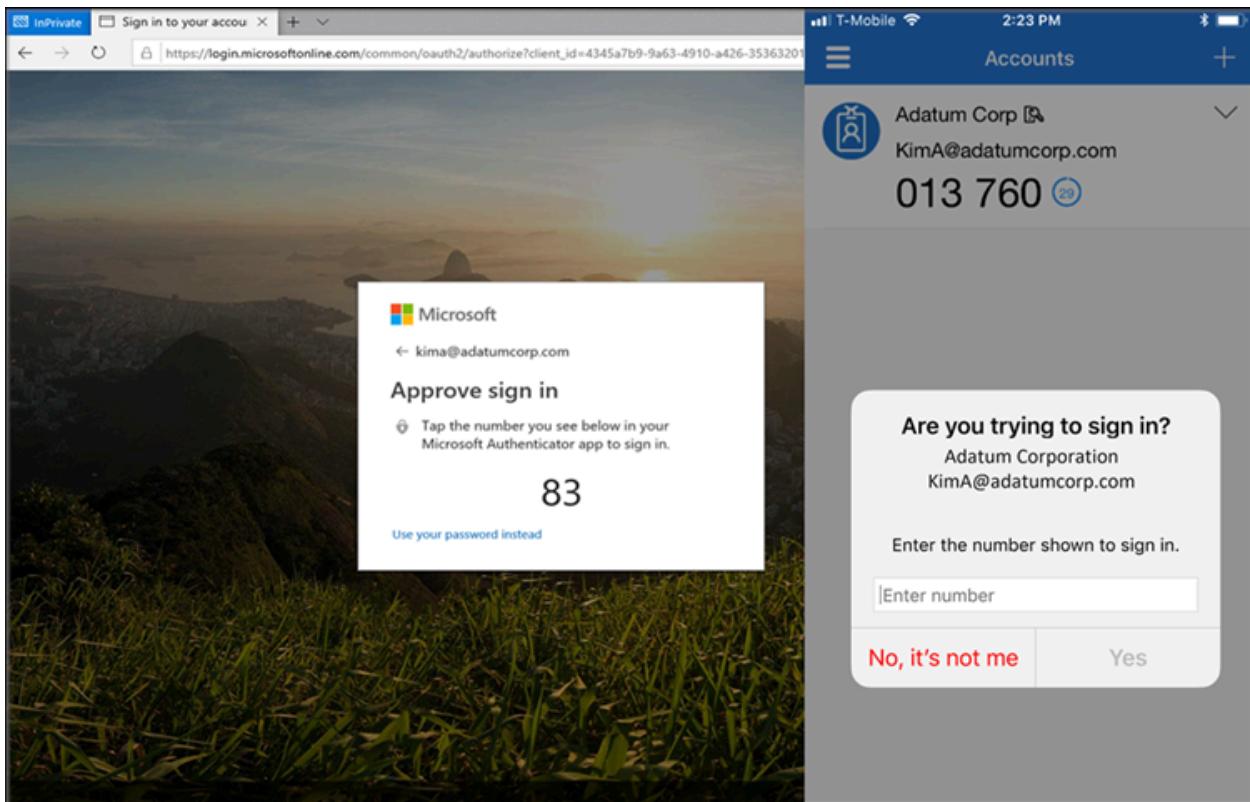
Number matching is available for the following scenarios. When it's enabled, all scenarios support number matching:

- [MFA](#)
- [Self-service password reset \(SSPR\)](#)
- [Combined SSPR and MFA registration during Authenticator app setup](#)
- [Active Directory Federation Services \(AD FS\) adapter](#)
- [Network Policy Server \(NPS\) extension](#)

Number matching isn't supported for push notifications for Apple Watch or Android wearable devices. Wearable device users need to use their phone to approve notifications when number matching is enabled.

Multifactor authentication

When users respond to an MFA push notification by using Authenticator, they see a number. They need to enter that number into the app to complete the approval. For more information about how to set up MFA, see [Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication](#).



SSPR

SSPR with Authenticator requires number matching when a user uses Authenticator. During SSPR, the sign-in page shows a number that the user needs to enter into the Authenticator notification. For more information about how to set up SSPR, see [Tutorial: Enable users to unlock their account or reset passwords](#).

Combined registration

Combined registration with Authenticator requires number matching. When a user goes through combined registration to set up Authenticator, the user needs to approve a notification to add the account. This notification shows a number that the user needs to enter into the Authenticator notification. For more information about how to set up combined registration, see [Enable combined security information registration](#).

AD FS adapter

The AD FS adapter requires number matching on supported versions of Windows Server. On earlier versions, users continue to see the Approve/Deny experience and don't see number matching until they upgrade. The AD FS adapter supports number matching only after they install one of the updates in the following table. For more information about how to set up the AD FS adapter, see [Configure Microsoft Entra Multifactor Authentication Server to work with AD FS in Windows Server](#).

Note

Unpatched versions of Windows Server don't support number matching. Users continue to see the **Approve/Deny** experience and don't see number matching unless these updates are applied.

 Expand table

Version	Update
Windows Server 2022	November 9, 2021—KB5007205 (OS Build 20348.350) ↗
Windows Server 2019	November 9, 2021—KB5007206 (OS Build 17763.2300) ↗
Windows Server 2016	October 12, 2021—KB5006669 (OS Build 14393.4704) ↗

NPS extension

Although NPS doesn't support number matching, the latest NPS extension does support time-based one-time password (TOTP) methods such as the TOTP available in Authenticator, other software tokens, and hardware FOBs. TOTP sign-in provides better security than the alternative **Approve/Deny** experience. Make sure that you run the latest version of the [NPS extension](#) ↗.

Anyone who performs a RADIUS connection with NPS extension version 1.2.2216.1 or later is prompted to sign in with a TOTP method instead of **Approve/Deny**. Users must have a TOTP authentication method registered to see this behavior. Without a TOTP method registered, users continue to see **Approve/Deny**.

Organizations that run any of these earlier versions of the NPS extension can modify the registry to require users to enter a TOTP:

- 1.2.2131.2
- 1.2.1959.1
- 1.2.1916.2
- 1.1.1892.2
- 1.0.1850.1
- 1.0.1.41
- 1.0.1.40

Note

NPS extensions versions earlier than 1.0.1.40 don't support TOTP enforced by number matching. These versions continue to use **Approve/Deny**.

To create the registry entry to override the **Approve/Deny** options in push notifications and require a TOTP instead:

1. On the NPS server, open the Registry Editor.
2. Go to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AzureMfa`.
3. Create the following string/value pair:
 - Name: `OVERRIDE_NUMBER_MATCHING_WITH OTP`
 - Value = `TRUE`
4. Restart the NPS service.

In addition:

- Users who perform TOTP must have either Authenticator registered as an authentication method or some other hardware or software OATH token. Users who can't use a TOTP method always see **Approve/Deny** options with push notifications if they use a version of the NPS extension earlier than 1.2.2216.1.
- The NPS server where the NPS extension is installed must be configured to use the Password Authentication Protocol (PAP). For more information, see [Determine which authentication methods your users can use](#).

Important

MSCHAPv2 doesn't support TOTP. If the NPS server isn't configured to use PAP, user authorization fails with events in the **AuthZOptCh** log of the NPS extension server in Event Viewer:

- NPS extension for Azure MFA: Challenge requested in the Authentication extension for the user `npstesting_ap`.

You can configure the NPS server to support PAP. If PAP isn't an option, set `OVERRIDE_NUMBER_MATCHING_WITH OTP = FALSE` to fall back to **Approve/Deny** push notifications.

If your organization uses Remote Desktop Gateway and the user registered for a TOTP code along with Authenticator push notifications, the user can't meet the Microsoft Entra MFA challenge and Remote Desktop Gateway sign-in fails. In this case, set

`OVERRIDE_NUMBER_MATCHING_WITH OTP = FALSE` to fall back to Approve/Deny push notifications with Authenticator.

FAQs

This section provides answers to common questions.

Can users opt out of number matching?

No, users can't opt out of number matching in Authenticator push notifications.

Does number matching only apply if Authenticator push notifications are set as the default authentication method?

Yes. If the user has a different default authentication method, there's no change to their default sign-in. If the default method is Authenticator push notifications, they get number matching. If the default method is anything else, such as TOTP in Authenticator or another provider, there's no change.

Regardless of their default method, any user who is prompted to sign in with Authenticator push notifications sees number matching. If they're prompted for another method, they won't see any change.

What happens for users who aren't specified in the Authentication methods policy but they're enabled for notifications through the mobile app in the legacy MFA tenant-wide policy?

Users who are enabled for MFA push notifications in the legacy MFA policy also see number match if the legacy MFA policy enabled **Notification through mobile app**. Users see number matching regardless of whether they're enabled for Authenticator in the Authentication methods policy.

verification options (learn more)

Methods available to users:

- Call to phone
- Text message to phone
- Notification through mobile app
- Verification code from mobile app or hardware token

Is number matching supported with Azure Multi-Factor Authentication Server?

No, number matching isn't enforced because it's not a supported feature for Azure Multi-Factor Authentication Server, which is [deprecated](#).

What happens if a user runs an older version of Authenticator?

If a user runs an older version of Authenticator that doesn't support number matching, authentication won't work. They need to upgrade to the latest version of Authenticator to use it for sign-in.

How can users recheck the number on mobile iOS devices after the match request appears?

During mobile iOS broker flows, the number match request appears over the number after a two-second delay. To recheck the number, select **Show me the number again**. This action occurs only in mobile iOS broker flows.

Is Apple Watch supported for Authenticator?

In the Authenticator release in January 2023 for iOS, there's no companion app for watchOS because it's incompatible with Authenticator security features. You can't install or use Authenticator on Apple Watch. We recommend that you [delete Authenticator from your Apple Watch](#) and sign in with Authenticator on another device.

Related content

- [Authentication methods in Microsoft Entra ID](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Use additional context in Authenticator notifications - Authentication methods policy

Article • 03/04/2025

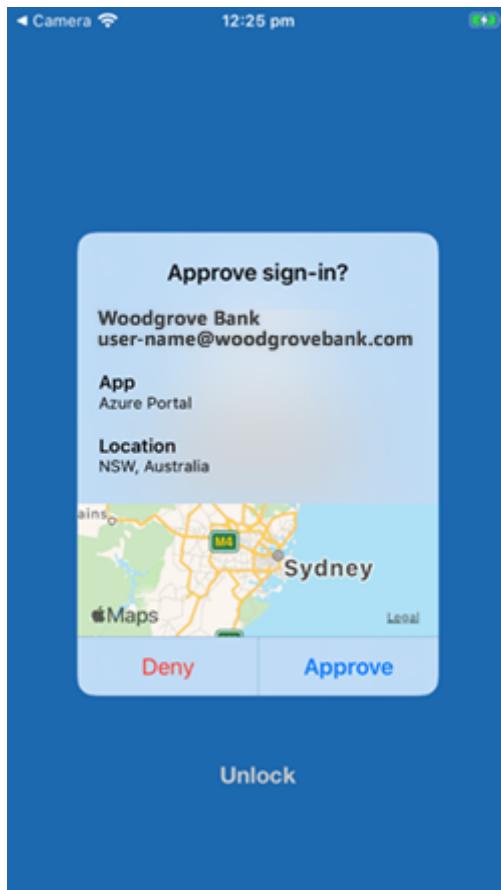
This article discusses how to improve the security of user sign-in by adding the application name and geographic location of the sign-in to Authenticator passwordless and push notifications.

Prerequisites

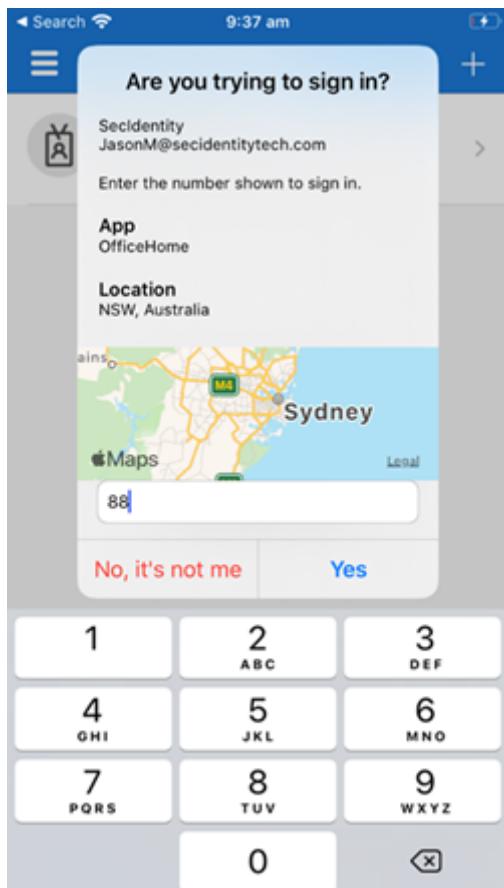
- Your organization needs to enable Authenticator passwordless and push notifications for some users or groups by using the new Authentication methods policy. You can edit the Authentication methods policy by using the Microsoft Entra admin center or Microsoft Graph API.
- Additional context can be targeted to only a single group, which can be dynamic or nested. The group can be synchronized from on-premises or cloud-only.

Passwordless phone sign-in and multifactor authentication

When a user receives a passwordless phone sign-in or multifactor authentication (MFA) push notification in Authenticator, they see the name of the application that requests the approval and the location based on the IP address from where the sign-in originated.



Admins can combine additional context with [number matching](#) to further improve sign-in security.



Policy schema changes

You can enable and disable the application name and geographic location separately.

Under `featureSettings`, you can use the following name mapping for each feature:

- **Application name:** `displayAppInformationRequiredState`
- **Geographic location:** `displayLocationInformationRequiredState`

 **Note**

Make sure that you use the new policy schema for Microsoft Graph APIs. In Graph Explorer, you need to consent to the `Policy.Read.All` and `Policy.ReadWrite.AuthenticationMethod` permissions.

Identify your single target group for each of the features. Then use the following API endpoint to change `displayAppInformationRequiredState` or `displayLocationInformationRequiredState` properties under `featureSettings` to `enabled` and include or exclude the groups you want:

```
msgraph
```

```
GET  
https://graph.microsoft.com/v1.0/authenticationMethodsPolicy/authenticationMethodConfigurations/MicrosoftAuthenticator
```

For more information, see [microsoftAuthenticatorAuthenticationMethodConfiguration resource type](#).

Example of how to enable additional context for all users

In `featureSettings`, change `displayAppInformationRequiredState` and `displayLocationInformationRequiredState` from `default` to `enabled`.

The value of Authentication mode is either `any` or `push`, depending on whether or not you also want to enable passwordless phone sign-in. In these examples, we use `any`, but if you don't want to allow passwordless, use `push`.

You might need to `PATCH` the entire schema to prevent overwriting any previous configuration. In that case, do a `GET` first. Then update only the relevant fields and then `PATCH`. The following example shows how to update `displayAppInformationRequiredState` and `displayLocationInformationRequiredState` under `featureSettings`.

Only users who are enabled for Authenticator under `includeTargets` see the application name or geographic location. Users who aren't enabled for Authenticator don't see these features.

JSON

```
//Retrieve your existing policy via a GET.  
//Leverage the Response body to create the Request body section. Then update  
the Request body similar to the Request body as shown below.  
//Change the Query to PATCH and Run query  
  
{  
    "@odata.context":  
    "https://graph.microsoft.com/v1.0/$metadata#authenticationMethodConfigurations/$entity",  
    "@odata.type":  
    "#microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration",  
    "id": "MicrosoftAuthenticator",  
    "state": "enabled",  
    "featureSettings": {  
        "displayAppInformationRequiredState": {  
            "state": "enabled",  
            "includeTarget": {  
                "targetType": "group",  
                "id": "all_users"  
            },  
            "excludeTarget": {  
                "targetType": "group",  
                "id": "00000000-0000-0000-0000-000000000000"  
            }  
        },  
        "displayLocationInformationRequiredState": {  
            "state": "enabled",  
            "includeTarget": {  
                "targetType": "group",  
                "id": "all_users"  
            },  
            "excludeTarget": {  
                "targetType": "group",  
                "id": "00000000-0000-0000-0000-000000000000"  
            }  
        }  
    },  
    "includeTargets@odata.context":  
    "https://graph.microsoft.com/v1.0/$metadata#authenticationMethodsPolicy/authenticationMethodConfigurations('MicrosoftAuthenticator')/microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration/includeTargets",  
    "includeTargets": [  
        {  
            "targetType": "group",  
            "id": "all_users",  
            "isRegistrationRequired": false,  
            "authenticationMode": "any",  
        }  
    ]  
}
```

```
        }
    ]
}
```

Example of how to enable application name and geographic location for separate groups

In `featureSettings`, change `displayAppInformationRequiredState` and `displayLocationInformationRequiredState` from `default` to `enabled`. Inside `includeTarget` for each `featureSetting`, change the ID from `all_users` to the object ID of the group from the Microsoft Entra admin center.

You need to `PATCH` the entire schema to prevent overwriting any previous configuration. We recommend that you do a `GET` first. Then update only the relevant fields and then `PATCH`. The following example shows an update to `displayAppInformationRequiredState` and `displayLocationInformationRequiredState` under `featureSettings`.

Only users who are enabled for Authenticator under `includeTargets` see the application name or geographic location. Users who aren't enabled for Authenticator don't see these features.

JSON

```
{
    "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#authenticationMethodConfigurations/$entity",
    "@odata.type": "#microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration",
    "id": "MicrosoftAuthenticator",
    "state": "enabled",
    "featureSettings": {
        "displayAppInformationRequiredState": {
            "state": "enabled",
            "includeTarget": {
                "targetType": "group",
                "id": "44561710-f0cb-4ac9-ab9c-e6c394370823"
            },
            "excludeTarget": {
                "targetType": "group",
                "id": "00000000-0000-0000-0000-000000000000"
            }
        },
        "displayLocationInformationRequiredState": {
            "state": "enabled",
            "includeTarget": {
                "targetType": "group",
                "id": "a229e768-961a-4401-aadb-11d836885c11"
            }
        }
    }
}
```

```
        },
        "excludeTarget": {
            "targetType": "group",
            "id": "00000000-0000-0000-0000-000000000000"
        }
    },
    "includeTargets@odata.context": "https://graph.microsoft.com/v1.0/$metadata#authenticationMethodsPolicy/authenticationMethodConfigurations('MicrosoftAuthenticator')/microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration/includeTargets",
    "includeTargets": [
        {
            "targetType": "group",
            "id": "all_users",
            "isRegistrationRequired": false,
            "authenticationMode": "any",
        }
    ]
}
```

To verify, run `GET` again and verify the object ID:

```
msgraph
GET
https://graph.microsoft.com/v1.0/authenticationMethodsPolicy/authenticationMethodConfigurations/MicrosoftAuthenticator
```

Example of how to disable the application name and only enable the geographic location

In `featureSettings`, change the state of `displayAppInformationRequiredState` to `default` or `disabled` and `displayLocationInformationRequiredState` to `enabled`. Inside `includeTarget` for each `featureSetting` value, change the ID from `all_users` to the object ID of the group from the Microsoft Entra admin center.

You need to `PATCH` the entire schema to prevent overwriting any previous configuration. We recommend that you do a `GET` first. Then update only the relevant fields and then `PATCH`. The following example shows an update to `displayAppInformationRequiredState` and `displayLocationInformationRequiredState` under `featureSettings`.

Only users who are enabled for Authenticator under `includeTargets` see the application name or geographic location. Users who aren't enabled for Authenticator don't see these features.

JSON

```
{  
    "@odata.context":  
        "https://graph.microsoft.com/v1.0/$metadata#authenticationMethodConfigurations/$entity",  
    "@odata.type":  
        "#microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration",  
    "id": "MicrosoftAuthenticator",  
    "state": "enabled",  
    "featureSettings": {  
        "displayAppInformationRequiredState": {  
            "state": "disabled",  
            "includeTarget": {  
                "targetType": "group",  
                "id": "44561710-f0cb-4ac9-ab9c-e6c394370823"  
            },  
            "excludeTarget": {  
                "targetType": "group",  
                "id": "00000000-0000-0000-0000-000000000000"  
            }  
        },  
        "displayLocationInformationRequiredState": {  
            "state": "enabled",  
            "includeTarget": {  
                "targetType": "group",  
                "id": "a229e768-961a-4401-aadb-11d836885c11"  
            },  
            "excludeTarget": {  
                "targetType": "group",  
                "id": "00000000-0000-0000-0000-000000000000"  
            }  
        }  
    },  
    "includeTargets@odata.context":  
        "https://graph.microsoft.com/v1.0/$metadata#authenticationMethodsPolicy/authenticationMethodConfigurations('MicrosoftAuthenticator')/microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration/includeTargets",  
    "includeTargets": [  
        {  
            "targetType": "group",  
            "id": "all_users",  
            "isRegistrationRequired": false,  
            "authenticationMode": "any",  
        }  
    ]  
}
```

Example of how to exclude a group from the application name and geographic location

In addition, for each of the features, you change the ID of `excludeTarget` to the object ID of the group from the Microsoft Entra admin center. This change excludes that group from seeing the application name or geographic location.

You need to `PATCH` the entire schema to prevent overwriting any previous configuration. We recommend that you do a `GET` first. Then update only the relevant fields and then `PATCH`. The following example shows an update to `displayAppInformationRequiredState` and `displayLocationInformationRequiredState` under `featureSettings`.

Only users who are enabled for Authenticator under `includeTargets` see the application name or geographic location. Users who aren't enabled for Authenticator don't see these features.

JSON

```
{  
    "@odata.context":  
        "https://graph.microsoft.com/v1.0/$metadata#authenticationMethodConfigurations/$entity",  
    "@odata.type":  
        "#microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration",  
    "id": "MicrosoftAuthenticator",  
    "state": "enabled",  
    "featureSettings": {  
        "displayAppInformationRequiredState": {  
            "state": "enabled",  
            "includeTarget": {  
                "targetType": "group",  
                "id": "44561710-f0cb-4ac9-ab9c-e6c394370823"  
            },  
            "excludeTarget": {  
                "targetType": "group",  
                "id": "5af8a0da-5420-4d69-bf3c-8b129f3449ce"  
            }  
        },  
        "displayLocationInformationRequiredState": {  
            "state": "enabled",  
            "includeTarget": {  
                "targetType": "group",  
                "id": "a229e768-961a-4401-aadb-11d836885c11"  
            },  
            "excludeTarget": {  
                "targetType": "group",  
                "id": "b6bab067-5f28-4dac-ab30-7169311d69e8"  
            }  
        },  
        "includeTargets@odata.context":  
            "https://graph.microsoft.com/v1.0/$metadata#authenticationMethodsPolicy/authenticationMethodConfigurations('MicrosoftAuthenticator')/microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration/includeTargets",
```

```
"includeTargets": [
  {
    "targetType": "group",
    "id": "all_users",
    "isRegistrationRequired": false,
    "authenticationMode": "any",
  }
]
}
```

Example of removing the excluded group

In `featureSettings`, change the states of `displayAppInformationRequiredState` from `default` to `enabled`. Change the ID of `excludeTarget` to `00000000-0000-0000-0000-000000000000`.

You need to `PATCH` the entire schema to prevent overwriting any previous configuration. We recommend that you do a `GET` first. Then update only the relevant fields and then `PATCH`. The following example shows an update to `displayAppInformationRequiredState` and `displayLocationInformationRequiredState` under `featureSettings`.

Only users who are enabled for Authenticator under `includeTargets` see the application name or geographic location. Users who aren't enabled for Authenticator don't see these features.

JSON

```
{
  "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#authenticationMethodConfigurations/$entity",
  "@odata.type": "#microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration",
  "id": "MicrosoftAuthenticator",
  "state": "enabled",
  "featureSettings": {
    "displayAppInformationRequiredState": {
      "state": "enabled",
      "includeTarget": {
        "targetType": "group",
        "id": "1ca44590-e896-4dbe-98ed-b140b1e7a53a"
      },
      "excludeTarget": {
        "targetType": "group",
        "id": "00000000-0000-0000-0000-000000000000"
      }
    }
  }
},
```

```

    "includeTargets@odata.context": 
"https://graph.microsoft.com/v1.0/$metadata#authenticationMethodsPolicy/authenticationMethodConfigurations('MicrosoftAuthenticator')/microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration/includeTargets",
    "includeTargets": [
        {
            "targetType": "group",
            "id": "all_users",
            "isRegistrationRequired": false,
            "authenticationMode": "any"
        }
    ]
}

```

Turn off additional context

To turn off additional context, you need to `PATCH` `displayAppInformationRequiredState` and `displayLocationInformationRequiredState` from `enabled` to `disabled/default`. You can also turn off only one of the features.

JSON

```
{
    "@odata.context": 
"https://graph.microsoft.com/v1.0/$metadata#authenticationMethodConfigurations/$entity",
    "@odata.type": 
"#microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration",
    "id": "MicrosoftAuthenticator",
    "state": "enabled",
    "featureSettings": {
        "displayAppInformationRequiredState": {
            "state": "disabled",
            "includeTarget": {
                "targetType": "group",
                "id": "44561710-f0cb-4ac9-ab9c-e6c394370823"
            },
            "excludeTarget": {
                "targetType": "group",
                "id": "00000000-0000-0000-0000-000000000000"
            }
        },
        "displayLocationInformationRequiredState": {
            "state": "disabled",
            "includeTarget": {
                "targetType": "group",
                "id": "a229e768-961a-4401-aadb-11d836885c11"
            },
            "excludeTarget": {
                "targetType": "group",
                "id": "00000000-0000-0000-0000-000000000000"
            }
        }
    }
}
```

```

        }
    },
},
"includeTargets@odata.context": "https://graph.microsoft.com/v1.0/$metadata#authenticationMethodsPolicy/authenticationMethodConfigurations('MicrosoftAuthenticator')/microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration/includeTargets",
"includeTargets": [
{
    "targetType": "group",
    "id": "all_users",
    "isRegistrationRequired": false,
    "authenticationMode": "any",
}
]
}

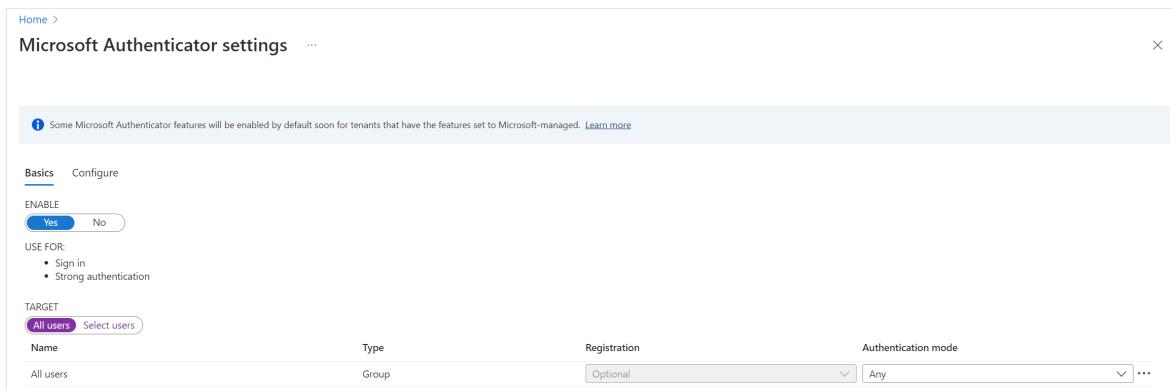
```

Enable additional context in the Microsoft Entra admin center

To enable the application name or geographic location in the Microsoft Entra admin center, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Microsoft Authenticator**.
3. On the **Basics** tab, select **Yes** and **All users** to enable the policy for everyone. Change **Authentication mode** to **Any**.

Only users who are enabled for Authenticator here are included in the policy to show the application name or geographic location of the sign-in, or excluded from it. Users who aren't enabled for Authenticator can't see the application name or geographic location.



4. On the **Configure** tab, for **Show application name in push and passwordless notifications**, change **Status** to **Enabled**. Choose who to include or exclude from the policy, and then select **Save**.

Show application name in push and passwordless notifications

Note: If the feature status is set to Microsoft-managed, it will be enabled by Microsoft at an appropriate time after the preview. [Learn more](#)

Status

Target All users Select group

Then do the same for **Show geographic location in push and passwordless notifications**.

Show geographic location in push and passwordless notifications

Note: If the feature status is set to Microsoft-managed, it will be enabled by Microsoft at an appropriate time after the preview. [Learn more](#)

Status

Target All users Select group

You can configure the application name and geographic location separately. For example, the following policy enables the application name and geographic location for all users but excludes the Operations group from seeing the geographic location.

Show application name in push and passwordless notifications

Note: If the feature status is set to Microsoft-managed, it will be enabled by Microsoft at an appropriate time after the preview. [Learn more](#)

Status

Target All users Select group

Show geographic location in push and passwordless notifications

Note: If the feature status is set to Microsoft-managed, it will be enabled by Microsoft at an appropriate time after the preview. [Learn more](#)

Status

Target Exclude Include

Add selected group

Exclude target

Operations group

Save **Discard**

Known issues

- Additional context isn't supported for Network Policy Server (NPS) or Active Directory Federation Services.
- Users can modify the location reported by iOS and Android devices. As a result, Authenticator is updating its security baseline for Location-Based Access Control (LBAC) Conditional Access policies. Authenticator denies authentications where the user might be using a different location than the actual GPS location of the mobile device where Authenticator is installed.

In the November 2023 release of Authenticator, users who modify the location of their device see a denial message in Authenticator when they do an LBAC authentication. Beginning in January 2024, any users who run older Authenticator versions are blocked from LBAC authentication with a modified location:

- Authenticator version 6.2309.6329 or earlier on Android
- Authenticator version 6.7.16 or earlier on iOS

To find which users run older versions of Authenticator, use [Microsoft Graph APIs](#).

Related content

- [Authentication methods in Microsoft Entra ID - Microsoft Authenticator app](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Enable Authenticator Lite for Outlook mobile

Article • 03/04/2025

Authenticator Lite is another surface for Microsoft Entra users to complete multifactor authentication (MFA) by using push notifications or time-based one-time passcodes (TOTP) on your Android or iOS device. With Authenticator Lite, users can satisfy an MFA requirement from the convenience of a familiar app. Authenticator Lite is currently enabled in [Outlook mobile](#).

Users receive a notification in Outlook mobile to approve or deny sign-in, or you can copy a TOTP to use during sign-in.

ⓘ Note

Use these important security enhancements if you're authenticating via telecom transports:

- The Microsoft-managed value of this feature is enabled in the Authentication methods policy. If you don't want to enable this feature, move the state from **Default** to **Disabled**, or scope it to only a group of users.
- Authenticator Lite is enabled as part of the Notification through mobile app verification option in the per-user MFA policy. If you don't want this feature enabled, you can disable it in the Authentication methods policy by following the steps in this article.

Prerequisites

- Your organization needs to enable Authenticator (second factor) push notifications for all users or select groups. We recommend that you enable Authenticator by using the modern [Authentication methods policy](#). You can edit the Authentication methods policy by using the Microsoft Entra admin center or Microsoft Graph API. Authenticator Lite isn't eligible for on-premises user accounts or organizations with an active MFA server.

💡 Tip

We recommend that you also enable [system-preferred MFA](#) when you enable Authenticator Lite. With system-preferred MFA enabled, users try to sign in with Authenticator Lite before they try less secure telephony methods like SMS or voice call.

- If your organization is using the Active Directory Federation Services (AD FS) adapter or Network Policy Server (NPS) extensions, upgrade to the latest versions for a consistent experience.
- Users enabled for shared device mode on Outlook mobile aren't eligible for Authenticator Lite.
- Users must run a minimum Outlook mobile version.

[\[+\] Expand table](#)

Operating system	Outlook version
Android	4.2310.1
iOS	4.2312.1

Enable Authenticator Lite

By default, Authenticator Lite is [Microsoft managed](#) in the Authentication methods policy. On June 26, the Microsoft-managed value of this feature changed from `disabled` to `enabled`. Authenticator Lite is also included as part of the [Notification through mobile app](#) verification option in the per-user MFA policy.

Disable Authenticator Lite in the Microsoft Entra admin center

To disable Authenticator Lite in the Microsoft Entra admin center, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Microsoft Authenticator**.
3. On the **Enable and Target** tab, select **Enable** and **All users** to enable the Authenticator policy for everyone, or add select groups. Set the Authentication mode for these users or groups to **Any or Push**.

Users who aren't enabled for Authenticator can't see the feature. Users who have Authenticator downloaded on the same device on which Outlook is downloaded aren't prompted to register for Authenticator Lite in Outlook. Android users who use a personal and work profile on their device might be prompted to register if Authenticator is present on a different profile from the Outlook application.

The Microsoft Authenticator app is a flagship authentication method, usable in passwordless or simple push notification approval modes. The app is free to download and use on Android/iOS mobile devices. [Learn more](#).

Enable and Target Configure

Enable

Include Exclude

Target All users Select groups

Name	Type	Registration	Authentication mode
All users	Group	Optional	Any

- On the **Configure** tab, for **Microsoft Authenticator on companion applications**, change **Status** to **Disabled**, and then select **Save**.

Microsoft Authenticator on companion applications

Note: If the feature status is set to Microsoft-managed, it will be enabled by Microsoft at an appropriate time after the preview. [Learn more](#)

Status

Target Include Exclude

All users Select group

Note: If the feature status is set to Microsoft-managed, it will be enabled by Microsoft at an appropriate time after the preview. [Learn more](#)

If your organization still manages authentication methods in the per-user MFA policy, you need to disable **Notification through mobile app** as a verification option there in addition to the preceding steps. We recommend that you do this step only after you enable Authenticator in the Authentication methods policy.

You can continue to manage the remainder of your authentication methods in the per-user MFA policy while Authenticator is managed in the modern Authentication methods policy. However, we recommend that you [migrate](#) management of all authentication methods to the modern Authentication methods policy. The ability to manage authentication methods in the per-user MFA policy retires on September 30, 2025.

Enable Authenticator Lite via Graph APIs

Expand table

Property	Type	Description
excludeTarget	featureTarget	A single entity that's excluded from this feature. You can exclude only one group from Authenticator Lite, which can be a dynamic or nested group.

Property	Type	Description
includeTarget	featureTarget	A single entity that's included in this feature. You can include only one group for Authenticator Lite, which can be a dynamic or nested group.
State	advancedConfigState	Possible values: Enabled explicitly enables the feature for the selected group. Disabled explicitly disables the feature for the selected group. Default allows Microsoft Entra ID to manage whether the feature is enabled or not for the selected group.

After you identify the single target group, use the following API endpoint to change the `CompanionAppsAllowedState` property under `featureSettings`.

HTTP

```
https://graph.microsoft.com/beta/authenticationMethodsPolicy/authenticationMethodConfigurations/MicrosoftAuthenticator
```

In Graph Explorer, you need to consent to the `Policy.ReadWrite.AuthenticationMethod` permission.

Request

JSON

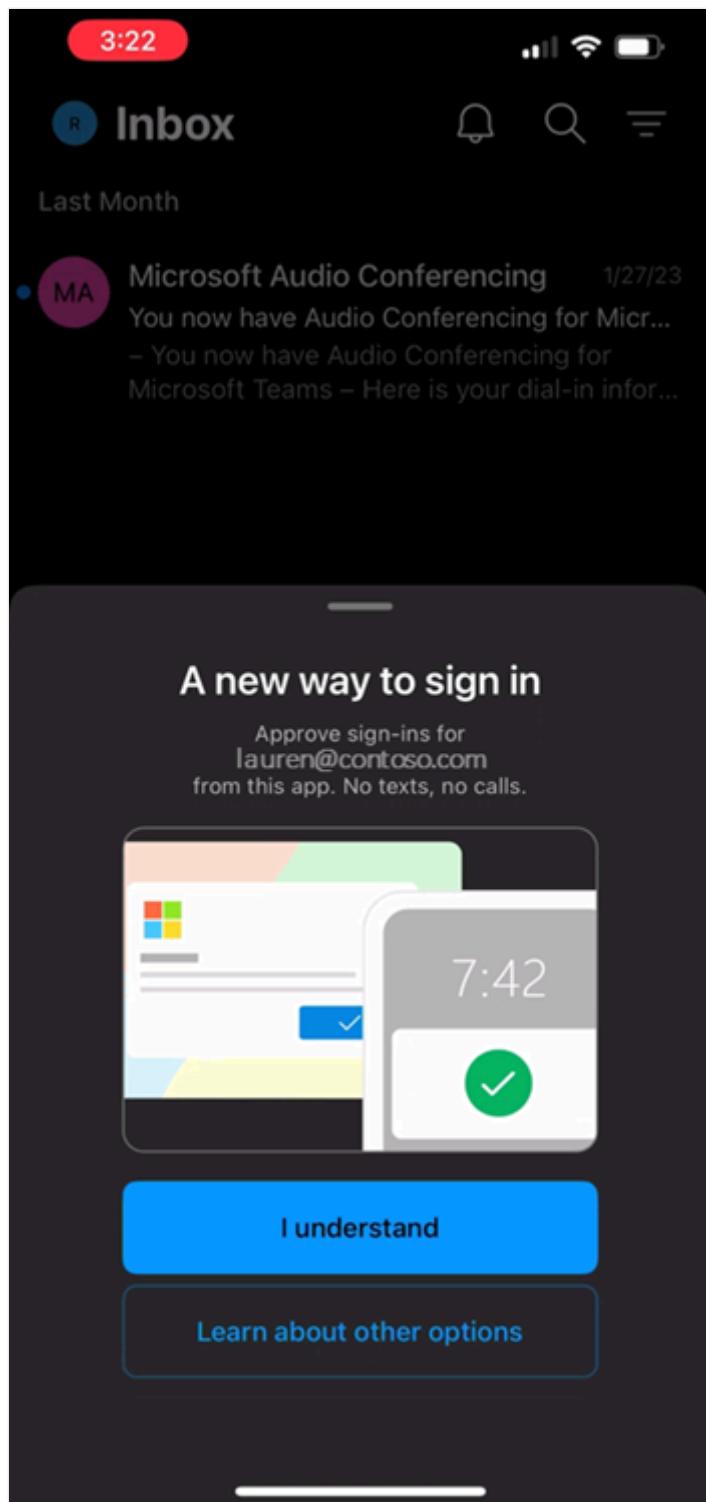
```
//Retrieve your existing policy via a GET.
//Leverage the Response body to create the Request body section. Then update
the Request body similar to the Request body as shown below.
//Change the query to PATCH and run the query.

{
    "@odata.context": "https://graph.microsoft.com/beta/$metadata#authenticationMethodConfigurations/$entity",
    "@odata.type": "#microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration",
    "id": "MicrosoftAuthenticator",
    "state": "enabled",
    "isSoftwareOathEnabled": false,
    "excludeTargets": [],
    "featureSettings": {
        "companionAppAllowedState": {
            "state": "enabled",
            "includeTarget": {
                "targetType": "group",
                "id": "group1"
            }
        }
    }
}
```

```
        "id": "s4432809-3bql-5m21-0p42-8rq4707rq36m"
    },
    "excludeTarget": {
        "targetType": "group",
        "id": "00000000-0000-0000-0000-000000000000"
    }
}
},
"includeTargets@odata.context": "https://graph.microsoft.com/beta/$metadata#authenticationMethodsPolicy/authenticationMethodConfigurations('MicrosoftAuthenticator')/microsoft.graph.microsoftAuthenticatorAuthenticationMethodConfiguration/includeTargets",
"includeTargets": [
{
    "targetType": "group",
    "id": "all_users",
    "isRegistrationRequired": false,
    "authenticationMode": "any"
}
]
}
```

User registration

If users are enabled for Authenticator Lite, they're prompted to register your account directly from Outlook mobile. Authenticator Lite registration isn't available by using [My Sign-Ins](#). Users can also enable or disable Authenticator Lite from within Outlook mobile. For more information about the user experience, see [Authenticator Lite support](#).



If users don't have any MFA methods registered, they're prompted to download Authenticator when they begin the registration flow. For the most seamless experience, provision users with a [Temporary Access Pass \(TAP\)](#) during Authenticator Lite registration.

Monitor Authenticator Lite usage

[Sign-in logs](#) can show which app was used to complete user authentication. To view the latest sign-ins, use the following call on the beta API endpoint:

HTTP

GET auditLogs/signIns

If the sign-in was done by phone app notification, under `authenticationAppDeviceDetails` the `clientApp` field returns `microsoftAuthenticator` or **Outlook**.

If a user has registered Authenticator Lite, the user's registered authentication methods include **Microsoft Authenticator (in Outlook)**.

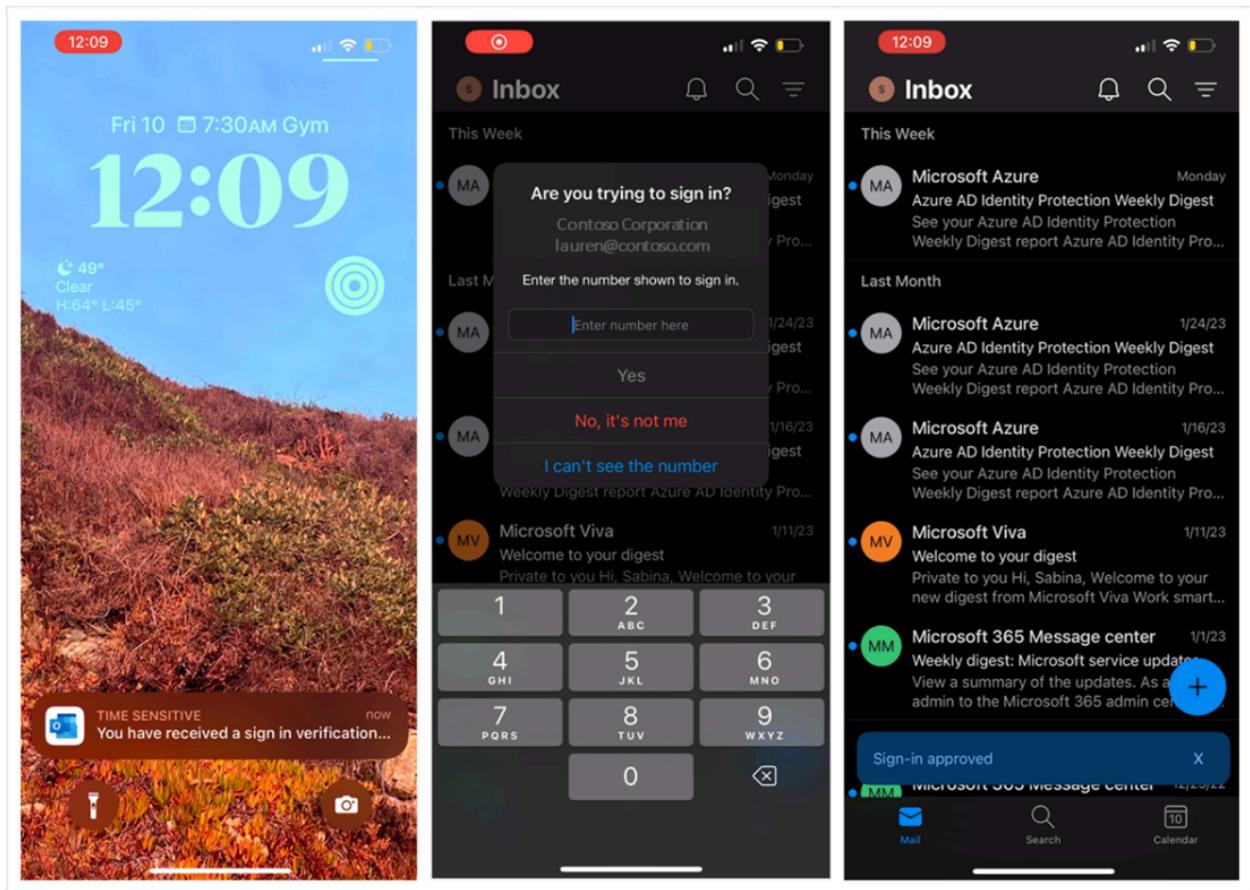
Push notifications in Authenticator Lite

Push notifications sent by Authenticator Lite aren't configurable and don't depend on the Authenticator feature settings. Authenticator Lite doesn't support passwordless authentication mode. The following table lists the settings for features included in the Authenticator Lite experience. Every authentication includes a number matching prompt and doesn't include app and location context, regardless of Authenticator feature settings.

 Expand table

Authenticator feature	Authenticator Lite experience
Number matching	Enabled
Location context	Disabled
Application context	Disabled

The following screenshots show what users see when Authenticator Lite sends a push notification.



AD FS adapter and NPS extension

Authenticator Lite enforces number matching in every authentication. If your tenant is using an AD FS adapter or an NPS extension, your users might not be able to complete Authenticator Lite notifications. For more information, see [AD FS adapter](#) and [NPS extension](#).

To learn more about verification notifications, see [Microsoft Authenticator authentication method](#).

Common questions

The following sections list common questions.

Does Authenticator Lite work as a broker app?

No, Authenticator Lite is available only for push notifications and TOTP.

Can Authenticator Lite be used for SSPR?

No, Authenticator Lite is available only for push notifications and TOTP.

Is Authenticator Lite available in the Outlook desktop app?

No, Authenticator Lite is available only on Outlook mobile.

Where can users register for Authenticator Lite?

Users can register for Authenticator Lite only from mobile Outlook. Authenticator Lite registration is managed from [My Sign-Ins](#).

Can users register Authenticator and Authenticator Lite?

Users who have Authenticator on their device can't register Authenticator Lite on that same device. If a user has an Authenticator Lite registration and then later downloads Authenticator, they can register both. If a user has two devices, they can register Authenticator Lite on one and Authenticator on the other.

Known issues

The following issues are known.

SSPR notifications

TOTP codes from Outlook work for SSPR, but the push notification won't work and returns an error.

Logs are showing added Conditional Access evaluations

The Conditional Access policies are evaluated each time a user opens their Outlook app to determine whether they're eligible to register for Authenticator Lite. These checks might appear in logs.

Related content

- [Authentication methods in Microsoft Entra ID](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

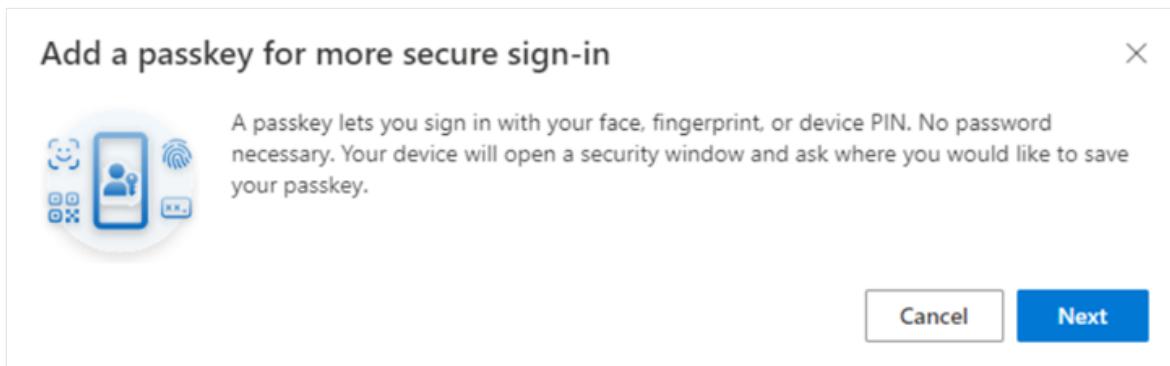
Register a passkey (FIDO2)

Article • 03/04/2025

This article shows how to register a passkey as an authentication method.

First-time registration

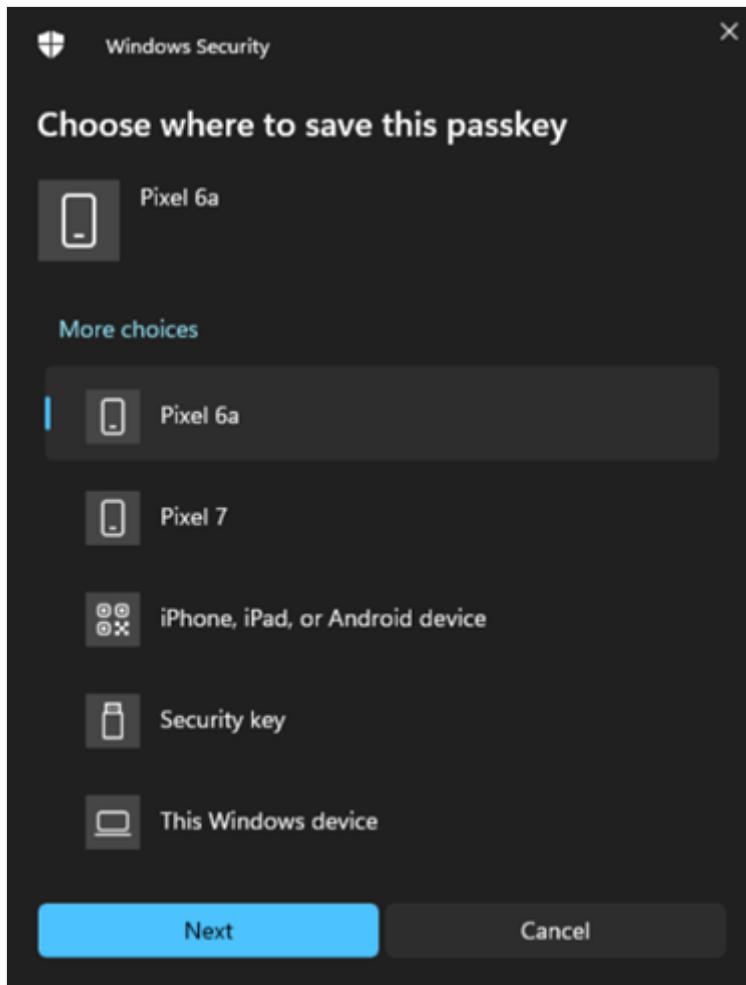
1. First-time users need to register a passkey (FIDO2) as an authentication method by navigating and completing the process from a browser at [Security info](#).
2. Tap **Add sign-in method > Choose a method > Passkey > Add**.
3. Sign in with multifactor authentication (MFA) before adding a passkey (FIDO2), then tap **Next**.



4. Select where you want to save your passkey (FIDO2).

ⓘ Note

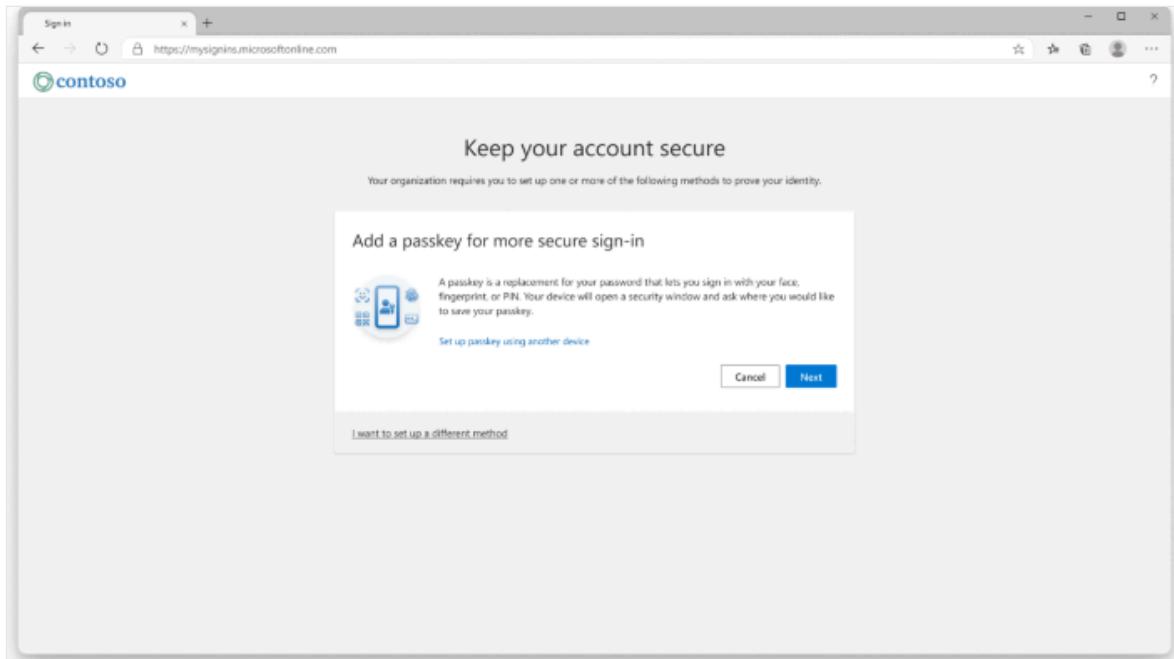
Options displayed vary depending on your browser and device operating system. If the device where you started the registration process supports passkeys (FIDO2), you'll be asked to save the passkey to that device. Select **Use another device** or **More options** to display additional ways for you to save the passkey.



5. (Optional) If you previously set up a passkey (FIDO2) on a mobile device and selected the option to remember that device for quicker sign-in, the device name may appear as a selectable option. In this case, do the following steps:
 - a. Choose **Security key**.
 - b. Follow the prompts to connect your security key and provide a PIN or biometric method.
 - c. Upon completion, you're redirected to the **My Security info** screen, where you can change the default name for the new sign-in method.
 - d. Select **Done** to finish registering the new method.

Prompted registration

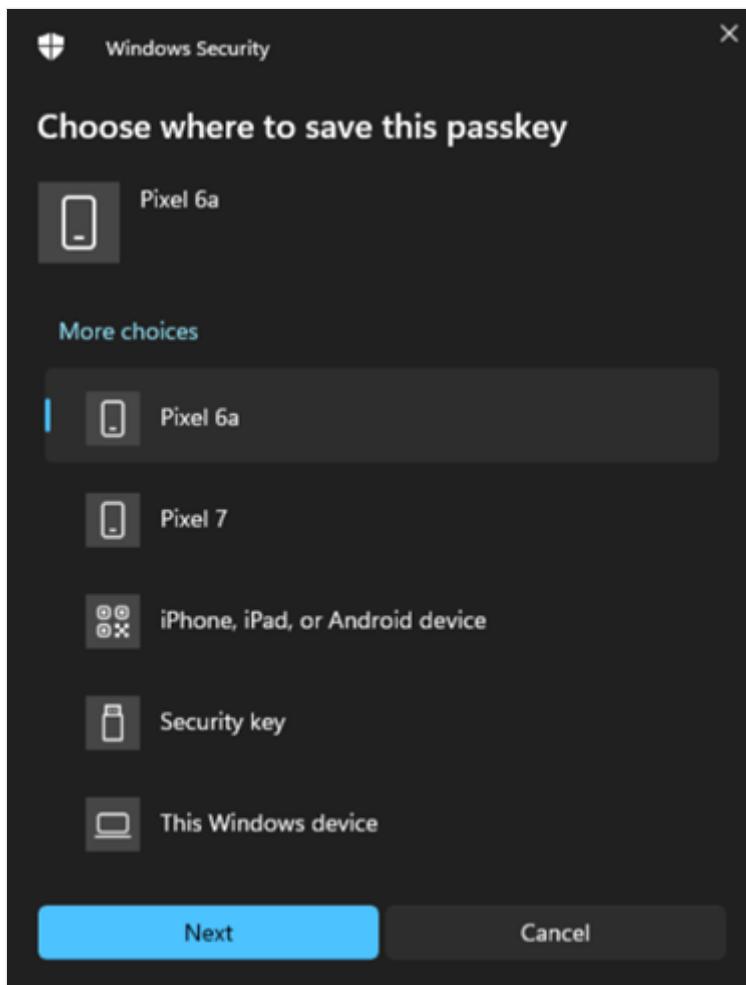
1. If your organization requires you to register a passkey (FIDO2), you'll be prompted after sign-in to add a passkey (FIDO2).



2. Tap **Next**, then you're directed to `login.microsoft.com`.
3. Select where you would like to save your passkey (FIDO2).

! **Note**

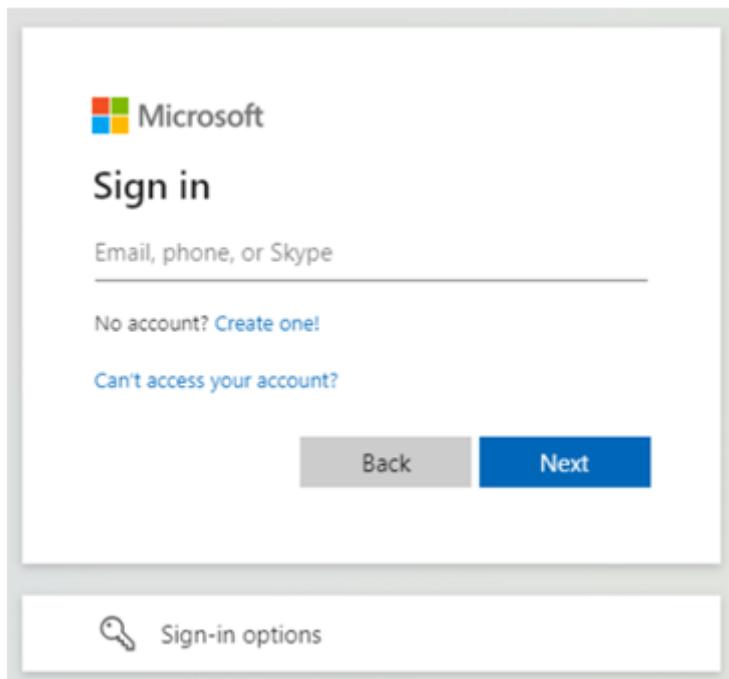
Options displayed vary depending on your browser and device operating system. If the device where you started the registration process supports passkeys (FIDO2), you'll be asked to save the passkey (FIDO2) to that device. Select **Use another device** or **More options** displays additional ways for you to save the passkey.



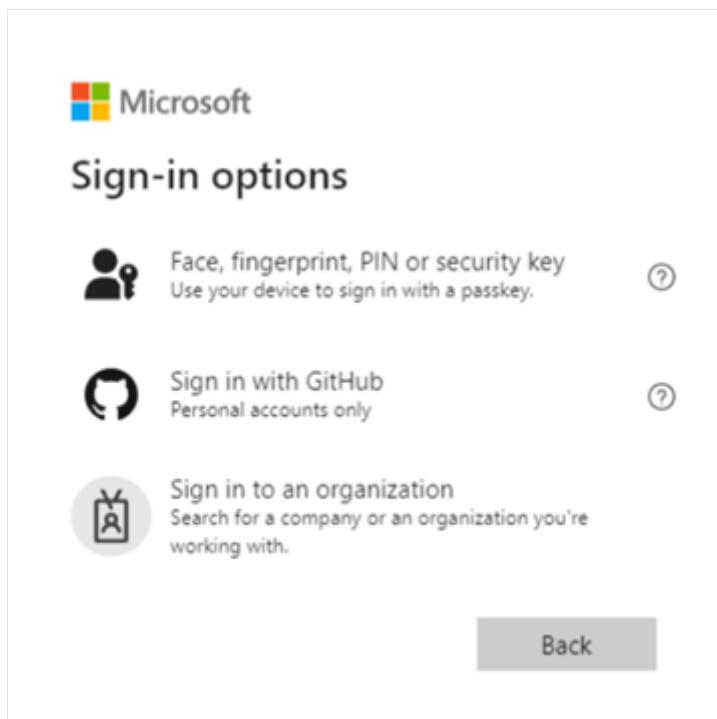
4. (Optional) If you previously set up a passkey (FIDO2) on a mobile device and selected the option to remember that device for quicker sign-in, the device name may appear as a selectable option. In this case, do the following steps:
 - a. Choose **Security key**.
 - b. Follow the prompts to connect your security key and provide a PIN or biometric method.
 - c. Upon completion, you're redirected to the **My Security info** screen, where you can change the default name for the new sign-in method.
 - d. Select **Done** to finish registering the new method.

Sign-in with your new passkey (FIDO2)

1. Navigate to login.microsoftonline.com.



2. Select Sign-in options.



3. Choose Face, Fingerprint, PIN, or Security key.

4. A security window opens. Follow the remaining prompts to sign-in with the method that you selected.

Next steps

- Choosing authentication methods for your organization
- Register security keys on behalf of users

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Enable FIDO2 security key sign-in to Windows 10 and 11 devices with Microsoft Entra ID

Article • 03/04/2025

This document focuses on enabling FIDO2 security key based passwordless authentication with Windows 10 and 11 devices. After completing the steps in this article, you're able to sign in to both your Microsoft Entra ID and Microsoft Entra hybrid joined Windows devices with your Microsoft Entra account using a FIDO2 security key.

Requirements

[+] Expand table

Device Type	Microsoft Entra joined	Microsoft Entra hybrid joined
Microsoft Entra multifactor authentication	X	X
Combined security information registration	X	X
Compatible FIDO2 security keys	X	X
WebAuthN requires Windows 10 version 1903 or higher	X	X
Microsoft Entra joined devices require Windows 10 version 1909 or higher	X	
Microsoft Entra hybrid joined devices require Windows 10 version 2004 or higher		X
Fully patched Windows Server 2016/2019 Domain Controllers.		X
Microsoft Entra Hybrid Authentication Management module ↗		X
Microsoft Intune (Optional)	X	X
Provisioning package (Optional)	X	X
Group Policy (Optional)		X

Unsupported scenarios

The following scenarios aren't supported:

- Signing in or unlocking a Windows device with a passkey in Microsoft Authenticator.
- Windows Server Active Directory Domain Services (AD DS) domain-joined (on-premises only devices) deployment.
- Scenarios, such as RDP, VDI, and Citrix, that use a security key other than [webauthn redirection](#).
- S/MIME using a security key.
- *Run as* using a security key.
- Signing in to a server using a security key.
- If you're not using a security key to sign in to your device while online, you can't use it to sign in or unlock offline.
- Signing in or unlocking a Windows device with a security key containing multiple Microsoft Entra accounts. This scenario utilizes the last account added to the security key. WebAuthN allows users to choose the account they wish to use.
- Unlocking a device running Windows 10 version 1809. For the best experience, use Windows 10 version 1903 or higher.

Prepare devices

Microsoft Entra joined devices must run Windows 10 version 1909 or higher.

Microsoft Entra hybrid joined devices must run Windows 10 version 2004 or newer.

Enable security keys for Windows sign-in

Organizations can choose to use one or more of the following methods to enable the use of security keys for Windows sign-in based on their organization's requirements:

- [Enable with Microsoft Intune](#)
- [Targeted Microsoft Intune deployment](#)
- [Enable with a provisioning package](#)
- [Enable with Group Policy \(Microsoft Entra hybrid joined devices only\)](#)

i Important

Organizations with Microsoft Entra hybrid joined devices must also complete the steps in the article, [Enable FIDO2 authentication to on-premises resources](#) before

Windows 10 FIDO2 security key authentication works.

Organizations with Microsoft Entra joined devices must do this before their devices can authenticate to on-premises resources with FIDO2 security keys.

Enable with Microsoft Intune

To enable the use of security keys using Intune, complete the following steps:

1. Sign in to the [Microsoft Intune admin center](#).
2. Browse to **Devices > Enroll Devices > Windows enrollment > Windows Hello for Business**.
3. Set **Use security keys for sign-in** to **Enabled**.

Configuration of security keys for sign-in isn't dependent on configuring Windows Hello for Business.

ⓘ Note

This will not enable security keys on already provisioned devices. In that case use the next method (Targeted Intune deployment)

Targeted Intune deployment

To target specific device groups to enable the credential provider, use the following custom settings via Intune:

1. Sign in to the [Microsoft Intune admin center](#).
2. Browse to **Devices > Windows > Configuration profiles > Create profile**.
3. Configure the new profile with the following settings:
 - Platform: Windows 10 and later
 - Profile type: Templates > Custom
 - Name: Security Keys for Windows Sign-In
 - Description: Enables FIDO Security Keys to be used during Windows Sign In
4. Select **Next > Add** and in **Add Row**, add the following Custom OMA-URI settings:
 - Name: Turn on FIDO Security Keys for Windows Sign-In
 - Description: (Optional)
 - OMA-URI:
`./Device/Vendor/MSFT/PassportForWork/SecurityKey/UseSecurityKeyForSignin`

- Data Type: Integer
- Value: 1

5. Assign the remainder of the policy settings, including specific users, devices, or groups. For more information, see [Assign user and device profiles in Microsoft Intune](#).

Enable with a provisioning package

For devices not managed by Microsoft Intune, a provisioning package can be installed to enable the functionality. The Windows Configuration Designer app can be installed from the [Microsoft Store](#). Complete the following steps to create a provisioning package:

1. Launch the Windows Configuration Designer.
2. Select **File > New project**.
3. Give your project a name and take note of the path where your project is created, then select **Next**.
4. Leave *Provisioning package* selected as the **Selected project workflow** and select **Next**.
5. Select *All Windows desktop editions* under **Choose which settings to view and configure**, then select **Next**.
6. Select **Finish**.
7. In your newly created project, browse to **Runtime settings > WindowsHelloForBusiness > SecurityKeys > UseSecurityKeyForSignIn**.
8. Set **UseSecurityKeyForSignIn** to *Enabled*.
9. Select **Export > Provisioning package**
10. Leave the defaults in the **Build** window under **Describe the provisioning package**, then select **Next**.
11. Leave the defaults in the **Build** window under **Select security details for the provisioning package** and select **Next**.
12. Take note of or change the path in the **Build** windows under **Select where to save the provisioning package** and select **Next**.
13. Select **Build** on the **Build the provisioning package** page.
14. Save the two files created (*ppkg* and *cat*) to a location where you can apply them to machines later.
15. To apply the provisioning package you created, see [Apply a provisioning package](#).

Note

Devices running Windows 10 Version 1903 must also enable shared PC mode (*EnableSharedPCMode*). For more information about enabling this functionality, see

[Set up a shared or guest PC with Windows 10.](#)

Enable with Group Policy

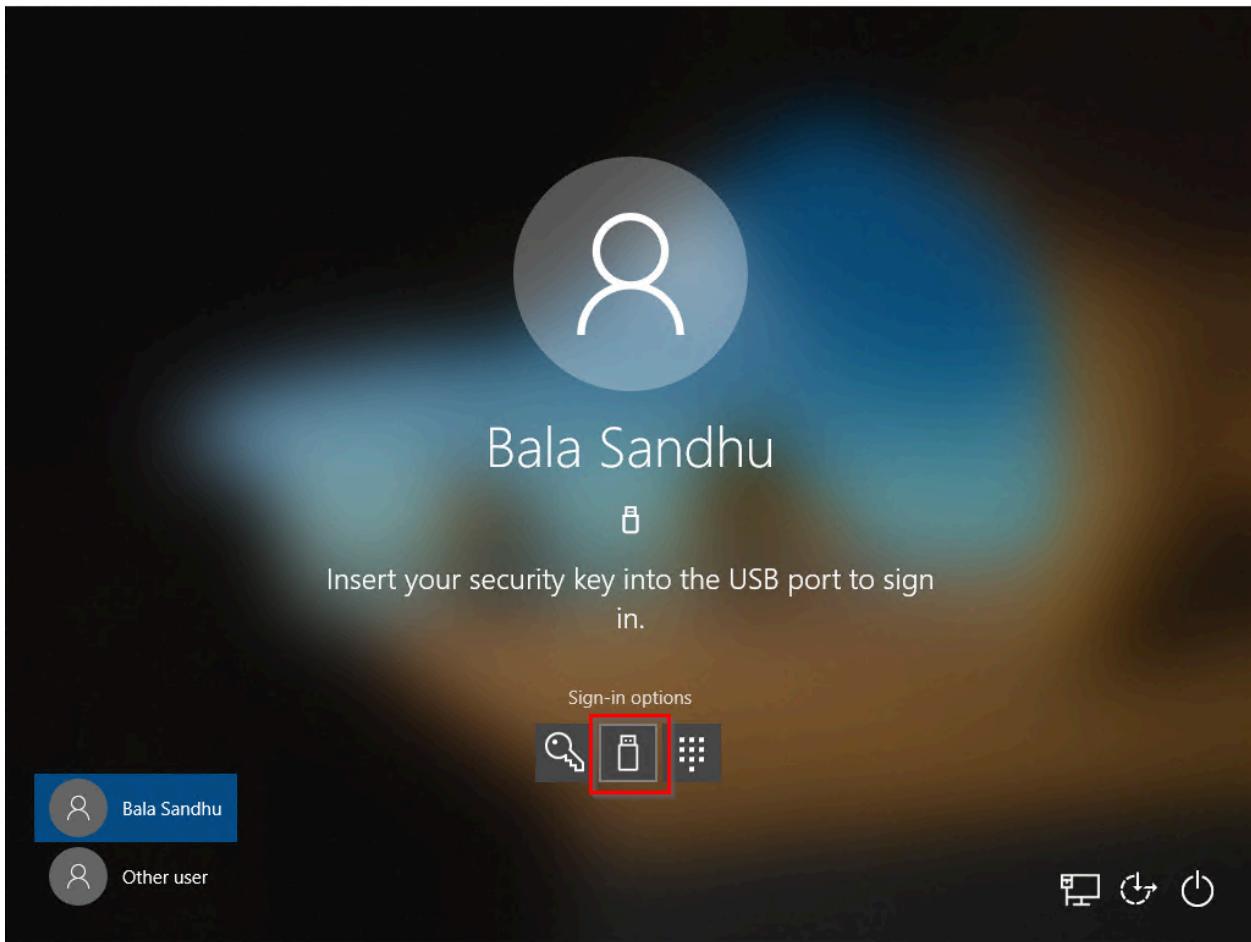
For Microsoft Entra hybrid joined devices, organizations can configure the following Group Policy setting to enable FIDO security key sign-in. The setting can be found under **Computer Configuration > Administrative Templates > System > Logon > Turn on security key sign-in**:

- Setting this policy to **Enabled** allows users to sign in with security keys.
- Setting this policy to **Disabled** or **Not Configured** stops users from signing in with security keys.

This Group Policy setting requires an updated version of the `CredentialProviders.admx` Group Policy template. This new template is available with the next version of Windows Server and with Windows 10 20H1. This setting can be managed with a device running one of these newer versions of Windows or centrally by following the guidance here: [How to create and manage the Central Store for Group Policy Administrative Templates in Windows](#).

Sign in with FIDO2 security key

In this example, a user named Bala Sandhu already provisioned their FIDO2 security key using the steps in the previous article, [Enable passwordless security key sign in](#). For Microsoft Entra hybrid joined devices, make sure you also [enabled passwordless security key sign-in to on-premises resources](#). Bala can choose the security key credential provider from the Windows 10 lock screen and insert the security key to sign into Windows.



Manage security key biometric, PIN, or reset security key

- Windows 10 version 1903 or higher
 - On a device, users can go to **Windows Settings > Accounts > Sign-in options > Security Key**, and then select the **Manage** button.
 - Users can change their PIN, update biometrics, or reset their security key

Troubleshooting and feedback

If you'd like to share feedback or encounter issues about this feature, share via the Windows Feedback Hub app using the following steps:

1. Launch **Feedback Hub** and make sure you're signed in.
2. Submit feedback under the following categorization:
 - Category: Security and Privacy
 - Subcategory: FIDO
3. To capture logs, use the option to **Recreate my Problem**.

Next steps

Enable access to on-premises resources for Microsoft Entra ID and Microsoft Entra hybrid joined devices

[Learn more about device registration](#)

[Learn more about Microsoft Entra multifactor authentication](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Enable passwordless security key sign-in to on-premises resources by using Microsoft Entra ID

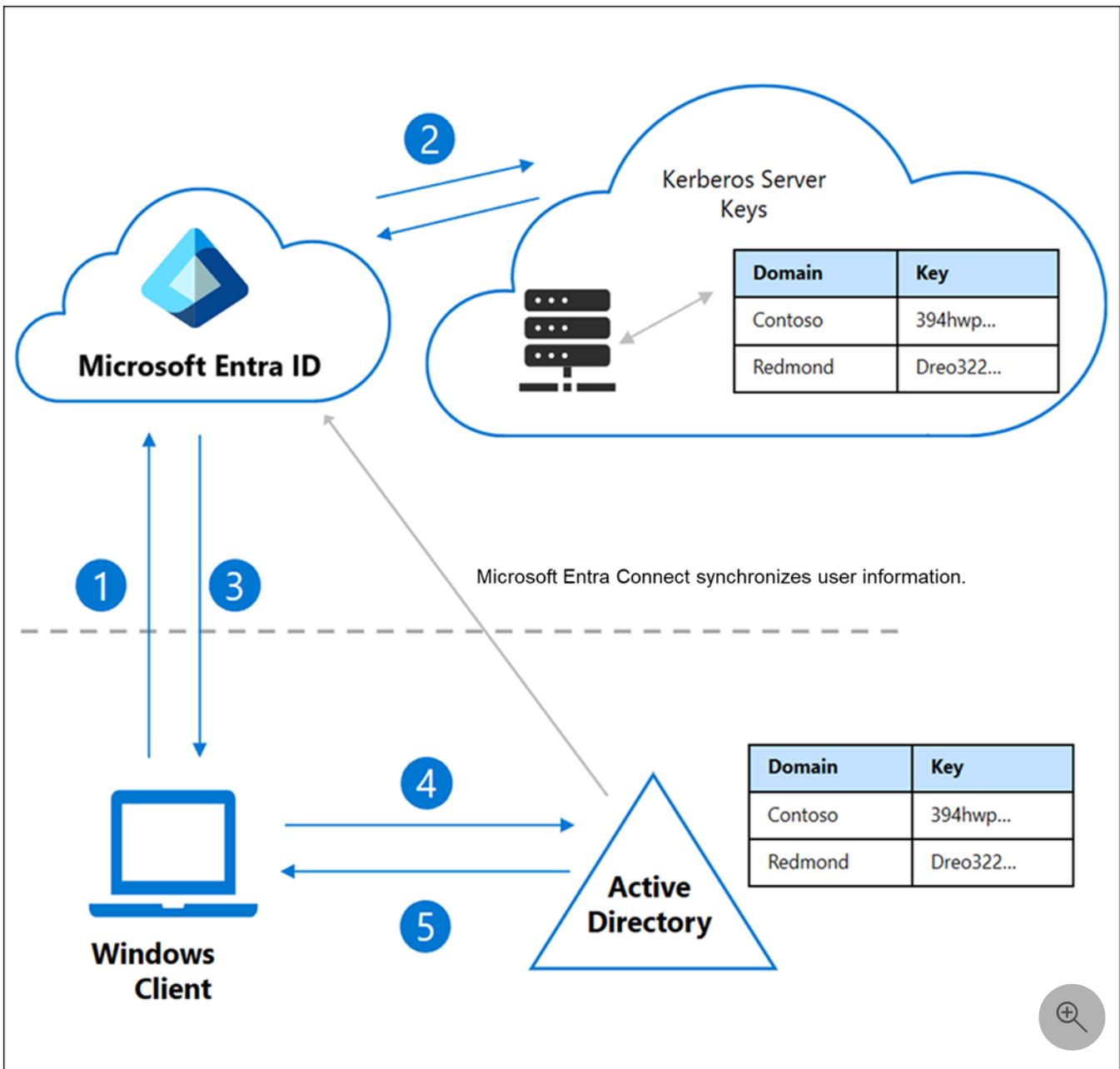
Article • 04/16/2025

This topic shows how to enable passwordless authentication to on-premises resources for environments with devices that run Windows 10 version 2004 or later. Devices can be *Microsoft Entra joined* or *Microsoft Entra hybrid joined*. This passwordless authentication functionality provides seamless single sign-on (SSO) to on-premises resources when you use Microsoft-compatible security keys, or with [Windows Hello for Business Cloud trust](#).

Use SSO to sign in to on-premises resources by using FIDO2 keys

Microsoft Entra ID can issue Kerberos ticket-granting tickets (TGTs) for one or more of your Active Directory domains. With this functionality, users can sign in to Windows with modern credentials, such as FIDO2 security keys, and then access traditional Active Directory-based resources. Kerberos Service Tickets and authorization continue to be controlled by your on-premises Active Directory domain controllers (DCs).

A Microsoft Entra Kerberos server object is created in your on-premises Active Directory instance and then securely published to Microsoft Entra ID by using Microsoft Entra Connect. The object isn't associated with any physical servers. It's simply a resource that can be used by Microsoft Entra ID to generate Kerberos TGTs for your Active Directory domain.



1. A user signs in to a Windows 10 device with an FIDO2 security key and authenticates to Microsoft Entra ID.
 2. Microsoft Entra ID checks the directory for a Kerberos Server key that matches the user's on-premises Active Directory domain.
- Microsoft Entra ID generates a Kerberos TGT for the user's on-premises Active Directory domain. The TGT includes the user's SID only, and no authorization data.
3. The TGT is returned to the client along with the user's Microsoft Entra Primary Refresh Token (PRT).
 4. The client machine contacts an on-premises Active Directory Domain Controller and trades the partial TGT for a fully formed TGT.

5. The client machine now has a Microsoft Entra PRT and a full Active Directory TGT and can access both cloud and on-premises resources.

Prerequisites

Before you begin the procedures in this article, your organization must complete the instructions in [Enable passkeys \(FIDO2\) for your organization](#).

You must also meet the following system requirements:

- Devices must be running Windows 10 version 2004 or later.
- Your Windows Server domain controllers must run Windows Server 2016 or later and have patches installed for the following servers:
 - [Windows Server 2016](#)
 - [Windows Server 2019](#)
- AES256_HMAC_SHA1 must be enabled when **Network security: Configure encryption types allowed for Kerberos** policy is [configured](#) on domain controllers.
- Have the credentials required to complete the steps in the scenario:
 - An Active Directory user who is a member of the Domain Admins group for a domain and a member of the Enterprise Admins group for a forest. Referred to as **\$domainCred**.
 - A Microsoft Entra user with the [Hybrid Identity Administrators](#) role. Referred to as **\$cloudCred**.
- Users must have the following Microsoft Entra attributes populated through Microsoft Entra Connect:
 - `onPremisesSamAccountName` (`accountName` in Microsoft Entra Connect)
 - `onPremisesDomainName` (`domainFQDN` in Microsoft Entra Connect)
 - `onPremisesSecurityIdentifier` (`objectSID` in Microsoft Entra Connect)

Microsoft Entra Connect synchronizes these attributes by default. If you change which attributes to synchronize, make sure you select `accountName`, `domainFQDN`, and `objectSID` for synchronization.

Supported scenarios

The scenario in this article supports SSO in both of the following instances:

- Cloud resources such as Microsoft 365 and other Security Assertion Markup Language (SAML)-enabled applications.

- On-premises resources, and Windows-integrated authentication to websites. The resources can include websites and SharePoint sites that require IIS authentication and/or resources that use NTLM authentication.

Unsupported scenarios

The following scenarios aren't supported:

- Windows Server Active Directory Domain Services (AD DS)-joined (on-premises only devices) deployment.
- Remote Desktop Protocol (RDP), virtual desktop infrastructure (VDI), and Citrix scenarios by using a security key.
- S/MIME by using a security key.
- *Run as* by using a security key.
- Log in to a server by using a security key.

Install the AzureADHybridAuthenticationManagement module

The [AzureADHybridAuthenticationManagement module](#) provides FIDO2 management features for administrators.

1. Open a PowerShell prompt using the Run as administrator option.
2. Install the `AzureADHybridAuthenticationManagement` module:

```
PowerShell

# First, ensure TLS 1.2 for PowerShell gallery access.
[Net.ServicePointManager]::SecurityProtocol =
[Net.ServicePointManager]::SecurityProtocol -bor
[Net.SecurityProtocolType]::Tls12

# Install the AzureADHybridAuthenticationManagement PowerShell module.
Install-Module -Name AzureADHybridAuthenticationManagement -AllowClobber
```

ⓘ Note

- As of update 2.3.331.0, the AzureADHybridAuthenticationManagement module doesn't install the AzureADPreview module.

- You can install the `AzureADHybridAuthenticationManagement` module on any computer from which you can access your on-premises Active Directory Domain Controller, without dependency on the Microsoft Entra Connect solution.
- The `AzureADHybridAuthenticationManagement` module is distributed through the [PowerShell Gallery](#). The PowerShell Gallery is the central repository for PowerShell content. In it, you can find useful PowerShell modules that contain PowerShell commands and Desired State Configuration (DSC) resources.

Create a Kerberos Server object

Administrators use the `AzureADHybridAuthenticationManagement` module to create a Microsoft Entra Kerberos server object in their on-premises directory. The object must be created on the Microsoft Entra Connect server or on a server that has the `Microsoft.Online.PasswordSynchronization.Rpc.dll` dependency installed.

Run the following steps in each domain and forest in your organization that contain Microsoft Entra users:

1. Open a PowerShell prompt using the Run as administrator option.
2. Run the following PowerShell commands to create a new Microsoft Entra Kerberos server object both in your on-premises Active Directory domain and in your Microsoft Entra tenant.

Select Azure Cloud (Default is Azure Commercial)

By default the `Set-AzureADKerberosServer` cmdlet will use the Commercial cloud endpoints. If you are configuring Kerberos in another cloud environment, you need to set the cmdlet to use the specified cloud.

To get a list of the available clouds and the numeric value needed to change, run the following:

```
Get-AzureADKerberosServerEndpoint
```

Example Output:

Console

```
Current Endpoint = 0(Public)
Supported Endpoints:
  0 :Public
  1 :China
  2 :Us Government
```

Note the numeric value next to your desired cloud environment.

To then set the desired cloud environment, run the following:

(Example: For US Government Cloud)

```
Set-AzureADKerberosServerEndpoint -TargetEndpoint 2
```

💡 Tip

For Additional information comparing Azure commercial and sovereign clouds, See:
[Differences between Azure Commercial and Azure sovereign clouds ↗](#).

Example 1 prompt for all credentials

PowerShell

```
# Specify the on-premises Active Directory domain. A new Microsoft Entra ID
# Kerberos Server object will be created in this Active Directory domain.
$domain = $env:USERDNSDOMAIN

# Enter an Azure Active Directory Hybrid Identity Administrator username and
# password.
$cloudCred = Get-Credential -Message 'An Active Directory user who is a member of
the Hybrid Identity Administrators group for Microsoft Entra ID.'

# Enter a Domain Administrator username and password.
$domainCred = Get-Credential -Message 'An Active Directory user who is a member of
the Domain Admins group.'

# Create the new Microsoft Entra ID Kerberos Server object in Active Directory
# and then publish it to Azure Active Directory.
Set-AzureADKerberosServer -Domain $domain -CloudCredential $cloudCred -
DomainCredential $domainCred
```

Example 2 prompt for cloud credential

❗ Note

If you're working on a domain-joined machine with an account that has domain administrator privileges, you can skip the "-DomainCredential" parameter. If the "-DomainCredential" parameter isn't provided, the current Windows login credential is used to access your on-premises Active Directory Domain Controller.

PowerShell

```
# Specify the on-premises Active Directory domain. A new Microsoft Entra ID
# Kerberos Server object will be created in this Active Directory domain.
$domain = $env:USERDNSDOMAIN

# Enter an Azure Active Directory Hybrid Identity Administrator username and
# password.
$cloudCred = Get-Credential

# Create the new Microsoft Entra ID Kerberos Server object in Active Directory
# and then publish it to Azure Active Directory.
# Use the current windows login credential to access the on-premises AD.
Set-AzureADKerberosServer -Domain $domain -CloudCredential $cloudCred
```

Example 3 prompt for all credentials using modern authentication

(!) Note

If your organization protects password-based sign-in and enforces modern authentication methods such as multifactor authentication, FIDO2, or smart card technology, you must use the `-UserPrincipalName` parameter with the User Principal Name (UPN) of a Hybrid Identity Administrator.

- Replace `contoso.corp.com` in the following example with your on-premises Active Directory domain name.
- Replace `administrator@contoso.onmicrosoft.com` in the following example with the UPN of a Hybrid Identity Administrator.

PowerShell

```
# Specify the on-premises Active Directory domain. A new Microsoft Entra ID
# Kerberos Server object will be created in this Active Directory domain.
$domain = $env:USERDNSDOMAIN

# Enter a UPN of a Hybrid Identity Administrator
$userPrincipalName = "administrator@contoso.onmicrosoft.com"

# Enter a Domain Administrator username and password.
$domainCred = Get-Credential

# Create the new Microsoft Entra ID Kerberos Server object in Active Directory
# and then publish it to Azure Active Directory.
# Open an interactive sign-in prompt with given username to access the Microsoft
# Entra ID.
```

```
Set-AzureADKerberosServer -Domain $domain -UserPrincipalName $userPrincipalName -  
DomainCredential $domainCred
```

Example 4 prompt for cloud credentials using modern authentication

! Note

If you are working on a domain-joined machine with an account that has domain administrator privileges and your organization protects password-based sign-in and enforces modern authentication methods such as multifactor authentication, FIDO2, or smart card technology, you must use the `-UserPrincipalName` parameter with the User Principal Name (UPN) of a Hybrid Identity Administrator. And you can skip the "`-DomainCredential`" parameter. > - Replace `administrator@contoso.onmicrosoft.com` in the following example with the UPN of a Hybrid Identity Administrator.

PowerShell

```
# Specify the on-premises Active Directory domain. A new Microsoft Entra ID  
# Kerberos Server object will be created in this Active Directory domain.  
$domain = $env:USERDNSDOMAIN  
  
# Enter a UPN of a Hybrid Identity Administrator  
$userPrincipalName = "administrator@contoso.onmicrosoft.com"  
  
# Create the new Microsoft Entra ID Kerberos Server object in Active Directory  
# and then publish it to Azure Active Directory.  
# Open an interactive sign-in prompt with given username to access the Microsoft  
Entra ID.  
Set-AzureADKerberosServer -Domain $domain -UserPrincipalName $userPrincipalName
```

View and verify the Microsoft Entra Kerberos server

You can view and verify the newly created Microsoft Entra Kerberos server by using the following command:

PowerShell

```
# When prompted to provide domain credentials use the userprincipalname format  
for the username instead of domain\username  
Get-AzureADKerberosServer -Domain $domain -UserPrincipalName $userPrincipalName -  
DomainCredential (get-credential)
```

This command outputs the properties of the Microsoft Entra Kerberos server. You can review the properties to verify that everything is in good order.

! Note

Running against another domain by supplying the credential in domain\username format will connect over NTLM, and then it fails. However, using the userprincipalname format for the domain administrator will ensure RPC bind to the DC is attempted using Kerberos correctly. If the users are in the Protected Users security group in Active Directory, complete these steps to resolve the issue: Sign in as another domain user in **ADConnect** and don't supply "-domainCredential". The Kerberos ticket of the user that's currently signed in is used. You can confirm by executing `whoami /groups` to validate whether the user has the required permissions in Active Directory to execute the preceding command.

[+] Expand table

Property	Description
ID	The unique ID of the AD DS DC object. This ID is sometimes referred to as its <i>slot</i> or its <i>branch ID</i> .
DomainDnsName	The DNS domain name of the Active Directory domain.
ComputerAccount	The computer account object of the Microsoft Entra Kerberos server object (the DC).
UserAccount	The disabled user account object that holds the Microsoft Entra Kerberos server TGT encryption key. The domain name of this account is <code>CN=krbtgt_AzureAD,CN=Users,<Domain-DN></code> .
KeyVersion	The key version of the Microsoft Entra Kerberos server TGT encryption key. The version is assigned when the key is created. The version is then incremented every time the key is rotated. The increments are based on replication metadata and likely greater than one. For example, the initial <i>KeyVersion</i> could be 192272. The first time the key is rotated, the version could advance to 212621. The important thing to verify is that the <i>KeyVersion</i> for the on-premises object and the <i>CloudKeyVersion</i> for the cloud object are the same.
KeyUpdatedOn	The date and time that the Microsoft Entra Kerberos server TGT encryption key was updated or created.
KeyUpdatedFrom	The DC where the Microsoft Entra Kerberos server TGT encryption key was last updated.
CloudId	The ID from the Microsoft Entra object. Must match the ID from the first line of the table.

Property	Description
CloudDomainDnsName	The <i>DomainDnsName</i> from the Microsoft Entra object. Must match the <i>DomainDnsName</i> from the second line of the table.
CloudKeyVersion	The <i>KeyVersion</i> from the Microsoft Entra object. Must match the <i>KeyVersion</i> from the fifth line of the table.
CloudKeyUpdatedOn	The <i>KeyUpdatedOn</i> from the Microsoft Entra object. Must match the <i>KeyUpdatedOn</i> from the sixth line of the table.

Rotate the Microsoft Entra Kerberos server key

The Microsoft Entra Kerberos server encryption *krbtgt* keys should be rotated on a regular basis. We recommend that you follow the same schedule you use to rotate all other Active Directory DC *krbtgt* keys.

⚠️ Warning

There are other tools that could rotate the *krbtgt* keys. However, you must use the tools mentioned in this document to rotate the *krbtgt* keys of your Microsoft Entra Kerberos server. This ensures that the keys are updated in both on-premises Active Directory and Microsoft Entra ID.

PowerShell

```
Set-AzureADKerberosServer -Domain $domain -CloudCredential $cloudCred -  
DomainCredential $domainCred -RotateServerKey
```

Remove the Microsoft Entra Kerberos server

If you want to revert the scenario and remove the Microsoft Entra Kerberos server from both the on-premises Active Directory and Microsoft Entra ID, run the following command:

PowerShell

```
Remove-AzureADKerberosServer -Domain $domain -CloudCredential $cloudCred -  
DomainCredential $domainCred
```

Multiforest and multidomain scenarios

The Microsoft Entra Kerberos server object is represented in Microsoft Entra ID as a *KerberosDomain* object. Each on-premises Active Directory domain is represented as a single *KerberosDomain* object in Microsoft Entra ID.

For example, let's say that your organization has an Active Directory forest with two domains, `contoso.com` and `fabrikam.com`. If you choose to allow Microsoft Entra ID to issue Kerberos TGTs for the entire forest, there are two *KerberosDomain* objects in Microsoft Entra ID, one *KerberosDomain* object for `contoso.com` and the other for `fabrikam.com`. If you have multiple Active Directory forests, there is one *KerberosDomain* object for each domain in each forest.

Follow the instructions in [Create a Kerberos Server object](#) in each domain and forest in your organization that contains Microsoft Entra users.

Known behavior

If your password has expired, signing in with FIDO is blocked. The expectation is that users reset their passwords before they can log in by using FIDO. This behavior also applies to hybrid on-premises synced user sign-in with Windows Hello for Business cloud kerberos trust.

Troubleshooting and feedback

If you encounter issues or want to share feedback about this passwordless security key sign-in feature, share via the Windows Feedback Hub app by doing the following:

1. Open **Feedback Hub**, and make sure that you're signed in.
2. Submit feedback by selecting the following categories:
 - Category: Security and Privacy
 - Subcategory: FIDO
3. To capture logs, use the **Recreate my Problem** option.

Passwordless security key sign-in FAQ

Here are some answers to commonly asked questions about passwordless sign-in:

Does passwordless security key sign-in work in my on-premises environment?

The feature doesn't work in a pure on-premises AD DS environment.

My organization requires two-factor authentication to access resources. What can I do to support this requirement?

Security keys come in a variety of form factors. Contact the device manufacturer of record to discuss how their devices can be enabled with a PIN or biometric as a second factor.

Can administrators set up security keys?

We are working on this capability for the general availability (GA) release of this feature.

Where can I go to find compliant security keys?

For information about compliant security keys, see [FIDO2 security keys](#).

What can I do if I lose my security key?

To delete an enrolled security key, sign in to the [myaccount.microsoft.com](#), and then go to the **Security info** page.

What can I do if I'm unable to use the FIDO security key immediately after I create a Microsoft Entra hybrid joined machine?

If you're clean-installing a Microsoft Entra hybrid joined machine, after the domain join and restart process, you must sign in with a password and wait for the policy to sync before you can use the FIDO security key to sign in.

- Check your current status by running `dsregcmd /status` in a Command Prompt window, and check to ensure that both the **AzureAdJoined** and **DomainJoined** statuses are showing as **YES**.
- This delay in syncing is a known limitation of domain-joined devices and isn't FIDO-specific.

What if I'm unable to get single sign-on to my NTLM network resource after I sign in with FIDO and get a credential prompt?

Make sure that enough DCs are patched to respond in time to service your resource request. To see whether a DC is running the feature, run `nltest /dsgetdc:contoso /keylist /kdc`, and

then review the output.

 **Note**

The `/keylist` switch in the `nltest` command is available in client Windows 10 v2004 and later.

Is there a maximum number of groups per token for Microsoft Entra Kerberos?

Yes, you can have up to 1,010 groups per token.

Do FIDO2 security keys work in a Windows login with RODC present in the hybrid environment?

An FIDO2 Windows login looks for a writable DC to exchange the user TGT. As long as you have at least one writable DC per site, the login works fine.

Next steps

[Learn more about passwordless authentication](#)

Deployment frequently asked questions (FAQs) for hybrid FIDO2 security keys in Microsoft Entra ID

Article • 03/04/2025

This article covers deployment frequently asked questions (FAQs) for Microsoft Entra hybrid joined devices and passwordless sign-in to on-premises resources. With this passwordless feature, you can enable Microsoft Entra authentication on Windows 10 devices for Microsoft Entra hybrid joined devices using FIDO2 security keys. Users can sign into Windows on their devices with modern credentials like FIDO2 keys and access traditional Active Directory Domain Services (AD DS) based resources with a seamless single sign-on (SSO) experience to their on-premises resources.

The following scenarios for users in a hybrid environment are supported:

- Sign in to Microsoft Entra hybrid joined devices using FIDO2 security keys and get SSO access to on-premises resources.
- Sign in to Microsoft Entra joined devices using FIDO2 security keys and get SSO access to on-premises resources.

To get started with FIDO2 security keys and hybrid access to on-premises resources, see the following articles:

- [Passwordless FIDO2 security keys](#)
- [Passwordless Windows 10](#)
- [Passwordless on-premises](#)

Security keys

- [My organization requires two factor authentication to access resources. What can I do to support this requirement?](#)
- [Where can I find compliant FIDO2 security keys?](#)
- [What do I do if I lose my security key?](#)
- [How is the data protected on the FIDO2 security key?](#)
- [How does the registering of FIDO2 security keys work?](#)
- [Is there a way for admins to provision the keys for the users directly?](#)

My organization requires multifactor authentication to access resources. What can I do to support this

requirement?

FIDO2 Security keys come in a variety of form factors. Contact the device manufacturer of interest to discuss how their devices can be enabled with a PIN or biometric as a second factor. For a list of supported providers, see [FIDO2 security keys providers](#).

Where can I find compliant FIDO2 security keys?

For a list of supported providers, see [FIDO2 security keys providers](#).

What if I lose my security key?

You can remove keys by navigating to the [Security info](#) page and removing the FIDO2 security key.

How is the data protected on the FIDO2 security key?

FIDO2 security keys have secure enclaves that protect the private keys stored on them. A FIDO2 security key also has anti-hammering properties built into it, like in Windows Hello, where you can't extract the private key.

How does the registering of FIDO2 security keys work?

For more information how to register and use FIDO2 security keys, see [Enable passwordless security key sign-in](#).

Is there a way for admins to provision the keys for the users directly?

No, not at this time.

Why am I getting "NotAllowedError" in the browser, when registering FIDO2 keys?

You'll receive "NotAllowedError" from fido2 key registration page. This typically happens when an error occurs while Windows attempts a CTAP2 authenticatorMakeCredential operation against the security key. You'll see more details in the Microsoft-Windows-WebAuthN/Operational event log.

Prerequisites

- Does this feature work if there's no internet connectivity?
- What are the specific end points that are required to be open to Microsoft Entra ID?
- How do I identify the domain join type (Microsoft Entra joined or Microsoft Entra hybrid joined) for my Windows 10 device?
- What's the recommendation on the number of DCs that should be patched?
- Can I deploy the FIDO2 credential provider on an on-premises only device?
- FIDO2 security key sign-in isn't working for my Domain Admin or other high privilege accounts. Why?

Does this feature work if there's no internet connectivity?

Internet connectivity is a pre-requisite to enable this feature. The first time a user signs in using FIDO2 security keys, they must have internet connectivity. For subsequent sign-in events, cached sign-in should work and let the user authenticate without internet connectivity.

For a consistent experience, make sure that devices have internet access and line of sight to DCs.

What are the specific end points that are required to be open to Microsoft Entra ID?

The following endpoints are needed for registration and authentication:

- *.microsoftonline.com
- *.microsoftonline-p.com
- *.msauth.net
- *.msauthimages.net
- *.msecnd.net
- *.msftauth.net
- *.msftauthimages.net
- *.phonefactor.net
- enterpriseregistration.windows.net
- management.azure.com
- policykeyservice.dc.ad.msft.net
- secure.aadcdn.microsoftonline-p.com

For a full list of endpoints needed to use Microsoft online products, see [Office 365 URLs and IP address ranges](#).

How do I identify the domain join type (Microsoft Entra joined or Microsoft Entra hybrid joined) for my Windows 10 device?

To check if the Windows 10 client device has the right domain join type, use the following command:

```
Console  
Dsregcmd /status
```

The following sample output shows that the device is Microsoft Entra joined as *AzureADJoined* is set to *YES*:

```
Output  
+-----+  
| Device State |  
+-----+  
  
AzureADJoined: YES  
EnterpriseJoined: NO  
DomainJoined: NO
```

The following sample output shows that the device is Microsoft Entra hybrid joined as *DomainJoined* is also set to *YES*. The *DomainName* is also shown:

```
Output  
+-----+  
| Device State |  
+-----+  
  
AzureADJoined: YES  
EnterpriseJoined: NO  
DomainJoined: YES  
DomainName: CONTOSO
```

On a Windows Server 2016 or 2019 domain controller, check that the following patches are applied. If needed, run Windows Update to install them:

- Windows Server 2016 - [KB4534307](#)

- Windows Server 2019 - [KB4534321](#)

From a client device, run the following command to verify connectivity to an appropriate domain controller with the patches installed:

```
Console
```

```
nltest /dsgetdc:<domain> /keylist /kdc
```

What's the recommendation on the number of DCs that should be patched?

We recommend patching a majority of your Windows Server 2016 or 2019 domain controllers with the patch to ensure they can handle the authentication request load of your organization.

On a Windows Server 2016 or 2019 domain controller, check that the following patches are applied. If needed, run Windows Update to install them:

- Windows Server 2016 - [KB4534307](#)
- Windows Server 2019 - [KB4534321](#)

Can I deploy the FIDO2 credential provider on an on-premises only device?

No, this feature isn't supported for on-premises only device. The FIDO2 credential provider wouldn't show up.

FIDO2 security key sign-in isn't working for my Domain Admin or other high privilege accounts. Why?

The default security policy doesn't grant Microsoft Entra permission to sign high privilege accounts on to on-premises resources.

Due to possible attack vectors from Microsoft Entra ID to Active Directory, it's not recommended to unblock these accounts by relaxing the Password Replication Policy of the computer object CN=AzureADKerberos,OU=Domain Controllers,<domain-DN>.

Under the hood

- How is Microsoft Entra Kerberos linked to my on-premises Active Directory Domain Services environment?
- Where can I view these Kerberos server objects that are created in AD and published in Microsoft Entra ID?
- Why can't we have the public key registered to on-premises AD DS so there's no dependency on the internet?
- How are the keys rotated on the Kerberos server object?
- Why do we need Microsoft Entra Connect? Does it write any info back to AD DS from Microsoft Entra ID?
- What does the HTTP request/response look like when requesting PRT+ partial TGT?

How is Microsoft Entra Kerberos linked to my on-premises Active Directory Domain Services environment?

There are two parts: the on-premises AD DS environment and the Microsoft Entra tenant.

Active Directory Domain Services (AD DS)

The Microsoft Entra Kerberos server is represented in an on-premises AD DS environment as a domain controller (DC) object. This DC object is made up of multiple objects:

- `CN=AzureADKerberos,OU=Domain Controllers,<domain-DN>`

A *Computer* object that represents a Read-Only Domain Controller (RODC) in AD DS. There's no computer associated with this object. Instead, it's a logical representation of a DC.

- `CN=krbtgt_AzureAD,CN=Users,<domain-DN>`

A *User* object that represents a RODC Kerberos Ticket Granting Ticket (TGT) encryption key.

- `CN=900274c4-b7d2-43c8-90ee-00a9f650e335,CN=AzureAD,CN=System,<domain-DN>`

A *ServiceConnectionPoint* object that stores metadata about the Microsoft Entra Kerberos server objects. The administrative tools use this object to identify and locate the Microsoft Entra Kerberos server objects.

Microsoft Entra ID

The Microsoft Entra Kerberos server is represented in Microsoft Entra ID as a *KerberosDomain* object. Each on-premises AD DS environment is represented as a single *KerberosDomain* object in the Microsoft Entra tenant.

For example, you may have an AD DS forest with two domains such as `contoso.com` and `fabrikam.com`. If you allow Microsoft Entra ID to issue Kerberos Ticket Granting Tickets (TGTs) for the entire forest, there are two *KerberosDomain* objects in Microsoft Entra ID - one object for `contoso.com` and one for `fabrikam.com`.

If you have multiple AD DS forests, you have one *KerberosDomain* object for each domain in each forest.

Where can I view these Kerberos server objects that are created in AD DS and published in Microsoft Entra ID?

To view all objects, use the Microsoft Entra Kerberos server PowerShell cmdlets included with the latest version of Microsoft Entra Connect.

For more information, including instructions on how to view the objects, see [create a Kerberos Server object](#).

Why can't we have the public key registered to on-premises AD DS so there's no dependency on the internet?

We received feedback around the complexity of deployment model for Windows Hello for Business, so wanted to simplify the deployment model without having to use certificates and PKI (FIDO2 doesn't use certificates).

How are the keys rotated on the Kerberos server object?

Like any other DC, the Microsoft Entra Kerberos server encryption *krbtgt* keys should be rotated on a regular basis. It's recommended to follow the same schedule as you use to rotate all other AD DS *krbtgt* keys.

Note

Although there are other tools to rotate the *krbtgt* keys, you must [use the PowerShell cmdlets to rotate the *krbtgt* keys](#) of your Microsoft Entra Kerberos

server. This method makes sure that the keys are updated in both the on-premises AD DS environment and in Microsoft Entra ID.

Why do we need Microsoft Entra Connect? Does it write any info back to AD DS from Microsoft Entra ID?

Microsoft Entra Connect doesn't write info back from Microsoft Entra ID to Active Directory DS. The utility includes the PowerShell module to create the Kerberos Server Object in AD DS and publish it in Microsoft Entra ID.

What does the HTTP request/response look like when requesting PRT+ partial TGT?

The HTTP request is a standard Primary Refresh Token (PRT) request. This PRT request includes a claim indicating a Kerberos Ticket Granting Ticket (TGT) is needed.

[\[+\] Expand table](#)

Claim	Value	Description
tgt	true	Claim indicates the client needs a TGT.

Microsoft Entra ID combines the encrypted client key and message buffer into the PRT response as additional properties. The payload is encrypted using the Microsoft Entra Device session key.

[\[+\] Expand table](#)

Field	Type	Description
tgt_client_key	string	Base64 encoded client key (secret). This key is the client secret used to protect the TGT. In this passwordless scenario, the client secret is generated by the server as part of each TGT request and then returned to the client in the response.
tgt_key_type	int	The on-premises AD DS key type used for both the client key and the Kerberos session key included in the KERB_MESSAGE_BUFFER.
tgt_message_buffer	string	Base64 encoded KERB_MESSAGE_BUFFER.

Do users need to be a member of the Domain Users Active Directory group?

Yes. A user must be in the Domain Users group to be able to sign-in using Microsoft Entra Kerberos.

Next steps

To get started with FIDO2 security keys and hybrid access to on-premises resources, see the following articles:

- [Passwordless FIDO2 security keys](#)
 - [Passwordless Windows 10](#)
 - [Passwordless on-premises](#)
-

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Troubleshooting for hybrid deployments of FIDO2 security keys in Microsoft Entra ID

Article • 03/04/2025

This article covers frequently asked questions for Microsoft Entra hybrid joined devices and passwordless sign-in to on-premises resources. With this passwordless feature, you can enable Microsoft Entra authentication on Windows 10 devices for Microsoft Entra hybrid joined devices using FIDO2 security keys. Users can sign into Windows on their devices with modern credentials like FIDO2 keys and access traditional Active Directory Domain Services (AD DS) based resources with a seamless single sign-on (SSO) experience to their on-premises resources.

The following scenarios for users in a hybrid environment are supported:

- Sign in to Microsoft Entra hybrid joined devices using FIDO2 security keys and get SSO access to on-premises resources.
- Sign in to Microsoft Entra joined devices using FIDO2 security keys and get SSO access to on-premises resources.

To get started with FIDO2 security keys and hybrid access to on-premises resources, see the following articles:

- [Passwordless security keys](#)
- [Passwordless Windows 10](#)
- [Passwordless on-premises](#)

Known issues

- [Users are unable to sign in using FIDO2 security keys as Windows Hello Face is too quick and is the default sign-in mechanism](#)
- [Users aren't able to use FIDO2 security keys immediately after they create a Microsoft Entra hybrid joined machine](#)
- [Users are unable to get SSO to my NTLM network resource after signing in with a FIDO2 security key and receiving a credential prompt](#)

Users are unable to sign in using FIDO2 security keys as Windows Hello Face is too quick and is the default sign-in

mechanism

Windows Hello Face is the intended best experience for a device where a user is enrolled. FIDO2 security keys are intended for use on shared devices or where Windows Hello for Business enrollment is a barrier.

If Windows Hello Face prevents the users from trying the FIDO2 security key sign-in scenario, users can turn off Hello Face sign in by removing Face Enrollment in **Settings > Sign-In Options**.

Users aren't able to use FIDO2 security keys immediately after they create a Microsoft Entra hybrid joined machine

After the domain-join and restart process on a clean install of a Microsoft Entra hybrid joined machine, you must sign in with a password and wait for policy to synchronize before you can use to use a FIDO2 security key to sign in.

This behavior is a known limitation for domain-joined devices, and isn't specific to FIDO2 security keys.

To check the current status, use the `dsregcmd /status` command. Check that both *AzureAdJoined* and *DomainJoined* show YES.

Users are unable to get SSO to my NTLM network resource after signing in with a FIDO2 security key and receiving a credential prompt

Make sure that enough DCs are patched to respond in time to service your resource request. To check if you can see a server that is running the feature, review the output of `nltest /dsgetdc:<dc name> /keylist /kdc`

If you're able to see a DC with this feature, the user's password may have changed since they signed in, or there's another issue. Collect logs as detailed in the following section for the Microsoft support team to debug.

Troubleshoot

There are two areas to troubleshoot - [Window client issues](#), or [deployment issues](#).

Windows Client Issues

To collect data that helps troubleshoot issues with signing in to Windows or accessing on-premises resources from Windows 10 devices, complete the following steps:

1. Open the **Feedback hub** app. Make sure that your name is on the bottom left of the app, then select **Create a new feedback item**.
For the feedback item type, choose *Problem*.
2. Select the *Security and Privacy* category, then the *FIDO* subcategory.
3. Toggle the check box for *Send attached files and diagnostics to Microsoft along with my feedback*.
4. Select *Recreate my problems*, then *Start capture*.
5. Lock and unlock the machine with FIDO2 security key. If the issue occurs, try to unlock with other credentials.
6. Return to **Feedback Hub**, select **Stop capture**, and submit your feedback.
7. Go to the *Feedback* page, then the *My Feedback* tab. Select your recently submitted feedback.
8. Select the *Share* button in the top-right corner to get a link to the feedback.
Include this link when you open a support case, or reply to the engineer assigned to an existing support case.

The following events logs and registry key info is collected:

Registry keys

- *HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FIDO* [*]
- *HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\PassportForWork** [*]
- *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Polices\PassportForWork** [*]

Diagnostic information

- Live kernel dump
- Collect AppX package information
- UIF context files

Event logs

- *%SystemRoot%\System32\winevt\Logs\Microsoft-Windows-AAD%40Operational.evtx*
- *%SystemRoot%\System32\winevt\Logs\Microsoft-Windows-WebAuthN%40Operational.evtx*

- %SystemRoot%\System32\winevt\Logs\Microsoft-Windows-HelloForBusiness%40Operational.evtx

Deployment Issues

To troubleshoot issues with deploying the Microsoft Entra Kerberos server, use the logs for the new [AzureADHybridAuthenticationManagement](#) PowerShell module.

Viewing the logs

The Microsoft Entra Kerberos server PowerShell cmdlets in the [AzureADHybridAuthenticationManagement](#) module use the same logging as the standard Microsoft Entra Connect Wizard. To view information or error details from the cmdlets, complete the following steps:

1. On the machine where the [AzureADHybridAuthenticationManagement](#) module was used, browse to C:\ProgramData\AADConnect\. This folder is hidden by default.
2. Open and view the most recent trace-* .log file located in the directory.

Viewing the Microsoft Entra Kerberos server Objects

To view the Microsoft Entra Kerberos server Objects and verify they are in good order, complete the following steps:

1. On the Microsoft Entra Connect Server or any other machine where the [AzureADHybridAuthenticationManagement](#) module is installed, open PowerShell and navigate to C:\Program Files\Microsoft Azure Active Directory Connect\AzureADKerberos\
2. Run the following PowerShell commands to view the Microsoft Entra Kerberos server from both Microsoft Entra ID and on-premises AD DS.

Replace corp.contoso.com with the name of your on-premises AD DS domain.

```
PowerShell

Import-Module ".\AzureAdKerberos.psd1"

# Specify the on-premises AD DS domain.
$domain = "corp.contoso.com"

# Enter an Azure Active Directory Global Administrator username and
# password.
$ccloudCred = Get-Credential
```

```

# Enter a Domain Admin username and password.
$domainCred = Get-Credential

# Get the Azure AD Kerberos Server Object
Get-AzureADKerberosServer -Domain $domain -CloudCredential $cloudCred -DomainCredential
$domainCred

```

The command outputs the properties of the Microsoft Entra Kerberos server from both Microsoft Entra ID and on-premises AD DS. Review the properties to verify that everything is in good order. Use the table below to verify the properties.

The first set of properties is from the objects in the on-premises AD DS environment. The second half (the properties that begin with *Cloud**) are from the Kerberos Server object in Microsoft Entra ID:

[Expand table](#)

Property	Description
Id	The unique <i>Id</i> of the AD DS domain controller object.
DomainDnsName	The DNS domain name of the AD DS domain.
ComputerAccount	The computer account object of the Microsoft Entra Kerberos server object (the DC).
UserAccount	The disabled user account object that holds the Microsoft Entra Kerberos server TGT encryption key. The DN of this account is <i>CN=krbtgt_AzureAD,CN=Users,<Domain-DN></i>
KeyVersion	<p>The key version of the Microsoft Entra Kerberos server TGT encryption key. The version is assigned when the key is created. The version is then incremented every time the key is rotated. The increments are based on replication meta-data and will likely be greater than one.</p> <p>For example, the initial <i>KeyVersion</i> could be 192272. The first time the key is rotated, the version could advance to 212621.</p> <p>The important thing to verify is that the <i>KeyVersion</i> for the on-premises object and the <i>CloudKeyVersion</i> for the cloud object are the same.</p>
KeyUpdatedOn	The date and time that the Microsoft Entra Kerberos server TGT encryption key was updated or created.
KeyUpdatedFrom	The DC where the Microsoft Entra Kerberos server TGT encryption key was last updated.

Property	Description
CloudId	The <i>Id</i> from the Microsoft Entra Object. Must match the <i>Id</i> above.
CloudDomainDnsName	The <i>DomainDnsName</i> from the Microsoft Entra Object. Must match the <i>DomainDnsName</i> above.
CloudKeyVersion	The <i>KeyVersion</i> from the Microsoft Entra Object. Must match the <i>KeyVersion</i> above.
CloudKeyUpdatedOn	The <i>KeyUpdatedOn</i> from the Microsoft Entra Object. Must match the <i>KeyUpdatedOn</i> above.

Next steps

To get started with FIDO2 security keys and hybrid access to on-premises resources, see the following articles:

- [Passwordless FIDO2 security keys](#)
- [Passwordless Windows 10](#)
- [Passwordless on-premises](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Overview of Microsoft Entra certificate-based authentication

Article • 03/04/2025

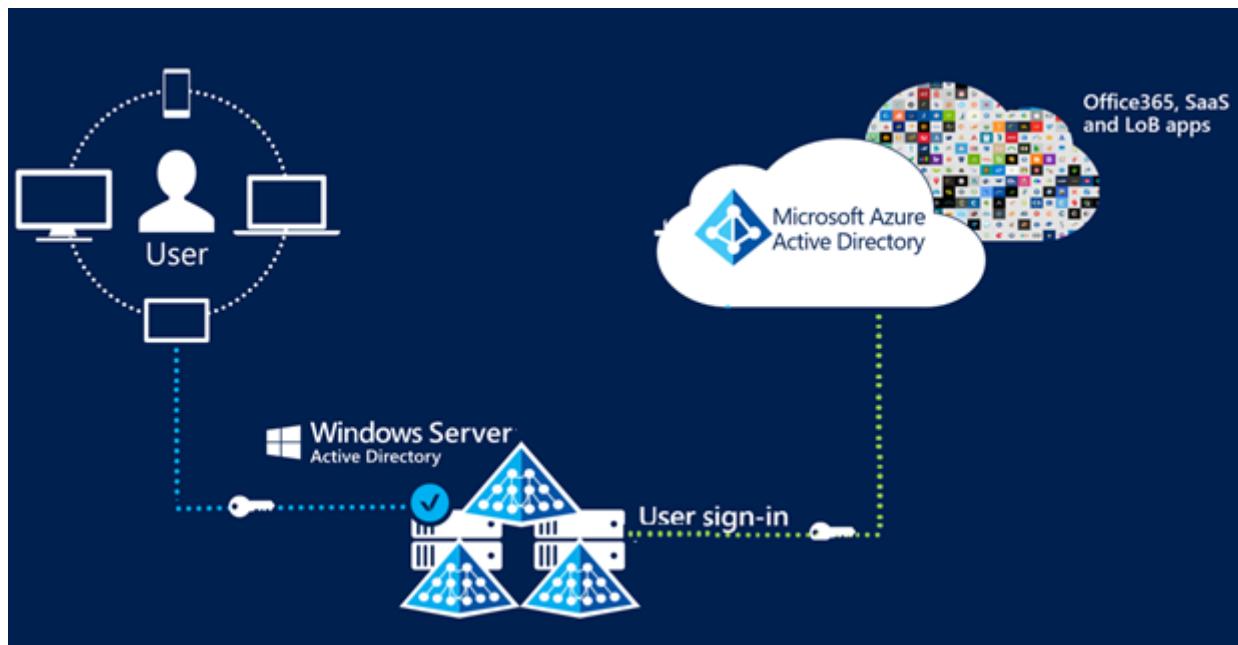
Microsoft Entra certificate-based authentication (CBA) enables customers to allow or require users to authenticate directly with X.509 certificates against their Microsoft Entra ID for applications and browser sign-in. This feature enables customers to adopt a phishing resistant authentication and authenticate with an X.509 certificate against their Public Key Infrastructure (PKI).

What is Microsoft Entra CBA?

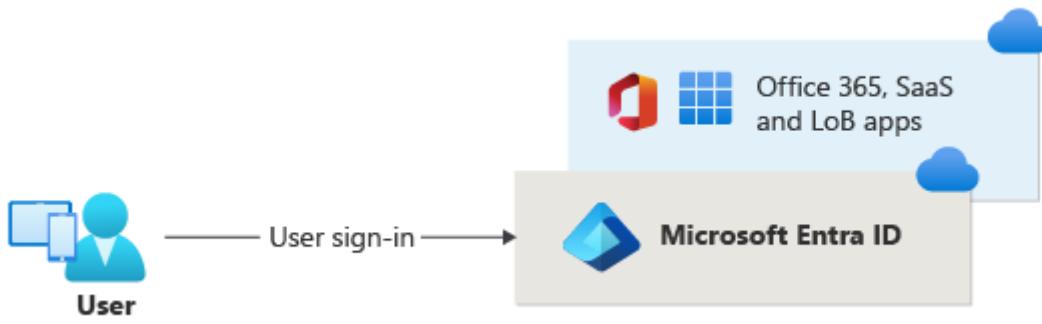
Before cloud-managed support for CBA to Microsoft Entra ID, customers had to implement federated certificate-based authentication, which requires deploying Active Directory Federation Services (AD FS) to be able to authenticate using X.509 certificates against Microsoft Entra ID. With Microsoft Entra certificate-based authentication, customers can authenticate directly against Microsoft Entra ID and eliminate the need for federated AD FS, with simplified customer environments and cost reduction.

The following images show how Microsoft Entra CBA simplifies the customer environment by eliminating federated AD FS.

Certificate-based authentication with federated AD FS



Microsoft Entra certificate-based authentication



Key benefits of using Microsoft Entra CBA

[Expand table](#)

Benefits	Description
Great user experience	<ul style="list-style-type: none"> - Users who need certificate-based authentication can now directly authenticate against Microsoft Entra ID and not have to invest in federated AD FS. - Portal UI enables users to easily configure how to map certificate fields to a user object attribute to look up the user in the tenant (certificate username bindings) - Portal UI to configure authentication policies to help determine which certificates are single-factor versus multifactor.
Easy to deploy and administer	<ul style="list-style-type: none"> - Microsoft Entra CBA is a free feature, and you don't need any paid editions of Microsoft Entra ID to use it. - No need for complex on-premises deployments or network configuration. - Directly authenticate against Microsoft Entra ID.
Secure	<ul style="list-style-type: none"> - On-premises passwords don't need to be stored in the cloud in any form. - Protects your user accounts by working seamlessly with Microsoft Entra Conditional Access policies, including Phishing-Resistant multifactor authentication (MFA requires licensed edition) and blocking legacy authentication. - Strong authentication support where users can define authentication policies through the certificate fields, such as issuer or policy OID (object identifiers), to determine which certificates qualify as single-factor versus multifactor. - The feature works seamlessly with Conditional Access features and authentication strength capability to enforce MFA to help secure your users.

Supported scenarios

The following scenarios are supported:

- User sign-ins to web browser-based applications on all platforms.

- User sign-ins to Office mobile apps on iOS/Android platforms as well as Office native apps in Windows, including Outlook, OneDrive, and so on.
- User sign-ins on mobile native browsers.
- Support for granular authentication rules for multifactor authentication by using the certificate issuer **Subject** and **policy OIDs**.
- Configuring certificate-to-user account bindings by using any of the certificate fields:
 - Subject Alternate Name (SAN) PrincipalName and SAN RFC822Name
 - Subject Key Identifier (SKI) and SHA1PublicKey
 - Issuer + Subject, Subject and Issuer + SerialNumber
- Configuring certificate-to-user account bindings by using any of the user object attributes:
 - User Principal Name
 - onPremisesUserPrincipalName
 - CertificateUserIds

Unsupported scenarios

The following scenarios aren't supported:

- Certificate Authority hints aren't supported, so the list of certificates that appears for users in the certificate picker UI isn't scoped.
- Only one CRL Distribution Point (CDP) for a trusted CA is supported.
- The CDP can be only HTTP URLs. We don't support Online Certificate Status Protocol (OCSP), or Lightweight Directory Access Protocol (LDAP) URLs.
- Password as an authentication method can't be disabled and the option to sign in using a password is displayed even with Microsoft Entra CBA method available to the user.

Known Limitation with Windows Hello For Business certificates

- While Windows Hello For Business (WHFB) can be used for multifactor authentication in Microsoft Entra ID, WHFB isn't supported for fresh MFA. Customers can choose to enroll certificates for your users using the WHFB key pair. When properly configured, these WHFB certificates can be used for multifactor authentication in Microsoft Entra ID. WHFB certificates are compatible with Microsoft Entra certificate-based authentication (CBA) in Microsoft Edge and Chrome browsers; however, at this time WHFB certificates aren't compatible with Microsoft Entra CBA in non-browser scenarios (such as Office 365 applications).

The workaround is to use the "Sign in Windows Hello or security key" option to sign in (when available) as this option doesn't use certificates for authentication and avoids the issue with Microsoft Entra CBA; however, this option may not be available in some older applications.

Out of Scope

The following scenarios are out of scope for Microsoft Entra CBA:

- Public Key Infrastructure for creating client certificates. Customers need to configure their own Public Key Infrastructure (PKI) and provision certificates to their users and devices.

Next steps

- [Technical deep dive for Microsoft Entra CBA](#)
- [How to configure Microsoft Entra CBA](#)
- [Microsoft Entra CBA on iOS devices](#)
- [Microsoft Entra CBA on Android devices](#)
- [Windows smart card sign in using Microsoft Entra CBA](#)
- [Certificate user IDs](#)
- [How to migrate federated users](#)
- [FAQ](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

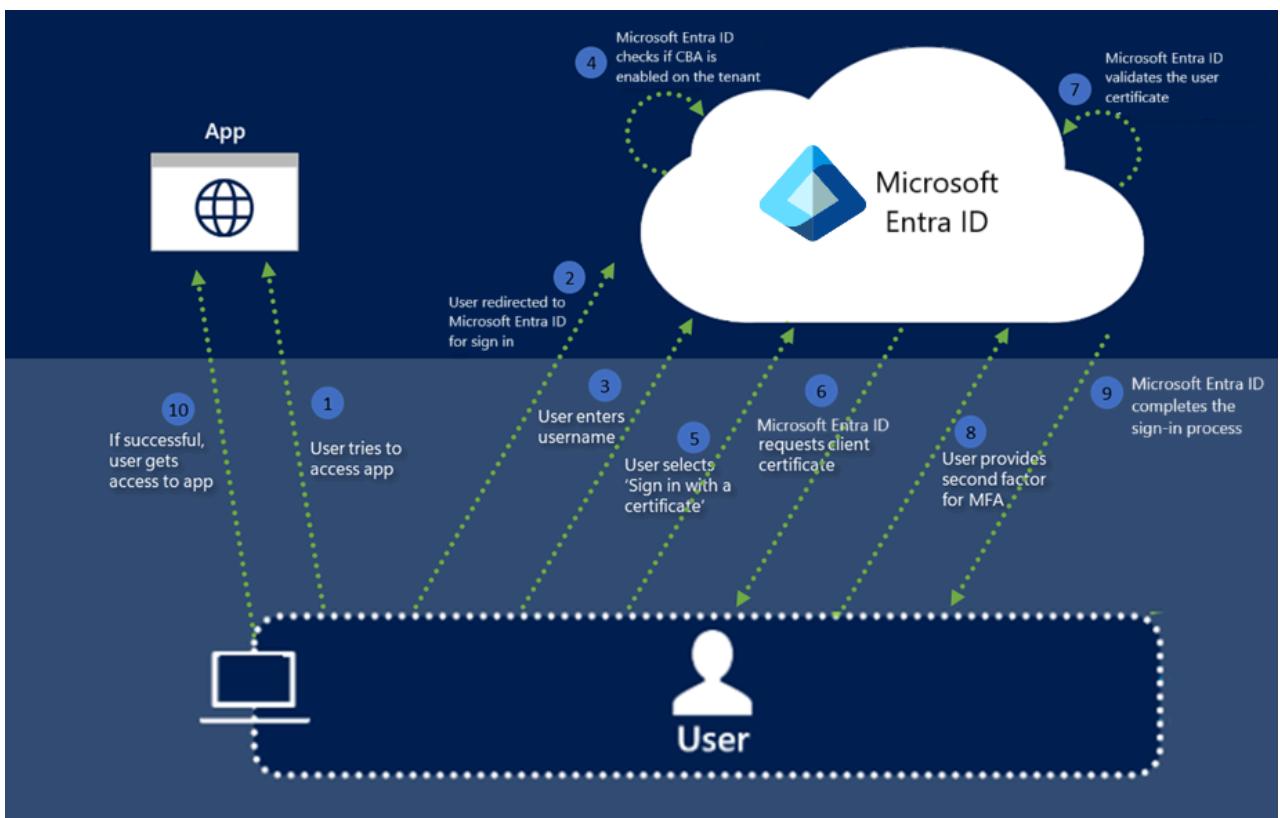
Microsoft Entra certificate-based authentication technical deep dive

Article • 03/04/2025

This article explains how Microsoft Entra certificate-based authentication (CBA) works, and dives into technical details on Microsoft Entra CBA configurations.

How does Microsoft Entra certificate-based authentication work?

The following image describes what happens when a user tries to sign in to an application in a tenant where Microsoft Entra CBA is enabled.



What follows are the steps to take:

1. The user tries to access an application, such as [MyApps portal](#).
2. If the user isn't already signed in, the user is redirected to the Microsoft Entra ID User Sign-in page at <https://login.microsoftonline.com/>.
3. The user enters their username into the Microsoft Entra sign-in page, and then selects **Next**. Microsoft Entra ID does home realm discovery using the tenant name and the username is used to look up the user in the tenant.



Sign in

mfauser@woodgroveorg.net

No account? [Create one!](#)

[Can't access your account?](#)

Back

Next



Sign-in options

4. Microsoft Entra ID checks whether CBA is enabled for the tenant. If CBA is enabled, the user sees a link to **Use a certificate or smartcard** on the password page. If the user doesn't see the sign-in link, make sure CBA is enabled on the tenant. For more information, see [How do I enable Microsoft Entra CBA?](#).

Note

If CBA is enabled on the tenant, all users see the link to **Use a certificate or smart card** on the password page. However, only the users in scope for CBA can authenticate successfully against an application that uses Microsoft Entra ID as their Identity provider (IdP).



← mfauser@woodgroveorg.net

Enter password

Password

[Forgot my password](#)

[Use a certificate or smart card](#)

Sign in

If you enabled other authentication methods like **Phone sign-in** or **Security keys**, users might see a different sign-in screen.



Choose a way to sign in



Approve a request on my Microsoft Authenticator app



Use my password



Use a certificate or smart card

Back

- Once the user selects certificate-based authentication, the client is redirected to the certauth endpoint, which is <https://certauth.login.microsoftonline.com> for public Microsoft Entra ID. For Azure Government, the certauth endpoint is <https://certauth.login.microsoftonline.us>.

The endpoint performs TLS mutual authentication, and requests the client certificate as part of the TLS handshake. An entry for this request appears in the sign-in log.

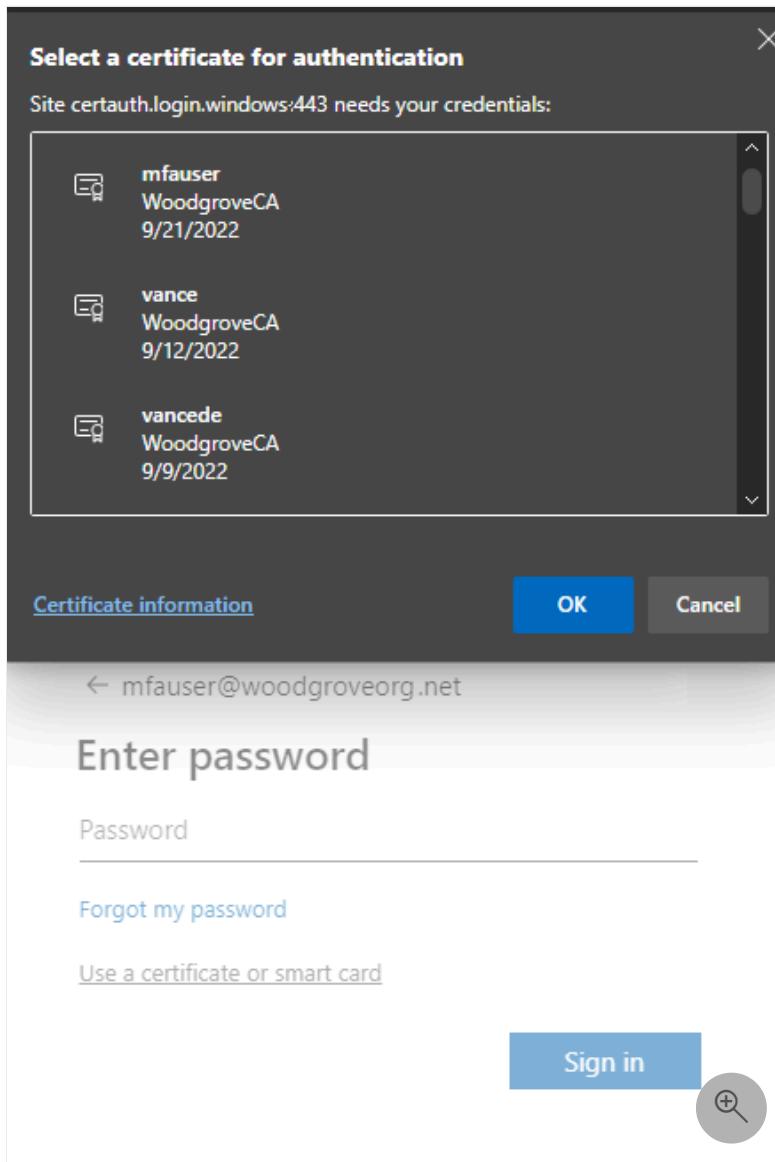
ⓘ Note

The network administrator should allow access to the User sign-in page and certauth endpoint `*.certauth.login.microsoftonline.com` for the customer's cloud environment. Disable TLS inspection on the certauth endpoint to make sure the client certificate request succeeds as part of the TLS handshake.

Make sure your TLS inspection disablement also works for the new url with issuer hints. Don't hardcode the url with tenantId because it might change for B2B users. Use a regular expression to allow both the old and new URL to work for TLS inspection disablement. For example, depending on the proxy, use `*.certauth.login.microsoftonline.com` or `*certauth.login.microsoftonline.com`. In Azure Government, use `*.certauth.login.microsoftonline.us` or `*certauth.login.microsoftonline.us`.

Unless access is allowed, certificate-based authentication fails if you enable [issuer hints](#).

- Microsoft Entra ID requests a client certificate. The user picks the client certificate, and selects Ok.



7. Microsoft Entra ID verifies the certificate revocation list to make sure the certificate isn't revoked and is valid. Microsoft Entra ID identifies the user by using the [username binding configured](#) on the tenant to map the certificate field value to the user attribute value.
8. If a unique user is found with a Conditional Access policy that requires multifactor authentication, and the [certificate authentication binding rule](#) satisfies MFA, then Microsoft Entra ID signs the user in immediately. If MFA is required but the certificate satisfies only a single factor, either passwordless sign-in or FIDO2 is offered as a second factor if they're already registered.
9. Microsoft Entra ID completes the sign-in process by sending a primary refresh token back to indicate successful sign-in.
10. If the user sign-in is successful, the user can access the application.

Understanding issuer hints (Preview)

Issuer hints send back a Trusted CA Indication as part of the TLS handshake. The trusted CA list is set to subject of the Certificate Authorities (CAs) uploaded by the tenant in the Entra trust store. A browser client or native application client can use the hints sent back by server to filter the certificates

shown in the certificate picker. The client shows only the authentication certificates issued by the CAs in the trust store.

Enable issuer hints

To enable select on the check box **Issuer Hints**. [Authentication Policy Administrators](#) should select **I acknowledge** after making sure that the proxy with TLS inspection enabled is updated correctly, and save.

! Note

If your organization has firewalls or proxies with TLS inspection, acknowledge that you disabled TLS inspection of the certauth endpoint capable of matching any name under `[*.]certauth.login.microsoftonline.com`, customized according to the specific proxy in use.

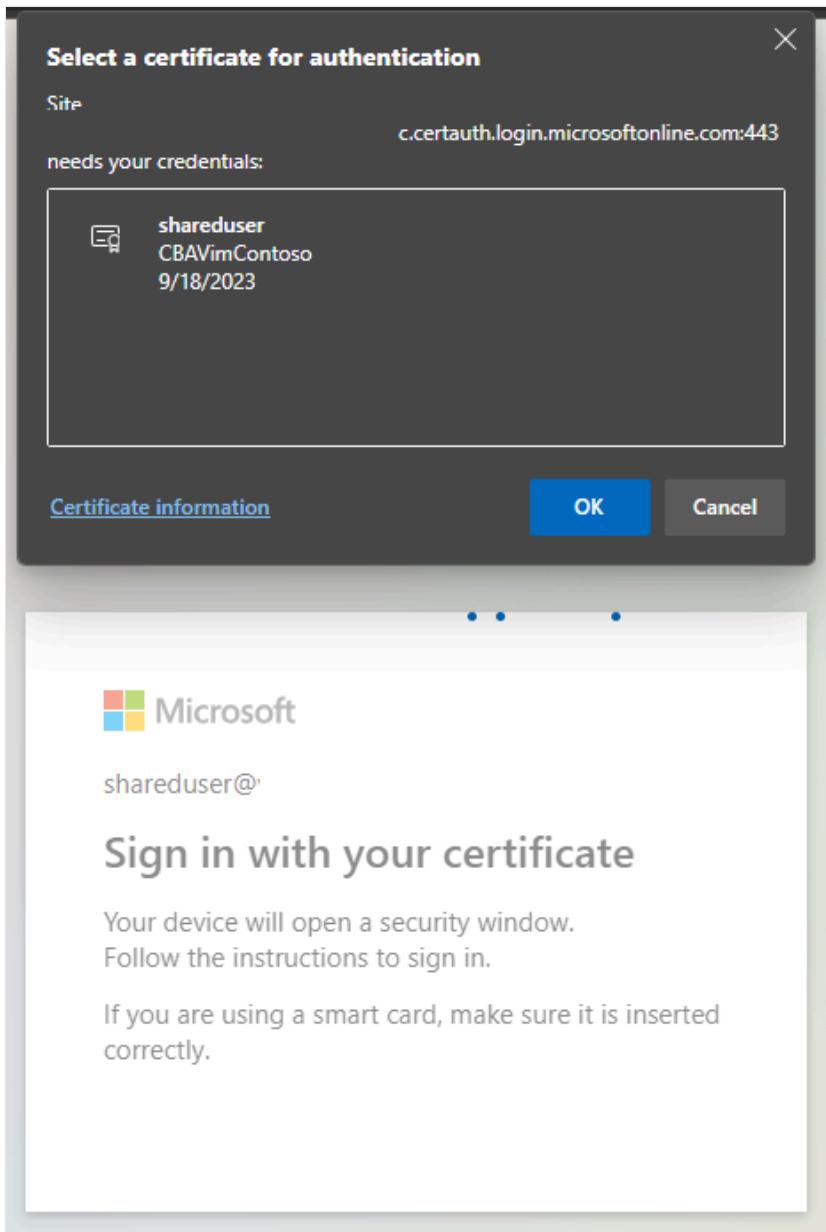
⚠ Please ensure the users enabled for certificate-based authentication (CBA) have a valid certificate. CBA is considered Multi-factor authentication capable, and without valid certificate users will be blocked from using CBA as second factor or registering other methods for MFA. [Learn more](#)

If your organization has firewalls or proxies with TLS Inspection, please acknowledge that you have disabled TLS inspection of the certauth endpoint capable of matching any name under `[*.]certauth.login.microsoftonline.com`, customized according to the specific proxy in use. [Learn more](#)

I Acknowledge

! Note

The certificate authority URL is in the format `t{tenantId}.certauth.login.microsoftonline.com` after issuer hints are enabled.



CA trust store update propagation

After you enable issuer hints and add, update, or delete CAs from the trust state, there's a delay of up to 10 minutes to propagate the issuer hints back to client. Users can't authenticate with certificates issued by the new CAs until the hints are propagated.

Authentication Policy Administrators should sign in with a certificate after they enable issuer hints to initiate the propagation. Users see the following error message when CA trust store updates are in propagation.



Certificate Authority certificates propagation error

The certificate authorities for Issuer Hints are under deployment, try again in few minutes

[More details](#)

[Other ways to sign in](#)

MFA with single-factor certificate-based authentication

Microsoft Entra CBA is supported as both first factor and second factor for authentication. Some of the supported combinations are:

1. CBA (first factor) and [passkeys](#) (second factor)
2. CBA (first factor) and [passwordless phone sign-in](#) (second factor)
3. CBA (first factor) and [FIDO2 security keys](#) (second factor)
4. Password (first factor) + CBA (second factor) (Preview)

ⓘ Note

CBA as a second factor on iOS has [known issues](#) and is blocked on iOS. We are working on fixing the issues and should be supported on iOS.

Users need to have a way to get MFA and register passwordless sign-in or FIDO2 in advance to signing in with Microsoft Entra CBA.

ⓘ Important

A user is considered MFA capable when they're included in the CBA method settings. This means the user can't use proof up as part of their authentication to register other available methods. Make sure users without a valid certificate aren't included in the CBA method settings. For more information about how authentication works, see [Microsoft Entra multifactor authentication](#).

Options to get MFA capability with Single factor certificates

Microsoft Entra CBA is capable of multifactor authentication (MFA). Microsoft Entra CBA can be either single-factor (SF) or multifactor (MF) depending on the tenant configuration. Enabling CBA makes a

user potentially capable to complete MFA. A user with single factor certificate needs another factor to complete MFA which is why we will not allow registration of other methods without satisfying MFA. If the user doesn't have any other MFA auth method registered and are added into scope for CBA auth method, the user can't proof up to register other authentication methods and get MFA.

If the CBA-enabled user only has a Single Factor (SF) certificate and needs to complete MFA:

1. Use a password and SF certificate (OR)
2. Authentication Policy Administrator can issue a Temporary Access Pass (OR)
3. Authentication Policy Administrator adds a phone number and allows voice/text message authentication for the user account.

If the CBA-enabled user hasn't yet been issued a certificate and needs to complete MFA:

1. Authentication Policy Administrator can issue a Temporary Access Pass (OR)
2. Authentication Policy Administrator adds a phone number and allows voice/text message authentication for the user account.

If the CBA-enabled user can't use an MF cert, such as on mobile device without smart card support, and needs to complete MFA:

1. Authentication Policy Administrator can issue a Temporary Access Pass (OR)
2. User needs to register another MFA method (when user can use MF cert on some device) (OR)
3. Authentication Policy Administrator adds a phone number and allows voice/text message authentication for the user account.

Steps to set up passwordless phone sign in (PSI) with CBA

For passwordless sign-in to work, users should disable legacy notification through their mobile app.

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Follow the steps at [Enable passwordless phone sign-in authentication](#).

Important

In the preceding configuration, make sure you chose **Passwordless** option. You need to change the **Authentication mode** for any groups added for PSI to **Passwordless**. If you choose **Any**, CBA and PSI don't work.

3. Select **Protection > Multifactor authentication > Additional cloud-based multifactor authentication settings**.

The screenshot shows the Microsoft Entra admin center. In the top left, there's a breadcrumb trail: Home > Multifactor authentication | Getting started. On the left, a sidebar lists various settings like Account lockout, Block/unblock users, Fraud alert, Notifications, OATH tokens (Preview), Phone call settings, and Providers. The 'Getting started' section is highlighted. The main content area has a heading 'Multifactor authentication' with a sub-section 'Configure' and links to 'Learn more' about cloud-based multifactor authentication. A red box highlights the 'Multifactor authentication' section.

4. Under **Verification options**, clear **Notification through mobile app**, and select **Save**.

The screenshot shows the 'verification options' configuration page. It lists 'Methods available to users:' with four options: 'Call to phone' (unchecked), 'Text message to phone' (checked), 'Notification through mobile app' (unchecked and highlighted with a red box), and 'Verification code from mobile app or hardware token' (unchecked). A link '(learn more)' is visible above the methods.

MFA authentication flow using single factor certificates and passwordless sign in

Let's look at an example of a user who has single-factor certificate, and is configured for passwordless sign-in.

1. Enter your user principal name (UPN) and select **Next**.

The screenshot shows the Microsoft Sign in page. It features the Microsoft logo and the word 'Sign in'. Below it, the email address 'mfauser@woodgroveorg.net' is entered in a text field. Below the email field, there are links for 'Create one!' and 'Can't access your account?'. At the bottom, there are 'Back' and 'Next' buttons. A 'Sign-in options' link is also present at the bottom.

2. Select **Sign in with a certificate**.



← mfauser@woodgroveorg.net

Enter password

Password

[Forgot my password](#)

[Use a certificate or smart card](#)

Sign in

If you enabled other authentication methods like Phone sign-in or FIDO2 security keys, users might see a different sign-in screen.



Choose a way to sign in



Approve a request on my Microsoft Authenticator app



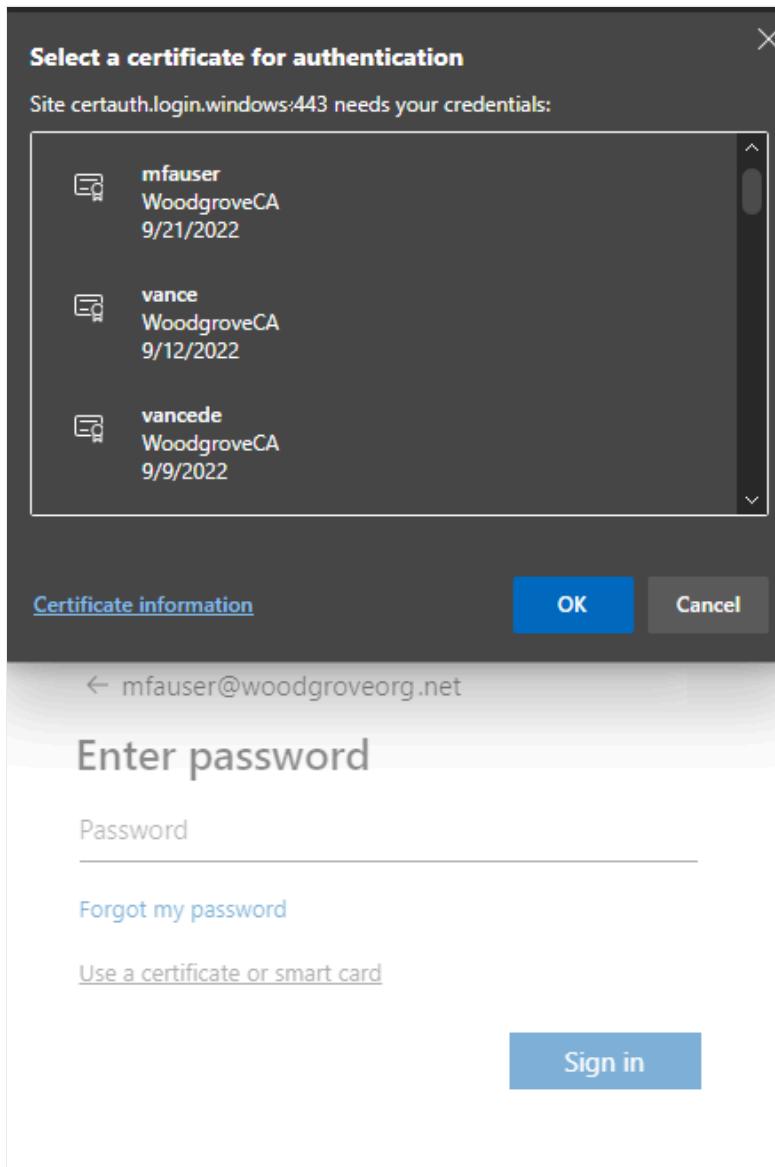
Use my password



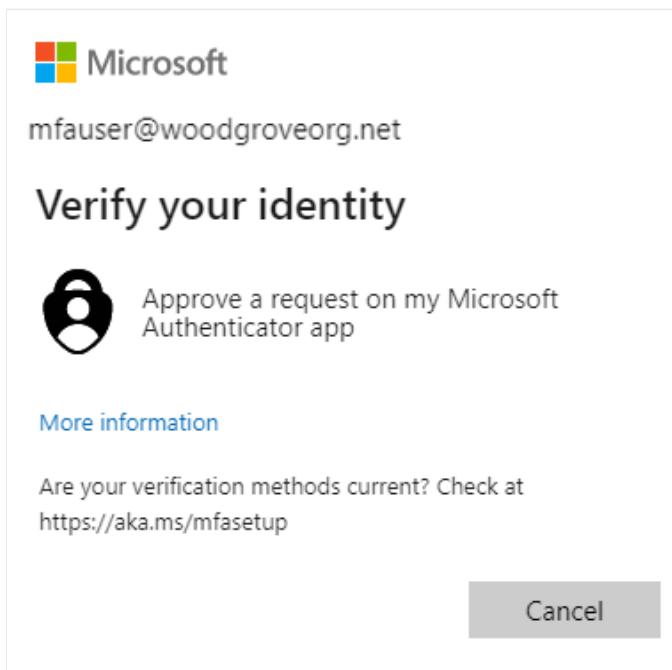
[Use a certificate or smart card](#)

Back

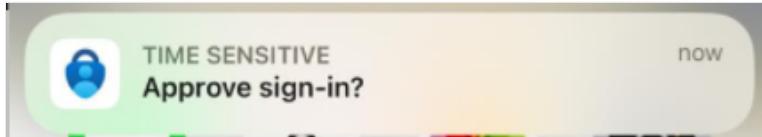
3. Pick the correct user certificate in the client certificate picker and select OK.



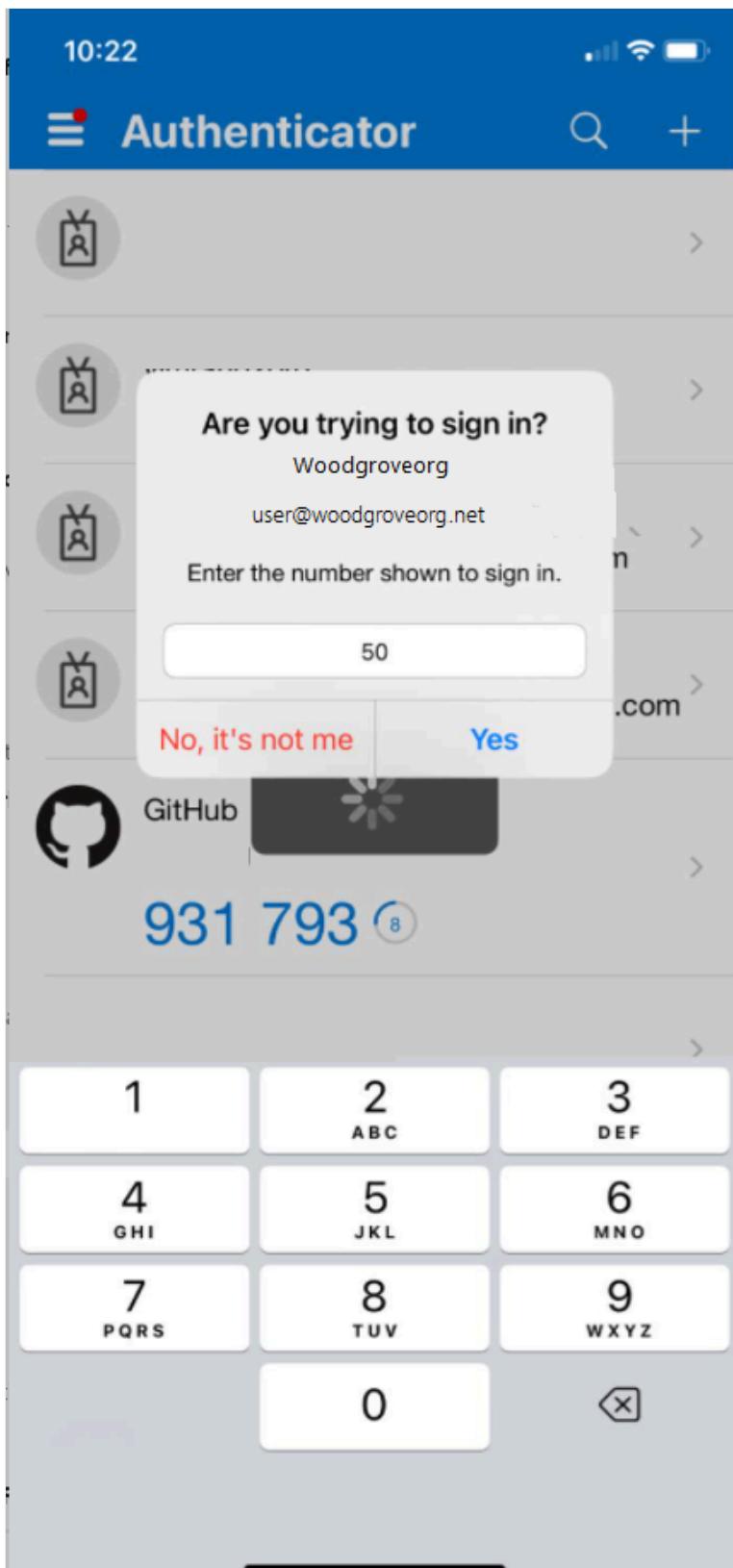
4. Because the certificate is configured to be single-factor authentication strength, the user needs a second factor to meet MFA requirements. The user sees available second factors, which in this case is passwordless sign-in. Select **Approve a request on my Microsoft Authenticator app**.



5. You'll get a notification on your phone. Select **Approve sign-in?**.



6. Enter the number you see on the browser or app screen into Microsoft Authenticator.



7. Select **Yes** and user can authenticate and sign in.

Understanding the authentication binding policy

The authentication binding policy helps determine the strength of authentication as either single-factor or multifactor. Authentication Policy Administrators can change the default value from single-factor to multifactor, or set up custom policy configurations either by using issuer subject or policy OID or Issuer and Policy OID fields in the certificate.

Certificate strengths

Authentication Policy Administrators can determine whether the certificates are single-factor or multifactor strength. For more information, see the documentation that maps [NIST Authentication Assurance Levels to Microsoft Entra auth Methods](#), which builds upon [NIST 800-63B SP 800-63B, Digital Identity Guidelines: Authentication and Lifecycle Mgmt](#).

Multifactor certificate authentication

When a user has a multifactor certificate, they can perform multifactor authentication only with certificates. However, an Authentication Policy Administrator should make sure the certificates are protected with a PIN or biometric to be considered multifactor.

How Microsoft Entra ID resolves multiple authentication policy binding rules

Because multiple custom authentication binding policy rules can be created with different certificate fields like using issuer + policy OID, or just Policy OID or just issuer. Below are the steps used to determine the authentication protection level when custom rules overlap. They are as follows:

1. Issuer + policy OID rules take precedence over Policy OID rules. Policy OID rules take precedence over certificate issuer rules.
2. Issuer + policy OID rules are evaluated first. If you have a custom rule with issuer CA1 and policy OID 1.2.3.4.5 with MFA, only certificate A that satisfies both issuer value and policy OID is given MFA.
3. Next, custom rules using policy OIDs are evaluated. If you have a certificate A with policy OID 1.2.3.4.5 and a derived credential B based on that certificate has a policy OID 1.2.3.4.5.6, and the custom rule is defined as **Policy OID** with value 1.2.3.4.5 with MFA, only certificate A satisfies MFA, and credential B satisfies only single-factor authentication. If the user used derived credential during sign-in and was configured to have MFA, the user is asked for a second factor for successful authentication.
4. If there's a conflict between multiple policy OIDs (such as when a certificate has two policy OIDs, where one binds to single-factor authentication and the other binds to MFA) then treat the certificate as a single-factor authentication.
5. Next, custom rules using issuer CA are evaluated.
6. If a certificate has both policy OID and Issuer rules matching, the policy OID is always checked first, and if no policy rule is found then the issuer bindings are checked. Policy OID has a higher strong authentication binding priority than the issuer.

7. If one CA binds to MFA, all user certificates that the CA issues qualify as MFA. The same logic applies for single-factor authentication.
8. If one policy OID binds to MFA, all user certificates that include this policy OID as one of the OIDs (A user certificate could have multiple policy OIDs) qualify as MFA.
9. One certificate issuer can only have one valid strong authentication binding (that is, a certificate can't bind to both single-factor and MFA).

 **Important**

There's a known issue where a Microsoft Entra Authentication Policy Administrator configures a CBA authentication policy rule using both Issuer and Policy OID impacts some device registration scenarios including:

- Windows Hello For Business enrollment
- Fido2 Security Key registration
- Windows Passwordless Phone Sign-in

Device registration with Workplace Join, Microsoft Entra ID and Hybrid Microsoft Entra device join scenarios aren't impacted. CBA authentication policy rules using either Issuer OR Policy OID aren't impacted. To mitigate, Authentication Policy Administrators should :

- Edit the certificate-based authentication policy rules currently using both Issuer and Policy OID options and remove either the Issuer or OID requirement and save. OR
- Remove the authentication policy rule currently using both Issuer and Policy OID and create rules using only issuer or policy OID

We are working to fix the issue.

Understanding the username binding policy

The username binding policy helps validate the certificate of the user. By default, Subject Alternate Name (SAN) Principal Name in the certificate is mapped to UserPrincipalName attribute of the user object to determine the user.

Achieve higher security with certificate bindings

There are seven supported methods for certificate bindings. In general, mapping types are considered high-affinity if they're based on identifiers that you can't reuse, such as Subject Key Identifiers or SHA1 Public Key. These identifiers convey a higher assurance that only a single certificate can be used to authenticate the respective user.

Mapping types based on user names and email addresses are considered low-affinity. Microsoft Entra ID implements three mappings considered low-affinity based on reusable identifiers. The others are considered high-affinity bindings. For more information, see [certificateUserIds](#).

Certificate mapping field	Examples of values in certificateUserIds	User object attributes	Type
PrincipalName	X509:<PN>bob@woodgrove.com	userPrincipalName onPremisesUserPrincipalName certificateUserIds	low-affinity
RFC822Name	X509:<RFC822>user@woodgrove.com	userPrincipalName onPremisesUserPrincipalName certificateUserIds	low-affinity
IssuerAndSubject (preview)	X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-CA<S>DC=com,DC=contoso,OU=UserAccounts,CN=mfatest	certificateUserIds	low-affinity
Subject (preview)	X509: <S>DC=com,DC=contoso,OU=UserAccounts,CN=mfatest	certificateUserIds	low-affinity
SKI	X509:<SKI>aB1cD2eF3gH4iJ5kL6-mN7oP8qR=	certificateUserIds	high-affinity
SHA1PublicKey	X509:<SHA1-PUKEY>aB1cD2eF3gH4iJ5kL6-mN7oP8qR	certificateUserIds	high-affinity
IssuerAndSerialNumber (preview)	X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-CA<SR>cD2eF3gH4iJ5kL6mN7-oP8qR9sT To get the correct value for serial number, run this command and store the value shown in CertificateUserIds: Syntax: <code>Certutil -dump -v [~certificate path~] >> [~dumpFile path~]</code> Example: <code>certutil -dump -v firstusercert.cer >> firstCertDump.txt</code>	certificateUserIds	high-affinity

Define Affinity binding at the tenant level and override with custom rules (Preview)

With this feature an Authentication Policy Administrator can configure whether a user can be authenticated by using low-affinity or high-affinity username binding mapping. You can set **Required affinity binding** for the tenant, which applies to all users. You can also override the tenant-wide default value by creating custom rules based on Issuer and Policy OID, or Policy OID, or Issuer.

How Microsoft Entra ID resolves multiple username policy binding rules

Use the highest priority (lowest number) binding.

1. Look up the user object by using the username or User Principal Name.

2. Get the list of all username bindings setup by the Authentication Policy Administrator in the CBA authentication method configuration ordered by the 'priority' attribute. Today the concept of priority isn't exposed in Portal UX. Graph returns the priority attribute for each binding and they're used in the evaluation process.
3. If the tenant has high affinity binding enabled or if the certificate value matches a custom rule that required high affinity binding, remove all the low affinity bindings from the list.
4. Evaluate each binding in the list until a successful authentication occurs.
5. If the X.509 certificate field of the configured binding is on the presented certificate, Microsoft Entra ID matches the value in the certificate field to the user object attribute value.
 - a. If a match is found, user authentication is successful.
 - b. If a match isn't found, move to the next priority binding.
6. If the X.509 certificate field isn't on the presented certificate, move to the next priority binding.
7. Validate all the configured username bindings until one of them results in a match and user authentication is successful.
8. If a match isn't found on any of the configured username bindings, user authentication fails.

Securing Microsoft Entra configuration with multiple username bindings

Each of the Microsoft Entra user object attributes (`userPrincipalName`, `onPremiseUserPrincipalName`, `certificateUserIds`) available to bind certificates to Microsoft Entra user accounts has a unique constraint to ensure a certificate only matches a single Microsoft Entra user account. However, Microsoft Entra CBA supports multiple binding methods in the username binding policy which allows an Authentication Policy Administrator to accommodate one certificate to multiple Microsoft Entra user accounts configurations.

Important

If you configure multiple bindings, Microsoft Entra CBA authentication is only as secure as your lowest affinity binding because CBA validates each binding to authenticate the user. To prevent a scenario where a single certificate matches multiple Microsoft Entra accounts, an Authentication Policy Administrator can:

- Configure a single binding method in the username binding policy.
- If a tenant has multiple binding methods configured and doesn't want to allow one certificate to map to multiple accounts, the Authentication Policy Administrator must ensure all allowable methods configured in the policy map to the same Microsoft Entra account. All user accounts should have values matching all of the bindings.
- If a tenant has multiple binding methods configured, the Authentication Policy Administrator should make sure that they don't have more than one low-affinity binding.

For example, lets suppose you have two username bindings on `PrincipalName` mapped to UPN and `SubjectKeyIdentifier` (SKI) to `certificateUserIds`. If you want a certificate to only be used for a single account, an Authentication Policy Administrator must make sure that account has the UPN that is

present in the certificate, and implement the SKI mapping in the certificateUserId attribute of the same account.

Support for multiple certificates with one Microsoft Entra user account (M:1)

There are scenarios where an organization issues multiple certificates for a single identity. Most commonly this could be a derived credential for a mobile device or can also be for a secondary smartcard or x509 credential holder capable device such as a Yubikey.

Cloud only accounts For cloud-only accounts you can map multiple certificates (up to 5) for use by populating the certificateUserIds field (Authorization info in the User Portal) with unique values identifying each certificate. If the organization is using high affinity bindings say Issuer + SerialNumber, values within CertificateUserIds may look like the following:

```
X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-CA<SR>cD2eF3gH4iJ5kL6mN7-oP8qR9sT
```

```
X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-CA<SR>eF3gH4iJ5kL6mN7oP8-qR9sT0uV
```

In this example the first value represents X509Certificate1 and the second value represents X509Certificate2. The user may present either certificate at sign-in and as long as the CBA Username Binding is set to point to the certificateUserIds field to look for the particular binding type (that is, Issuer+SerialNumber in this example), then the user successfully signs in.

Hybrid Synchronized accounts For synchronized accounts you can map multiple certificates for use by populating the altSecurityIdentities field in AD the values identifying each certificate. If the organization is using high affinity (that is, strong authentication) bindings say Issuer + SerialNumber, this could look like the following:

```
X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-CA<SR>cD2eF3gH4iJ5kL6mN7-oP8qR9sT
```

```
X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-CA<SR>eF3gH4iJ5kL6mN7oP8-qR9sT0uV
```

In this example the first value represents X509Certificate1 and the second value represents X509Certificate2. These values must then be synchronized to the certificateUserIds field in Microsoft Entra ID.

Support for one certificate with multiple Microsoft Entra user accounts (1:M)

There are scenarios where an organization needs the user to use the same certificate to authenticate into multiple identities. Most commonly this is for administrative accounts. It can also be for developer accounts or temporary duty accounts. In traditional AD the altSecurityIdentities field is used to populate the certificate values and a Hint is used during sign-in to direct AD to the desired account to check for the sign-in. With Microsoft Entra CBA this is different and there is no Hint. Instead, Home Realm Discovery identifies the desired account to check the certificate values. The other key difference is that Microsoft Entra CBA enforces uniqueness in the certificateUserIds field. This means that two accounts can't both populate the same certificate values.

Important

It isn't a very secure configuration to use same credential to authenticate into different Microsoft Entra accounts and it's recommended not to allow one certificate for multiple Microsoft Entra user accounts.

Cloud only accounts For cloud-only accounts you need to create multiple username bindings and map unique values to each user account that is to use the certificate. Each account is authenticated into using a different username binding. This applies within the boundary of a single directory/tenant (that is, Authentication Policy Administrators can map the certificate for use in another directory/tenant as well, as long as the values remain unique per account too).

Populate the certificateUserIds field (Authorization info in the User Portal) with a unique value identifying the desired certificate. If the organization is using high affinity bindings (that is, strong authentication) bindings say Issuer + SerialNumber and SKI, this could look like the following:

Username bindings:

- Issuer + Serial Number -> CertificateUserIds
- SKI -> CertificateUserIds

User account CertificateUserIds values:

```
X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-CA<SR>aB1cD2eF3gH4iJ5kL6-mN7oP8qR
```

```
X509:<SKI>cD2eF3gH4iJ5kL6mN7-oP8qR9sT
```

Now, when either user presents the same certificate at sign-in the user is successfully sign-in because their account matches a unique value on that certificate. One account is authenticated into using Issuer+SerialNumber and the other using SKI binding.

Note

The number of accounts that can be used in this manner is limited by the number of username bindings configured on the tenant. If the organization is using only High Affinity bindings the number of accounts supported is limited to 3. If the organization is also utilizing low affinity bindings then this number increases to 7 accounts (1 PrincipalName, 1 RFC822Name, 1 SubjectKeyIdentifier, 1 SHA1PublicKey, 1 Issuer+Subject, 1 Issuer+SerialNumber, 1 Subject).

Hybrid Synchronized accounts For synchronized accounts the approach is different. While Authentication Policy Administrators can map unique values to each user account that use the certificate, the common practice of populating all values to each account in Microsoft Entra ID makes this difficult. Instead, Microsoft Entra Connect should filter the desired values per account to unique values populated into the account in Microsoft Entra ID. The uniqueness rule applies within the boundary of a single directory/tenant (that is, Authentication Policy Administrators can map the certificate for use in another directory/tenant as well, as long as the values remain unique per account too). Further, the organization may have multiple AD forests contributing users into a single Microsoft Entra tenant. In this case, Microsoft Entra Connect applies the filter to each of these AD forests with the same goal to populate only a desired unique value to the cloud account.

Populate the altSecurityIdentities field in AD with the values identifying the desired certificate and to include the desired certificate value for that user account type (such as detailee, admin, developer, and so on). Select a key attribute in AD which tells the synchronization which user account type the user is evaluating (such as msDS-cloudExtensionAttribute1). Populate this attribute with the user type value you desire such as detailee, admin, or developer. If this is the user's primary account, the value can be left empty/null.

The accounts could look like the following:

Forest 1 - Account1 (bob@woodgrove.com):

```
X509:<SKI>aB1cD2eF3gH4iJ5kL6mN7oP8qR  
X509:<SHA1-PUKEY>cD2eF3gH4iJ5kL6mN7oP8qR9sT  
X509:<PN>bob@woodgrove.com
```

Forest 1 - Account2 (bob-admin@woodgrove.com):

```
X509:<SKI>aB1cD2eF3gH4iJ5kL6mN7oP8qR  
X509:<SHA1-PUKEY>>cD2eF3gH4iJ5kL6mN7oP8qR9sT  
X509:<PN>bob@woodgrove.com
```

Forest 2 – ADAccount1 (bob-tdy@woodgrove.com):

```
X509:<SKI>aB1cD2eF3gH4iJ5kL6mN7oP8qR  
X509:<SHA1-PUKEY>>cD2eF3gH4iJ5kL6mN7oP8qR9sT  
X509:<PN>bob@woodgrove.com
```

These values must then be synchronized to the certificateUserIds field in Microsoft Entra ID.

Steps to synchronize to certificateUserIds

1. Configure Microsoft Entra Connect to add the alternativeSecurityIds field to the Metaverse
2. For each AD Forest, configure a new custom inbound rule with a high precedence (low number below 100). Add an Expression transform with the altSecurityIdentities field as the source. The target expression uses the Key Attribute that you selected and populated as well as the mapping to the User Types that you defined.
3. For example:

PowerShell

```
IIF((IsPresent([msDS-cloudExtensionAttribute1]) && IsPresent([altSecurityIdentities])),  
    IIF((InStr(LCase([msDS-cloudExtensionAttribute1]),LCASE("detailee"))>0),  
        Where($item,[altSecurityIdentities],(InStr($item, "X509:<SHA1-PUKEY>")>0)),  
        IIF((InStr(LCase([msDS-cloudExtensionAttribute1]),LCASE("developer"))>0),  
            Where($item,[altSecurityIdentities],(InStr($item, "X509:<SKI>")>0)), NULL) ),  
        IIF(IsPresent([altSecurityIdentities]),  
            Where($item,[altSecurityIdentities],(BitAnd(InStr($item, "X509:<I>"),InStrRev($item,  
                "<SR>"))>0)), NULL)  
    )
```

In the example above, altSecurityIdentities and the key attribute msDS-cloudExtensionAttribute1is are first checked to see if they're populated. If not, altSecurityIdentities is checked to see if it's populated. If it's empty then we set it to NULL. Otherwise the account falls into the default case and in this

example we filter to only the Issuer+SerialNumber mapping. If the key attribute is populated, then the value is checked to see if it's equal to one of our defined user types. In this example if that value is detailee, then we filter to the SHA1PublicKey value from altSecurityIdentities. If the value is developer, then we filter to the SubjectKeyIssuer value from altSecurityIdentities. There may be multiple certificate values of a specific type. For example, multiple PrincipalName values or multiple SKI or SHA1-PUKEY values. The filter gets all the values and sync into Microsoft Entra ID – not just the first one it finds.

1. A second example that shows how to push an empty value if the control attribute is empty is:

PowerShell

```
IIF((IsPresent([msDS-cloudExtensionAttribute1]) && IsPresent([altSecurityIdentities])),
    IIF((InStr(LCase([msDS-cloudExtensionAttribute1]),LCase("detailee"))>0),
        Where($item,[altSecurityIdentities],(InStr($item, "X509:<SHA1-PUKEY>")>0)),
        IIF((InStr(LCase([msDS-cloudExtensionAttribute1]),LCase("developer"))>0),
            Where($item,[altSecurityIdentities],(InStr($item, "X509:<SKI>")>0)), NULL) ),
        IIF(IsPresent([altSecurityIdentities]),
            AuthoritativeNull, NULL)
    )
```

If the value in altSecurityIdentities doesn't match any of the search values in the control attribute, then an AuthoritativeNull is passed. This ensures that prior or subsequent rules which populate alternativeSecurityId are ignored and the result is empty in Microsoft Entra ID.

1. Configure a new custom outbound rule with a low precedence (high number above 160 – bottom of list).
2. Add a direct transform with the alternativeSecurityIds field as the source and the certificateUserIds field as the target.
3. Run a synchronization cycle to complete the population of the data in Microsoft Entra ID.

Ensure that CBA in each tenant is configured with the Username Bindings pointing to the certificateUserIds field for the field types that you have mapped from the certificate. Now any of these users may present the certificate at sign-in and after the unique value from the certificate is validated against the certificateUserIds field, that user is successfully signed-in.

Understanding the certificate revocation process

The certificate revocation process allows Authentication Policy Administrators to revoke a previously issued certificate from being used for future authentication. The certificate revocation won't revoke already issued tokens of the user. Follow the steps to manually revoke tokens at [Configure revocation](#).

Microsoft Entra ID downloads and caches the customers certificate revocation list (CRL) from their certificate authority to check if certificates are revoked during the authentication of the user.

Authentication Policy Administrators can configure the CRL distribution point during the setup process of the trusted issuers in the Microsoft Entra tenant. Each trusted issuer should have a CRL that can be referenced by using an internet-facing URL.

ⓘ Important

The maximum size of a CRL for Microsoft Entra ID to successfully download on an interactive sign-in and cache is 20 MB in public Microsoft Entra ID and 45 MB in Azure US Government clouds, and the time required to download the CRL must not exceed 10 seconds. If Microsoft Entra ID can't download a CRL, certificate-based authentications using certificates issued by the corresponding CA fail. As a best practice to keep CRL files within size limits, keep certificate lifetimes within reasonable limits and to clean up expired certificates. For more information, see [Is there a limit for CRL size?](#).

When a user performs an interactive sign-in with a certificate, and the CRL exceeds the interactive limit for a cloud, their initial sign-in fails with the following error:

"The Certificate Revocation List (CRL) downloaded from {uri} has exceeded the maximum allowed size ({size} bytes) for CRLs in Microsoft Entra ID. Try again in few minutes. If the issue persists, contact your tenant administrators."

After the error, Microsoft Entra ID attempts to download the CRL subject to the service-side limits (45 MB in public Microsoft Entra ID and 150 MB in Azure US Government clouds).

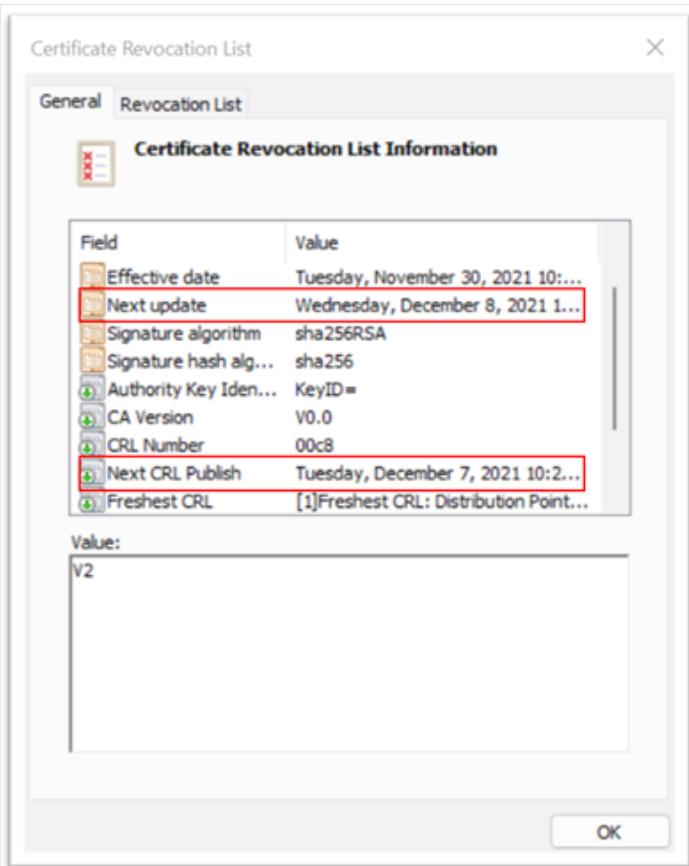
ⓘ Important

If an Authentication Policy Administrator skips the configuration of the CRL, Microsoft Entra ID doesn't perform any CRL checks during the certificate-based authentication of the user. This can be helpful for initial troubleshooting, but shouldn't be considered for production use.

As of now, we don't support Online Certificate Status Protocol (OCSP) because of performance and reliability reasons. Instead of downloading the CRL at every connection by the client browser for OCSP, Microsoft Entra ID downloads once at the first sign-in and caches it. This action improves the performance and reliability of CRL verification. We also index the cache so the search is much faster every time. Customers must publish CRLs for certificate revocation.

The following steps are a typical flow of the CRL check:

1. Microsoft Entra ID attempts to download the CRL at the first sign-in event of any user with a certificate of the corresponding trusted issuer or certificate authority.
2. Microsoft Entra ID caches and reuses the CRL for any subsequent usage. It honors the **Next update date** and, if available, **Next CRL Publish date** (used by Windows Server CAs) in the CRL document.
3. The user certificate-based authentication fails if:
 - A CRL is configured for the trusted issuer and Microsoft Entra ID can't download the CRL, due to availability, size, or latency constraints.
 - The user's certificate is listed as revoked on the CRL.



- Microsoft Entra ID attempts to download a new CRL from the distribution point if the cached CRL document is expired.

(!) Note

Microsoft Entra ID checks the CRL of the issuing CA and other CAs in the PKI trust chain up to the root CA. We have a limit of up to 10 CAs from the leaf client certificate for CRL validation in the PKI chain. The limitation is to make sure a bad actor doesn't bring down the service by uploading a PKI chain with a huge number of CAs with a bigger CRL size. If the tenant's PKI chain has more than 5 CAs, and if there's a CA compromise, Authentication Policy Administrators should remove the compromised trusted issuer from the Microsoft Entra tenant configuration.

(i) Important

Due to the nature of CRL caching and publishing cycles, it's highly recommended that, if there's a certificate revocation, to also revoke all sessions of the affected user in Microsoft Entra ID.

As of now, there's no way to manually force or retrigger the download of the CRL.

How to configure revocation

To revoke a client certificate, Microsoft Entra ID fetches the certificate revocation list (CRL) from the URLs uploaded as part of certificate authority information and caches it. The last publish timestamp (**Effective Date** property) in the CRL is used to ensure the CRL is still valid. The CRL is periodically referenced to revoke access to certificates that are a part of the list.

If a more instant revocation is required (for example, if a user loses a device), the authorization token of the user can be invalidated. To invalidate the authorization token, set the **StsRefreshTokensValidFrom** field for this particular user using Windows PowerShell. You must update the **StsRefreshTokensValidFrom** field for each user you want to revoke access for.

To ensure that the revocation persists, you must set the **Effective Date** of the CRL to a date after the value set by **StsRefreshTokensValidFrom** and ensure the certificate in question is in the CRL.

The following steps outline the process for updating and invalidating the authorization token by setting the **StsRefreshTokensValidFrom** field.

```
https

# Authenticate to Microsoft Graph
Connect-MgGraph -Scopes "User.Read.All"

# Get the user
$user = Get-MgUser -UserPrincipalName "test@yourdomain.com"

# Get the StsRefreshTokensValidFrom property
$user.StsRefreshTokensValidFrom
```

The date you set must be in the future. If the date is not in the future, the **StsRefreshTokensValidFrom** property is not set. If the date is in the future, **StsRefreshTokensValidFrom** is set to the current time (not the date indicated by Set-MsolUser command).

Understanding CRL validation (Preview)

A CRL is a record of digital certificates that have been revoked before the end of their validity period by a certificate authority (CA). When CAs are uploaded to the Microsoft Entra trust store, a CRL, or more specifically the CrlDistributionPoint attribute, isn't required. A CA can be uploaded without a CRL endpoint, and certificate-based authentication won't fail if an issuing CA doesn't have a CRL specified.

To strengthen security and avoid misconfigurations, an Authentication Policy Administrator can require CBA authentication to fail if no CRL is configured for a CA that issues an end user certificate.

Enable CRL validation

To enable CRL validation, select **Require CRL validation (recommended)**.

Certificate revocation list (CRL) validation

This setting requires a CRL check for every certificate authority (CA). If the CRL distribution point is empty or not configured for your CAs, the authentication will fail. You can exempt certificate authorities from the CRL validation requirement.

Require CRL validation (recommended)

Once enabled, any CBA fail is because the end user certificate was issued by a CA with no CRL configured.

An Authentication Policy Administrator can exempt a CA if its CRL has issues that should be fixed. Select **Add Exemption** and select any CAs that should be exempted.

Certificate revocation list (CRL) validation

This setting requires a CRL check for every certificate authority (CA). If the CRL distribution point is empty or not configured for your CAs, the authentication will fail. You can exempt certificate authorities from the CRL validation requirement.

Require CRL validation (recommended)

Exempt CAs from CRL validation 0 CAs selected

+ Add exemption

The CAs in the exempted list aren't required to have CRL configured and the end-user certificates that they issue don't fail authentication.

! Note

There's a known issue with the object picker where the selected items aren't displayed correctly. Use the **Certificate Authorities** tab to select or remove CAs.

Select certificate authorities without CRLs

Try changing or adding filters if you don't see what you're looking for.

Search Search

8 results found

All Certificate authorities

Name	Type	Details
<input checked="" type="checkbox"/> CN=CBAIntermediate1	Certificate aut...	A0:A0:A0:B1:B1:B1:C2:C2:C2:D3:D3:D3:E4:E4:E4
<input checked="" type="checkbox"/> CN=CBAIntermediate2	Certificate aut...	F5:F5:F5:A1:A1:A1:B2:B2:B2:C3:C3:C3:D4:D4:D4
<input type="checkbox"/> CN=CBARoot	Certificate aut...	E5:E5:E5:F6:F6:F6:A7:A7:A7:B8:B8:B8:C9:C9:C9
<input type="checkbox"/> CN=Intermediateroot	Certificate aut...	D0:D0:D0:D0:E1:E1:F1:F2:F2:A3:A3:A3:B4:B4:B4

Selected (2)

Reset

CN=CBAIntermediate1
A0:A0:A0:B1:B1:B1:C2:C2:C2:D3:D3:D3:E4:E4:E4

CN=CBAIntermediate2
F5:F5:F5:A1:A1:A1:B2:B2:B2:C3:C3:C3:D4:D4:D4

Add

How CBA works with a Conditional Access authentication strength policy

Customers can create a Conditional Access authentication strength policy to specify that CBA be used to access a resource.

You can use the built-in **Phishing-resistant MFA** authentication strength. That policy only allows authentication methods that are phishing-resistant like CBA, FIDO2 security keys, and Windows Hello for Business.

You can also create a custom authentication strength to allow only CBA to access sensitive resources. You can allow CBA as a single-factor, multifactor, or both. For more information, see [Conditional Access authentication strength](#).

CBA authentication strength with advanced options

In the CBA Authentication methods policy, an Authentication Policy Administrator can determine the strength of the certificate by using [authentication binding policy](#) on the CBA method. Now you can configure **Advanced options** when you create a custom authentication strength to require a specific certificate to be used, based on issuer and policy OIDs, when users perform CBA to access certain sensitive resources. This feature provides a more precise configuration to determine the certificates and users that can access resources. For more information, see [Advanced options for Conditional Access authentication strength](#).

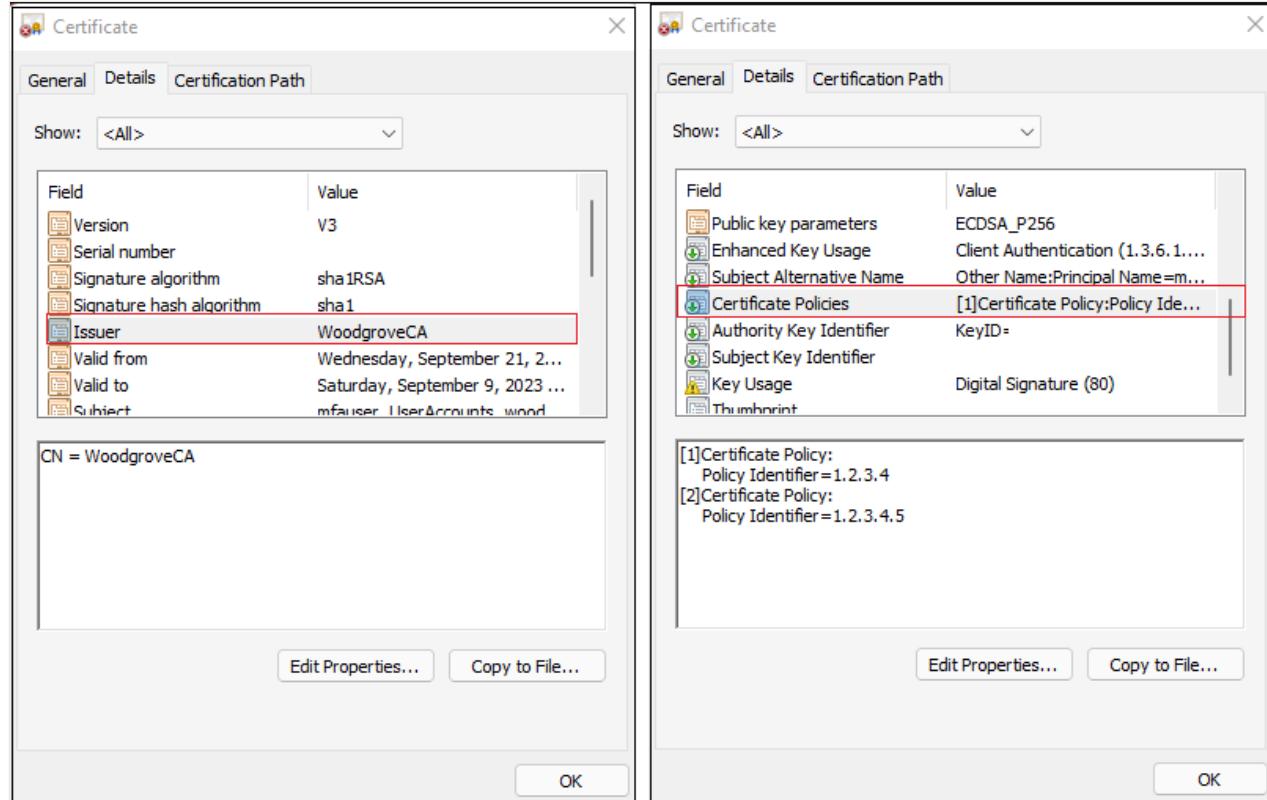
Understanding Sign-in logs

Sign-in logs provide information about sign-ins and how your resources are used by your users. For more information about sign-in logs, see [Sign-in logs in Microsoft Entra ID](#).

Let's walk through two scenarios, one where the certificate satisfies single-factor authentication and another where the certificate satisfies MFA.

For the test scenarios, choose a user with a Conditional Access policy that requires MFA. Configure the user binding policy by mapping SAN Principal Name to UserPrincipalName.

The user certificate should be configured like this screenshot:



Troubleshooting sign-in issues with dynamic variables in sign-in logs

Although sign-in logs provide all the information to debug a user's sign-in issues, there are times when specific values are required and since sign-in logs don't support dynamic variables, the sign-in logs would have missing information. For ex: The failure reason in sign-in log would show something like "The Certificate Revocation List (CRL) failed signature validation. Expected Subject Key Identifier {expectedSKI} doesn't match CRL Authority Key {crlAK}. Request your tenant administrator to check the CRL configuration." where {expectedSKI} and {crlAK} aren't populated with correct values.

When users sign-in with CBA fails, please copy the log details from 'More Details' link in the error page. For more detailed info, look at [understanding CBA error page](#)

Test single-factor authentication

For the first test scenario, configure the authentication policy where the Issuer subject rule satisfies single-factor authentication.

Certificate issuer	Policy OID	Authentication strength	Affinity binding
CN=WOODGROVECA	N/A	Single-factor	Low
N/A	1.2.3.4	Multi-factor	Low

1. Sign in to the [Microsoft Entra admin center](#) as the test user by using CBA. The authentication policy is set where Issuer subject rule satisfies single-factor authentication.
2. Search for and select **Sign-in logs**.

Let's look closer at some of the entries you can find in the **Sign-in logs**.

The first entry requests the X.509 certificate from the user. The status **Interrupted** means that Microsoft Entra ID validated that CBA is enabled in the tenant and a certificate is requested for authentication.

Date	Request ID	User	Application	Status	IP address	Location	Conditional Acc...
2/27/2024, 10:27:57 ...	8b416eb8-9ffcc-47f4...	MOD Administrator	Azure Portal	Success		Lake Worth, Florida, ...	Not Applied
2/27/2024, 10:27:53 ...	a6a2eff4-e3e0-47ca...	MOD Administrator	Azure Portal	Success		Lake Worth, Florida, ...	Not Applied
2/27/2024, 10:27:03 ...	c364dccf-2111-4bbd...	MOD Administrator	Azure Portal	Interrupted		Lake Worth, Florida, ...	Not Applied

The **Activity Details** shows this is just part of the expected sign-in flow where the user selects a certificate.

Basic info	Location	Device info	Authentication Details	Conditional Access	Report-only	...
Date	2/27/2024, 10:27:57 AM					
Request ID	2111-42111-42111-442111-4211142111-42111					
Correlation ID	2111-442111-4211142111-42111-4211142111					
Authentication requirement	Single-factor authentication					
Status	Success					
Continuous access evaluation	No					
Follow these steps:						
Troubleshoot Event	Launch the Sign-in Diagnostic.					
1. Review the diagnosis and act on suggested fixes.						
User						
Username	admin@domain.com					
User ID	2111-442111-4211142111-42111-4211142111					
Sign-in identifier						
User type	Member					
Cross tenant access type	None					
Application	Azure Portal					

The **Additional Details** show the certificate information.

The screenshot shows the Azure portal interface for sign-in events. On the left, there's a navigation menu under 'Identity' with various options like Overview, Users, Groups, Devices, Applications, Roles & admins, Billing, Settings, Protection, Identity governance, External identities, User experiences, Hybrid management, Monitoring & health, Sign-in logs, Audit logs, Provisioning logs, and Health (Preview). The 'Sign-in events' section is selected. On the right, a detailed view of a sign-in event is shown. The event details include:

Basic info	Location	Device info	Authentication Details	Conditional Access	Report-only
Date	2/27/2024, 10:27:57 AM				
Request ID					
Correlation ID					
Authentication requirement	Single-factor authentication				
Status	Success				
Continuous access evaluation	No				

Below this, there's a 'Follow these steps:' section with a 'Troubleshoot Event' link and a note about reviewing diagnosis and suggested fixes. Further down, user details are listed:

User	MOD Administrator
Username	admin@.com
User ID	
Sign-in identifier	
User type	Member
Cross tenant access type	None
Application	Azure Portal

A vertical sidebar on the right lists additional details: Basic info, Location, Device info, Authentication Details, Conditional Access, Report-only, and Authentication Events. A red box highlights the 'Additional Details' section at the bottom of this sidebar.

These additional entries show that the authentication is complete, a primary refresh token is sent back to the browser, and user is given access to the resource.

This screenshot is similar to the one above, showing the Azure portal interface for sign-in events. The left navigation menu is identical. The 'Sign-in events' section is selected. The event details are the same as the previous screenshot, including the table of basic information and user details. A red box highlights the 'Status' field in the 'Basic info' table, which is set to 'Success'. The 'Follow these steps:' section and user details are also present.

Test multifactor authentication

For the next test scenario, configure the authentication policy where the **policyOID** rule satisfies multifactor authentication.

Certificate-based authentication is a passwordless, phishing-resistant authentication method that uses x.509 certificates and an enterprise public key infrastructure (PKI) for authentication. [Learn more](#).

Authentication binding

The authentication binding policy helps determine the strength of your certificate-based authentication method policy as single-factor or multi-factor and low affinity or high affinity. Override default settings with special rules. [Learn more](#)

Certificate issuer	Policy OID	Authentication strength	Affinity binding
CN=WOODGROVECA	N/A	Single-factor	Low
N/A	1.2.3.4	Multi-factor	Low

Username binding

Select one of the X.509 certificates fields to bind with one of the user attributes in the cloud. [Learn more](#)

Certificate field	Affinity binding	User attribute
PrincipalName	Low	userPrincipalName
Issuer and serial number	High	CertificateUserIDs
SKI	High	CertificateUserIDs
RFC822Name	Low	userPrincipalName

Actions: [Save](#) [Discard](#)

1. Sign in to the [Microsoft Entra admin center](#) using CBA. Since the policy was set to satisfy multifactor authentication, the user sign-in is successful without a second factor.
2. Search for and select **Sign-ins**.

You'll see several entries in the Sign-in logs, including an entry with **Interrupted** status.

Sign-in events

Date : Last 24 hours Show dates as : Local Add filters

Date	Request ID	User	Application	Status	IP address	Location	Conditional Acc...
2/27/2024, 10:27:57 ...	8b416eb8-9ffc-47f4...	MOD Administrator	Azure Portal	Success	Lake Worth, Florida, ...	Not Applied	
2/27/2024, 10:27:53 ...	a6a2eff4-e3e0-47ca-...	MOD Administrator	Azure Portal	Success	Lake Worth, Florida, ...	Not Applied	
2/27/2024, 10:27:03 ...	c364dccf-2111-4b9d...	MOD Administrator	Azure Portal	Interrupted	Lake Worth, Florida, ...	Not Applied	

The **Activity Details** shows this is just part of the expected sign-in flow where the user selects a certificate.

Activity Details: Sign-ins

X

Basic info Location Device info Authentication Details Conditional Access Report-only ...

Date 9/21/2022, 10:59:54 AM

Request ID

Correlation ID

Authentication requirement Single-factor authentication

Status Interrupted

Continuous access evaluation No

Additional Details This is an expected part of the login flow, where a user is asked to provide a certificate to resume login flow. No remediation action is required.

Follow these steps:

Troubleshoot Event [Launch the Sign-in Diagnostic.](#)

1. Review the diagnosis and act on suggested fixes.

User MFA User

Username

User ID

Sign-in identifier

User type Member

Cross tenant access type None

Application

Application ID

Resource

Resource ID

Resource tenant ID

Home tenant ID

Home tenant name

Client app

The entry with **Interrupted** status has more diagnostic info on the **Additional Details** tab.

Activity Details: Sign-ins

X

Basic info	Location	Device info	Authentication Details	Conditional Access	Report-only	Additional Details
User certificate subject						
User certificate issuer						
User certificate serial number						
User certificate thumbprint						
User certificate valid from	9/21/2022 5:49:21 PM					
User certificate expiration	9/9/2023 11:30:12 PM					
User certificate binding identifier	m@.net					
User certificate binding	Certificate: PrincipalName; User Attribute: userPrincipalName; Rank: 1					
User certificate authentication level	multiFactorAuthentication					
User certificate authentication level type	Issuer					
User certificate authentication level identifier	CN=WoodgroveCA					
Root Key Type	Unknown					

The following table has a description of each field.

 Expand table

Field	Description
User certificate subject name	Refers to the subject name field in the certificate.
User certificate binding	Certificate: Principal Name; User Attribute: userPrincipalName; Rank: 1 This shows which SAN PrincipalName certificate field was mapped to userPrincipalName user attribute and was priority 1.
User certificate authentication level	multiFactorAuthentication
User certificate authentication level type	PolicyId This shows policy OID was used to determine the authentication strength.
User certificate authentication level identifier	1.2.3.4 This shows the value of the identifier policy OID from the certificate.

Understanding the certificate-based authentication error page

Certificate-based authentication can fail for reasons such as the certificate being invalid, or the user selected the wrong certificate or an expired certificate, or because of a Certificate Revocation List (CRL) issue. When certificate validation fails, the user sees this error:



Certificate validation failed

Try again by doing the following:

1. Close the current browser
2. Open a new browser to sign in
3. Select the certificate

If you are using a smart card, make sure it is inserted correctly.

[More details](#)

[Other ways to sign in](#)

If CBA fails on a browser, even if the failure is because you cancel the certificate picker, you need to close the browser session and open a new session to try CBA again. A new session is required because browsers cache the certificate. When CBA is retried, the browser sends the cached certificate during the TLS challenge, which causes sign-in failure and the validation error.

Select **More details** to get logging information that can be sent to an Authentication Policy Administrator, who in turn can get more information from the Sign-in logs.

Troubleshooting details

X

If you contact your administrator, send this info to them.

[Copy info to clipboard](#)

Request Id: 10b6935f-a443-4bd9-a9b3-09da33e40a00

Correlation Id: 137233e9-62cb-4ddb-b1c4-286532a96f2c

Timestamp: 2022-09-06T19:47:25Z

Message: STS50192: Invalid request.

Flag sign-in errors for review: [Enable flagging](#)

If you plan on getting help for this problem, enable flagging and try to reproduce the error within 20 minutes. Flagged events make diagnostics available and are raised to admin attention.

Select **Other ways to sign in** to try other methods available to the user to sign in.

Note

If you retry CBA in a browser, it'll keep failing due to the browser caching issue. Users need to open a new browser session and sign in again.



Choose a way to sign in



Use my password



Use a certificate or smart card

Certificate-based authentication in MostRecentlyUsed (MRU) methods

Once a user authenticates successfully using CBA, the user's MostRecentlyUsed (MRU) authentication method is set to CBA. Next time, when the user enters their UPN and selects **Next**, the user is taken to the CBA method directly, and need not select **Use the certificate or smart card**.

To reset the MRU method, the user needs to cancel the certificate picker, select **Other ways to sign in**, and select another method available to the user and authenticate successfully.

External identity support

An external identity B2B guest user can use CBA on the home tenant and if the cross tenant settings for the resource tenant is set up to trust MFA from the home tenant, user's CBA auth on home tenant is honored. For more information about how to enable **Trust multifactor authentication from Microsoft Entra tenants**, see [Configure B2B collaboration cross-tenant access](#). CBA on resource tenant isn't supported yet.

Next steps

- [Overview of Microsoft Entra CBA](#)
- [How to configure Microsoft Entra CBA](#)
- [Microsoft Entra CBA on iOS devices](#)
- [Microsoft Entra CBA on Android devices](#)
- [Windows smart card logon using Microsoft Entra CBA](#)
- [Certificate user IDs](#)
- [How to migrate federated users](#)
- [FAQ](#)
- [Troubleshoot Microsoft Entra CBA](#)

Feedback

Was this page helpful?  

Provide product feedback 

How to configure Microsoft Entra certificate-based authentication

Article • 03/04/2025

Microsoft Entra certificate-based authentication (CBA) enables organizations to configure their Microsoft Entra tenants to allow or require users to authenticate with X.509 certificates created by their Enterprise Public Key Infrastructure (PKI) for app and browser sign-in. This feature enables organizations to adopt phishing-resistant modern passwordless authentication by using an x.509 certificate.

During sign-in, users also see an option to authenticate with a certificate instead of entering a password. If multiple matching certificates are present on the device, the user can pick which one to use. The certificate is validated against the user account and if successful, they sign in.

Follow these instructions to configure and use Microsoft Entra CBA for tenants in Office 365 Enterprise and US Government plans. You should already have a [public key infrastructure \(PKI\)](#) configured.

Prerequisites

Make sure that the following prerequisites are in place:

- Configure at least one certificate authority (CA) and any intermediate CAs in Microsoft Entra ID.
- The user must have access to a user certificate (issued from a trusted Public Key Infrastructure configured on the tenant) intended for client authentication to authenticate against Microsoft Entra ID.
- Each CA should have a certificate revocation list (CRL) that can be referenced from internet-facing URLs. If the trusted CA doesn't have a CRL configured, Microsoft Entra ID doesn't perform any CRL checking, revocation of user certificates doesn't work, and authentication isn't blocked.

Important

Make sure the PKI is secure and can't be easily compromised. In the event of a compromise, the attacker can create and sign client certificates and compromise any user in the tenant, both users whom are synchronized from on-premises and cloud-only users. However, a strong key protection strategy, along with other physical and logical controls, such as HSM activation cards or tokens for the secure

storage of artifacts, can provide defense-in-depth to prevent external attackers or insider threats from compromising the integrity of the PKI. For more information, see [Securing PKI](#).

Important

Please visit the [Microsoft recommendations](#) for best practices for Microsoft Cryptographic involving algorithm choice, key length and data protection. Please make sure to use one of the recommended algorithms, key length and NIST approved curves.

Important

As part of ongoing security improvements Azure/M365 endpoints are adding support for TLS1.3 and this process is expected to take a few months to cover the thousands of service endpoints across Azure/M365. This includes the Microsoft Entra endpoint used by Microsoft Entra certificate-based authentication (CBA) `*.certauth.login.microsoftonline.com` and `*.certauth.login.microsoftonline.us`. TLS 1.3 is the latest version of the internet's most deployed security protocol, which encrypts data to provide a secure communication channel between two endpoints. TLS 1.3 eliminates obsolete cryptographic algorithms, enhances security over older versions, and aims to encrypt as much of the handshake as possible. We highly recommend for developers to start testing TLS 1.3 in their applications and services.

Note

When evaluating a PKI, it is important to review certificate issuance policies and enforcement. As mentioned, adding certificate authorities (CAs) to Microsoft Entra configuration allows certificates issued by those CAs to authenticate any user in Microsoft Entra ID. For this reason, it is important to consider how and when the CAs are allowed to issue certificates, and how they implement reusable identifiers. Where administrators need to ensure only a specific certificate is able to be used to authenticate a user, admins should exclusively use high-affinity bindings to achieve a higher level of assurance that only a specific certificate is able to authenticate the user. For more information, see [high-affinity bindings](#).

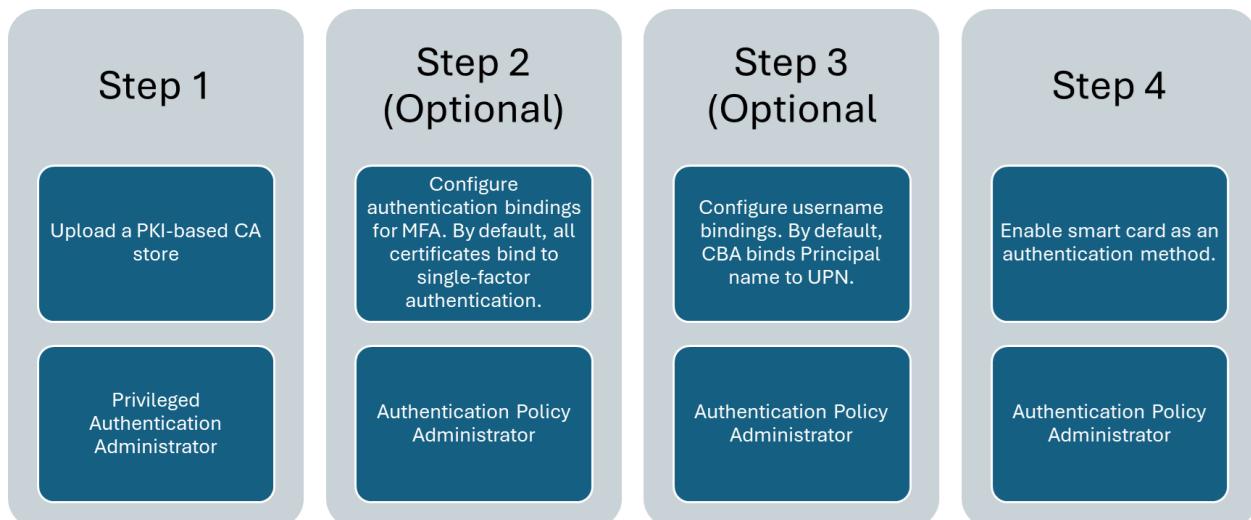
Steps to configure and test Microsoft Entra CBA

Some configuration steps need to be done before you enable Microsoft Entra CBA. First, an admin must configure the trusted CAs that issue user certificates. As seen in the following diagram, we use role-based access control to make sure only least-privileged administrators are needed to make changes.

ⓘ Important

Microsoft recommends that you use roles with the fewest permissions. This practice helps improve security for your organization. Global Administrator is a highly privileged role that should be limited to emergency scenarios or when you can't use an existing role.

Optionally, you can also configure authentication bindings to map certificates to single-factor or multifactor authentication, and configure username bindings to map the certificate field to an attribute of the user object. [Authentication Policy Administrators](#) can configure user-related settings. Once all the configurations are complete, enable Microsoft Entra CBA on the tenant.



Step 1: Configure the certificate authorities with PKI-based trust store (Preview)

Entra has a new public key infrastructure (PKI) based certificate authorities (CA) trust store. The PKI-based CA trust store keeps CAs within a container object for each different PKI. Admins can manage CAs in a container based on PKI easier than one flat list of CAs.

The PKI-based trust store has higher limits for the number of CAs and the size of each CA file. A PKI-based trust store supports up to 250 CAs and 8-KB size for each CA object. We highly recommended you use the new PKI-based trust store for storing CAs, which is scalable and supports new functionality like issuer hints.

 **Note**

If you use [the old trust store to configure CAs](#), we recommended you configure a PKI-based trust store.

An admin must configure the trusted CAs that issue user certificates. Only least-privileged administrators are needed to make changes. A PKI-based trust store has RBAC role [Privilege Authentication Administrator](#).

Upload PKI feature of the PKI-based trust store is available only with Microsoft Entra ID P1 or P2 license. However, with free license as well, admins can upload all the CAs individually instead of the PKI file and configure the PKI-based trust store.

Configure certificate authorities by using the Microsoft Entra admin center

Create a PKI container object

1. Create a PKI container object.
2. Sign in to the Microsoft Entra admin center as an [Privilege Authentication Administrator](#).
3. Browse to **Protection > Show more > Security Center (or Identity Secure Score) > Public key infrastructure (Preview)**.
4. Click **+ Create PKI**.
5. Enter **Display Name**.
6. Click **Create**.

Name	Root cert	Issuer hints enabled	Thumbprint	CRL endpoint	Created	Expires
CN=CBALintermediate1	No	No	<CA Thumbprint goes here>		Sep 11, 2024, 8:42 AM	Jun 18, 2025, 1:31 PM
CN=CBALintermediate2	No	Yes	<CA Thumbprint goes here>		Sep 11, 2024, 8:42 AM	Jun 18, 2025, 1:31 PM
CN=CBARootCA	Yes	No	<CA Thumbprint goes here>		Sep 18, 2024, 10:55 AM	Jun 18, 2025, 1:31 PM

7. Select **Columns** to add or delete columns.

8. Select **Refresh** to refresh the list of PKIs.

Delete a PKI container object

1. To delete a PKI, select the PKI and select **Delete**. If the PKI has CAs in it, enter the name of the PKI to acknowledge the deletion of all CAs within it and select **Delete**.

Name	Root cert	Issuer hints enabled	Thumbprint	CRL endpoint	Created	Expires
CN=CBALintermediate1	No	No	<CA Thumbprint goes here>		Sep 11, 2024, 8:42 AM	Jun 18, 2025, 1:31 PM
CN=CBALintermediate2	No	Yes	<CA Thumbprint goes here>		Sep 11, 2024, 8:42 AM	Jun 18, 2025, 1:31 PM
CN=CBARootCA	Yes	No	<CA Thumbprint goes here>		Sep 18, 2024, 10:55 AM	Jun 18, 2025, 1:31 PM

Upload individual CAs into PKI container object

1. To upload a CA into the PKI container:

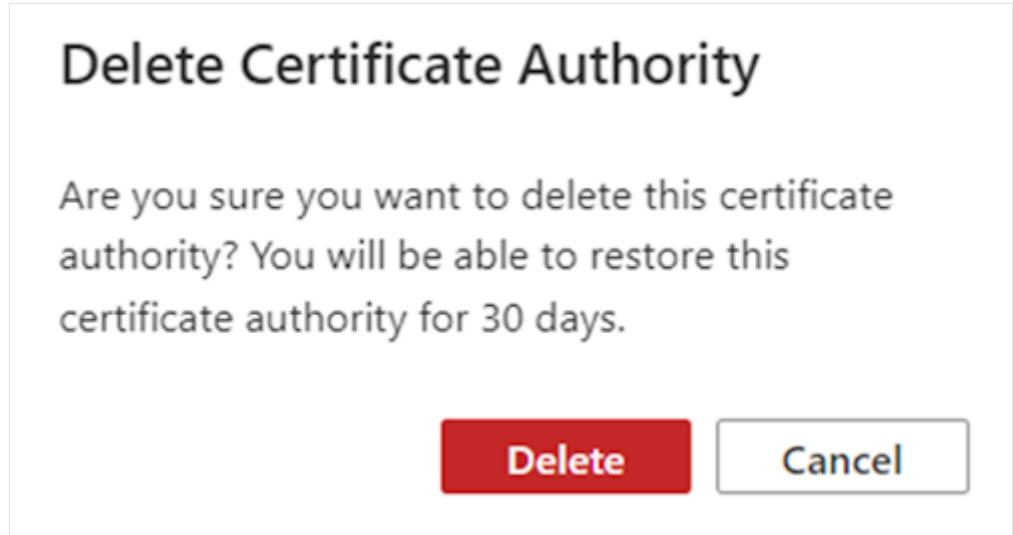
a. Click on **+ Add certificate authority**.

b. Select the CA file.

c. Select **Yes** if the CA is a root certificate, otherwise select **No**.

d. For **Certificate Revocation List URL**, set the internet-facing URL for the CA base CRL that contains all revoked certificates. If the URL isn't set, authentication with revoked certificates doesn't fail.

- e. For **Delta Certificate Revocation List URL**, set the internet-facing URL for the CRL that contains all revoked certificates since the last base CRL was published.
- f. The **Issuer hints** flag is enabled by default. Turn off **Issuer hints** if the CA shouldn't be included in issuer hints.
- g. Select **Save**.
- h. To delete a CA certificate, select the certificate and select **Delete**.



- i. Select **Columns** to add or delete columns.
- j. Select **Refresh** to refresh the list of CAs.

Upload all CAs with upload PKI into PKI container object

1. To upload all CAs at once into the PKI container:
 - a. Create a PKI container object, or open one.
 - b. Select **Upload PKI**.
 - c. Enter the http internet facing URL where the .p7b file is available.
 - d. Enter the SHA256 checksum of the file.
 - e. Select the upload.
 - f. Upload PKI is an asynchronous process. As each CA is uploaded, it's available in the PKI. Completion of PKI upload can take up to 30 minutes.
 - g. Select **Refresh** to refresh the CAs.

To generate the SHA256 checksum of the PKI .p7b file, run this command:

PowerShell

```
Get-FileHash .\CBARootPKI.p7b -Algorithm SHA256
```

Edit a PKI

1. To edit PKI, select ... on the PKI row and select **Edit**.
2. Enter a new PKI name and select **Save**.

Edit a CA

1. To edit CA, select ... on the CA row and select **Edit**.
2. Enter new values for Certificate Authority type (root/intermediate), CRL URL, Delta CRL URL, Issuer Hints enabled flag as necessary and select **Save**.

Restore a PKI

1. Select the **Deleted PKIs** tab.
2. Select the PKI and select **Restore PKI**.

Restore a CA

1. Select the **Deleted CAs** tab.
2. Select the CA file and select **Restore certificate authority**.

Understanding `isIssuerHintEnabled` attribute on CA

Issuer hints send back a Trusted CA Indication as part of the Transport Layer Security (TLS) handshake. The trusted CA list is set to the subject of the CAs uploaded by the tenant in the Entra trust store. For more information about issuer hints, see [Understanding Issuer Hints](#).

By default, the subject names of all CAs in the Microsoft Entra trust store are sent as hints. If you want to send back a hint with only specific CAs, set the issuer hint attribute `isIssuerHintEnabled` to `true`.

There's a character limit of 16 KB for the issuer hints (subject name of the CA) that the server can send back to the TLS client. As a good practice, set the attribute `isIssuerHintEnabled` to true only for the CAs that issue user certificates.

If multiple intermediate CAs from the same root certificate issue the end user certificates, then by default all the certificates show up in the certificate picker. But if you set `isIssuerHintEnabled` to `true` for specific CAs, only the proper user certificates appear in the certificate picker. To enable `isIssuerHintEnabled`, edit the CA, and update the value to `true`.

Configure certificate authorities using the Microsoft Graph APIs

Microsoft Graph APIs can be used to configure CAs. The following examples show how to use Microsoft Graph to run Create, Read, Update, or Delete (CRUD) operations for a PKI or CA.

Create a PKI container object

HTTP

PATCH

```
https://graph.microsoft.com/beta/directory/publicKeyInfrastructure/certifica  
teBasedAuthConfigurations/  
Content-Type: application/json  
{  
    "displayName": "ContosoPKI"  
}
```

Get all the PKI objects

HTTP

GET

```
https://graph.microsoft.com/beta/directory/publicKeyInfrastructure/certifica  
teBasedAuthConfigurations  
ConsistencyLevel: eventual
```

Get PKI object by PKI-id

HTTP

GET

```
https://graph.microsoft.com/beta/directory/publicKeyInfrastructure/certifica  
teBasedAuthConfigurations/{PKI-id}/  
ConsistencyLevel: eventual
```

Upload CAs with a .p7b file

HTTP

PATCH

```
https://graph.microsoft.com/beta/directory/publicKeyInfrastructure/certifica
```

```
teBasedAuthConfigurations/{PKI-id}/certificateAuthorities/{CA-id}
Content-Type: application/json
{
    "uploadUrl": "https://CBA/demo/CBARootPKI.p7b",
    "sha256FileHash":
"AAAAAAD7F909EC2688567DE4B4B0C404443140D128FE14C577C5E0873F68C0FE861E6F"
}
```

Get all CAs in a PKI

HTTP

```
GET
https://graph.microsoft.com/beta/directory/publicKeyInfrastructure/certifica
teBasedAuthConfigurations/{PKI-id}/certificateAuthorities
ConsistencyLevel: eventual
```

Get a specific CA within a PKI by CA-id

HTTP

```
GET
https://graph.microsoft.com/beta/directory/publicKeyInfrastructure/certifica
teBasedAuthConfigurations/{PKI-id}/certificateAuthorities/{CA-id}
ConsistencyLevel: eventual
```

Update specific CA issuer hints flag

HTTP

```
PATCH
https://graph.microsoft.com/beta/directory/publicKeyInfrastructure/certifica
teBasedAuthConfigurations/{PKI-id}/certificateAuthorities/{CA-id}
Content-Type: application/json
{
    "isIssuerHintEnabled": true
}
```

Configure certificate authorities (CA) using PowerShell. For this configuration, you can use [Microsoft Graph PowerShell] (/powershell/microsoftgraph/installation).

1. Start PowerShell with administrator privileges.
2. Install and import the Microsoft Graph PowerShell SDK.

PowerShell

```
Install-Module Microsoft.Graph -Scope AllUsers
Import-Module Microsoft.Graph.Authentication
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

3. Connect to the tenant and accept all.

PowerShell

```
Connect-MGGraph -Scopes "Directory.ReadWrite.All",
"User.ReadWrite.All" -TenantId <tenantId>
```

Audit log

Any CRUD operations on a PKI or CA within the trust store are logged into the Microsoft Entra audit logs.

The screenshot shows the Microsoft Entra Audit Log Details page. On the left, there is a table of audit events with columns: Date, Service, Category, Activity, and Status. Most events show a success status. On the right, there is a detailed view of a single audit event. The event details include:

Audit Log Details		
Activity	Target(s)	Modified Properties
Activity		
Date	9/20/2024, 11:36 AM	
Activity Type	Update PublicKeyInfrastructure	
Correlation ID		
Category	PublicKeyInfrastructure	
Status	SUCCESS	
Status reason		
User Agent		
Initiated by (actor)		
Type	User	Additional Details
Display Name		
Object ID		
IP address		

FAQs

Question: Why does upload PKI fail?

Answer: Check if the PKI file is valid and can be accessed without any issues. The max size of PKI file should be

Question: What is the service level agreement (SLA) for PKI upload?

Answer: PKI upload is an asynchronous operation and may take up to 30 minutes for completion.

Question: How do you generate SHA256 checksum for PKI file?

Answer: To generate the SHA256 checksum of the PKI.p7b file, run this command:

PowerShell

```
Get-FileHash .\CBARootPKI.p7b -Algorithm SHA256
```

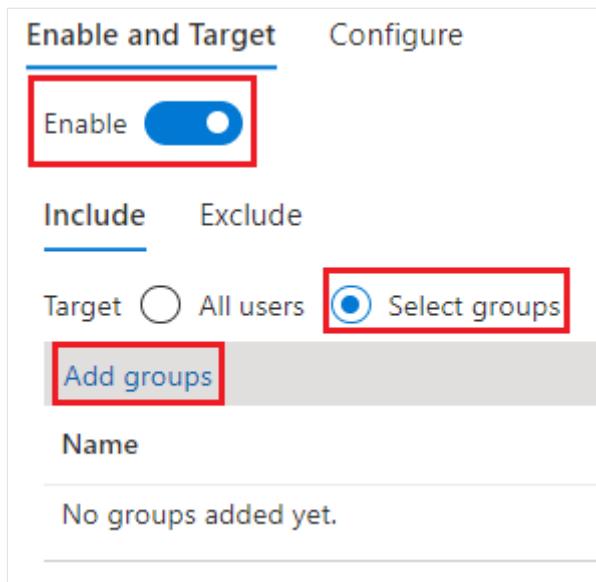
Step 2: Enable CBA on the tenant

i Important

A user is considered capable for MFA when the user is in scope for **Certificate-based authentication** in the Authentication methods policy. This policy requirement means a user can't use proof up as part of their authentication to register other available methods. If the users don't have access to certificates, they get locked out and can't register other methods for MFA. Authentication Policy Administrators need to enable CBA only for users who have valid certificates. Don't include **All users** for CBA. Only use groups of users with valid certificates available. For more information, see [Microsoft Entra multifactor authentication](#).

To enable CBA in the Microsoft Entra admin center, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Policy Administrator**.
2. Browse to **Groups > All groups** > select **New group** and create a group for CBA users.
3. Browse to **Protection > Authentication methods > Certificate-based Authentication**.
4. Under **Enable and Target**, select **Enable**, and click **I Acknowledge**.
5. Click **Select groups**, click **Add groups**.
6. Choose specific groups like the one you created, and click **Select**. Use specific groups rather than **All users**.
7. When you are done, click **Save**.



Once certificate-based authentication is enabled on the tenant, all users in the tenant see the option to sign in with a certificate. Only users who are enabled for CBA can authenticate by using the X.509 certificate.

① Note

The network administrator should allow access to certauth endpoint for the customer's cloud environment in addition to `login.microsoftonline.com`. Disable TLS inspection on the certauth endpoint to make sure the client certificate request succeeds as part of the TLS handshake.

Step 3: Configure authentication binding policy

The authentication binding policy helps determine the strength of authentication to either a single factor or multifactor. The default protection level for all the certificates on the tenant is **Single-factor authentication**. The default affinity binding at the tenant level is **Low**. An Authentication Policy Administrator can change the default value from single-factor to multifactor and if changes, all the certificates on the tenant will be considered of strength **Multi-factor authentication**. Similarly, the affinity binding at the tenant level can be set to **High** which means all the certificates will be validated using only high affinity attributes.

① Important

Admin should set the tenant default to a value that is applicable for most certificates and create custom rules only for specific certificates that needs different protection level and/or affinity binding than tenant default. All the authentication

methods configuration go into the same policy file so creating multiple redundant rules can hit the policy file limit.

Authentication binding rules map certificate attributes, such as Issuer, or Policy Object ID (OID), or Issuer and Policy OID, to a value and select default protection level as well as affinity binding for that rule. To modify tenant default settings and create custom rules in the Microsoft Entra admin center, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Policies**.
3. Under **Manage**, select **Authentication methods > Certificate-based Authentication**.

The screenshot shows the Microsoft Entra admin center interface. The left sidebar has 'Home' and 'Authentication methods | Policies' selected. Under 'Manage', 'Policies' is selected. The main content area is titled 'Manage migration' with a note about deprecating legacy policies by September 30th, 2025. It shows a table of authentication methods with their targets and enable status. A red box highlights the 'Certificate-based authentication' row, which is currently disabled.

Method	Target	Enabled
FIDO2 security key	2 users, 1 group	Yes
Microsoft Authenticator	All users	Yes
SMS	All users	Yes
Temporary Access Pass	All users	Yes
Hardware OATH tokens (Preview)		No
Third-party software OATH tokens		No
Voice call		No
Email OTP		Yes
Certificate-based authentication		No

4. Select **Configure** to set up authentication binding and username binding.
5. The protection level attribute has a default value of **Single-factor authentication**. Select **Multifactor authentication** to change the default value to MFA.

Note

The default protection level value is in effect if no custom rules are added. If custom rules are added, the protection level defined at the rule level is honored instead.

Enable and Target Configure

Authentication binding

The authentication binding policy helps determine the strength of your certificate-based authentication method policy as single-factor or multi-factor and low affinity or high affinity. Override default settings with special rules. [Learn more](#)

Protection Level ⓘ

Single-factor authentication

Multi-factor authentication

6. You can also set up custom authentication binding rules to help determine the protection level for client certificates that need different values for protection level or affinity binding than tenant default. The rules can be configured using either the issuer Subject or Policy OID or both fields in the certificate.

Authentication binding rules map the certificate attributes (issuer or Policy OID) to a value, and select default protection level for that rule. Multiple rules can be created. For the config below let us assume the tenant default is **Multifactor authentication** and **Low affinity binding**.

To add custom rules, select **Add rule**.

Enable and Target Configure

Authentication binding

The authentication binding policy helps determine the strength of your certificate-based authentication method policy as single-factor or multi-factor and low affinity or high affinity. Override default settings with special rules. [Learn more](#)

Protection Level ⓘ

Single-factor authentication

Multi-factor authentication

Required Affinity Binding ⓘ

Low

High

+ Add rule

Certificate issuer	Policy OID	Authentication strength	Affinity binding
You do not have any authentication binding policy rules.			

To create a rule by certificate issuer, select **Certificate issuer**.

- Select a **Certificate issuer identifier** from the list box.
- Select **Multifactor authentication** but **High affinity binding**, and then click **Add**. When prompted, click **I acknowledge** to finish adding the rule.

Add authentication binding policy rule

Certificate attribute

Certificate issuer

Policy OID

Certificate issuer identifier i

▼

*

Authentication strength *

Single-factor authentication

Multi-factor authentication

Affinity binding *

Low

High

Add

Cancel

To create a rule by Policy OID, select **Policy OID**.

a. Enter a value for **Policy OID**.

b. Select **Single-factor authentication**, **Low affinity binding**, and then click **Add**.

When prompted, click **I acknowledge** to finish adding the rule.

Add authentication binding policy rule

Certificate attribute

Certificate issuer

Policy OID

Policy OID i

1.2.3.4

*

Authentication strength *

Single-factor authentication

Multi-factor authentication

Affinity binding *

Low

High

To create a rule by Issuer and Policy OID:

a. Select **Certificate Issuer** and **Policy OID**.

b. Select an issuer and enter the policy OID.

c. For Authentication strength, select **Multifactor authentication**.

d. For Affinity binding, select **High**.

Add authentication binding policy rule

Certificate attribute

Certificate issuer
 Policy OID

Certificate issuer identifier ⓘ

Policy OID ⓘ

Authentication strength *

Single-factor authentication
 Multi-factor authentication

Affinity binding *

Low
 High

Add **Cancel**

e. Select Add.

Certificate-based authentication is a passwordless, phishing-resistant authentication method that uses X.509 certificates and an enterprise public key infrastructure (PKI) for authentication. [Learn more](#).

Enable and Target **Configure**

Authentication binding

The authentication binding policy helps determine the strength of your certificate-based authentication method policy as single-factor or multi-factor and low affinity or high affinity. Override default settings with special rules. [Learn more](#)

Protection Level ⓘ	<input checked="" type="radio"/> Single-factor authentication <input type="radio"/> Multi-factor authentication
Required Affinity Binding ⓘ	<input checked="" type="radio"/> Low <input type="radio"/> High

+ Add rule

Certificate issuer	Policy OID	Authentication strength	Affinity binding
	1.2.3.4	Multi-factor	Low
	1.2.3.4.7	Single-factor	Low
CN=CBATestRootProd	3.4.5.6	Multi-factor	Low

Username binding

Select one of the X.509 certificates fields to bind with one of the user attributes in the cloud. [Learn more](#)

+ Add rule		
Certificate field	Affinity binding	User attribute
PrincipalName	Low	userPrincipalName
RFC222Name	Low	userPrincipalName

Save **Discard**

f. Authenticate with a certificate that has policy OID of 3.4.5.6 and Issued by CN=CBATestRootProd. Authentication should pass and get a multifactor claim.

ⓘ Important

There's a known issue where a Microsoft Entra tenant Authentication Policy Administrator configures a CBA authentication policy rule by using both Issuer and Policy OID. The issue impacts some device registration scenarios, including:

- Windows Hello For Business enrollment

- FIDO2 security key registration
- Windows passwordless phone sign-in

Device registration with Workplace Join, Microsoft Entra ID and Hybrid Microsoft Entra device join scenarios aren't impacted. CBA authentication policy rules using either Issuer OR Policy OID aren't impacted. To mitigate, admins should:

- Edit the certificate-based authentication policy rules that use both Issuer and Policy OID options. Remove either the Issuer or Policy OID requirement and **Save**. -Or-
- Remove the authentication policy rule that uses both Issuer and Policy OID. Create rules that use only Issuer or Policy OID.

We're working to fix the issue.

To create a rule by Issuer and Serial Number:

1. Add an authentication binding policy. The policy requires that any certificate issued by CN=CBATestRootProd with policyOID 1.2.3.4.6 needs only high affinity binding. Issuer and serial number are used.

Certificate-based authentication settings

Certificate-based authentication is a passwordless, phising-resistant authentication method that uses x.509 certificates and an enterprise public key infrastructure (PKI) for authentication. [Learn more](#).

Enable and Target [Configure](#)

Authentication binding

The authentication binding policy helps determine the strength of your certificate-based authentication method policy as single-factor or multi-factor and low affinity or high affinity. Override default settings with special rules. [Learn more](#)

Protection Level	<input type="radio"/> Single-factor authentication <input checked="" type="radio"/> Multi-factor authentication			
Required Affinity Binding	<input checked="" type="radio"/> Low <input type="radio"/> High			
+ Add rule				
Certificate issuer	Policy OID	Authentication strength	Affinity binding	
CN=CBATestRootProd	1.2.3.4.6	Multi-factor	High	...

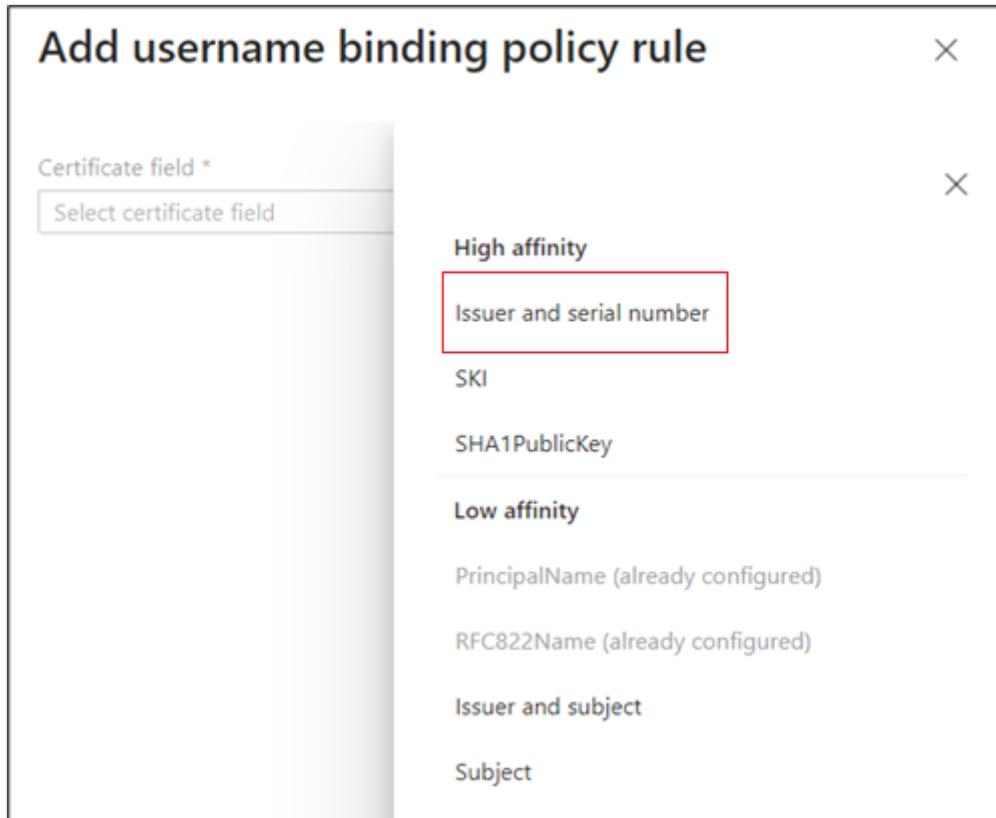
Username binding

Select one of the X.509 certificates fields to bind with one of the user attributes in the cloud. [Learn more](#)

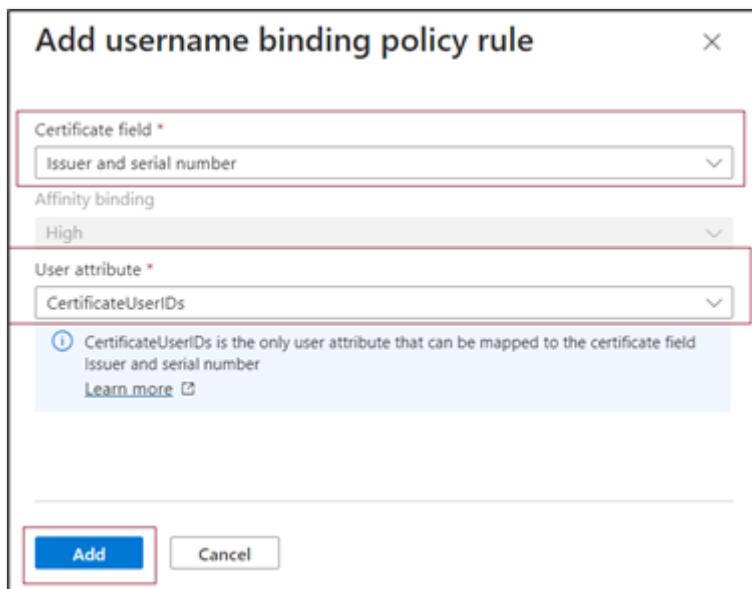
+ Add rule			
Certificate field	Affinity binding	User attribute	
PrincipalName	Low	userPrincipalName	...
RFC822Name	Low	userPrincipalName	...
Issuer and serial number	High	CertificateUserIDs	...

[Save](#) [Discard](#)

2. Select the certificate field. In this example, let's select **Issuer** and **Serial number**.



3. The only user attribute supported is **CertificateUserIds**. Select **Add**.



4. Select **Save**.

The sign-in log shows which binding was used for sign-in, and the details from the certificate.

Activity Details: Sign-ins

Authentication Details	Conditional Access	Report-only	Authentication Events	Additional Details
User certificate subject	CN=			
User certificate issuer	CN=			
User certificate serial number				
User certificate thumbprint				
User certificate valid from	9/21/2022 12:33:47AM			
User certificate expiration	8/18/2023 10:33:18PM			
User certificate binding identifier				
User certificate binding	Certificate:			
User certificate authentication level	singleFactorAuthentication			
User certificate authentication level type	2			
User certificate authentication level identifier	CN=			
Root Key Type	Unknown			

5. Select **Ok** to save any custom rule.

ⓘ Important

Enter the PolicyOID by using the [object identifier format](#). For example, if the certificate policy says **All Issuance Policies**, enter the OID as **2.5.29.32.0** when you add the rule. The string **All Issuance Policies** is invalid for the rules editor and doesn't take effect.

Step 4: Configure username binding policy

The username binding policy helps validate the certificate of the user. By default, we map Principal Name in the certificate to UserPrincipalName in the user object to determine the user.

An Authentication Policy Administrator can override the default and create a custom mapping. To determine how to configure username binding, see [How username binding works](#).

For other scenarios that use the certificateUserIds attribute, see [Certificate user IDs](#).

ⓘ Important

If a username binding policy uses synchronized attributes, such as the certificateUserIds, onPremisesUserPrincipalName, and userPrincipalName attribute

of the user object, be aware that accounts with administrative privileges in Active Directory (such as those with delegated rights on user objects or administrative rights on the Microsoft Entra Connect Server) can make changes that impact these attributes in Microsoft Entra ID.

1. Create the username binding by selecting one of the X.509 certificate fields to bind with one of the user attributes. The username binding order represents the priority level of the binding. The first one has the highest priority, and so on.

The screenshot shows the 'Certificate-based authentication settings' page in the Microsoft Entra ID portal. It includes sections for 'Protection Level' (set to Multi-factor authentication), 'Required Affinity Binding' (set to Low), and a 'Username binding' table. The 'Username binding' table has four columns: Certificate field, Affinity binding, User attribute, and three dots for each row. The rows are: PrincipalName (Low, userPrincipalName), Issuer and serial number (High, CertificateUserIDs), SKI (High, CertificateUserIDs), and RFC822Name (Low, userPrincipalName). A red box highlights the entire 'Username binding' table.

Certificate field	Affinity binding	User attribute	...
PrincipalName	Low	userPrincipalName	...
Issuer and serial number	High	CertificateUserIDs	...
SKI	High	CertificateUserIDs	...
RFC822Name	Low	userPrincipalName	...

If the specified X.509 certificate field is found on the certificate, but Microsoft Entra ID doesn't find a user object using that value, the authentication fails. Microsoft Entra ID tries the next binding in the list.

2. Select **Save** to save the changes.

The final configuration looks like this image:

Dashboard > woodgrove | Security > Security | Authentication methods > Authentication methods | Policies >

Certificate-based authentication settings ...

Certificate-based authentication is a passwordless, unphishable authentication method that uses x.509 certificates and an enterprise public key infrastructure (PKI) for authentication. [Learn more.](#)

Certificate based authentication is usable only as a first-factor authentication method.

Basics **Configure**

Authentication binding

Select the default protection level for all certificate bindings. To override the default, create special rules.

Protection Level Single-factor authentication Multifactor authentication

Add rule

Rule type	Identifier	Protection Level	...
Certificate issuer	CN=WoodgroveCA	Single-factor authentication	...
Policy OID	1.2.3.4	Multifactor authentication	...

Username binding

Select user attribute to create binding. The first certificate field has the highest priority in the username binding.

Certificate field

- PrincipalName
- RFC822Name
- SubjectKeyIdentifier
- SHA1PublicKey

User attribute

- userPrincipalName
- userPrincipalName
- certificateUserids
- Select user attribute

Save **Discard**

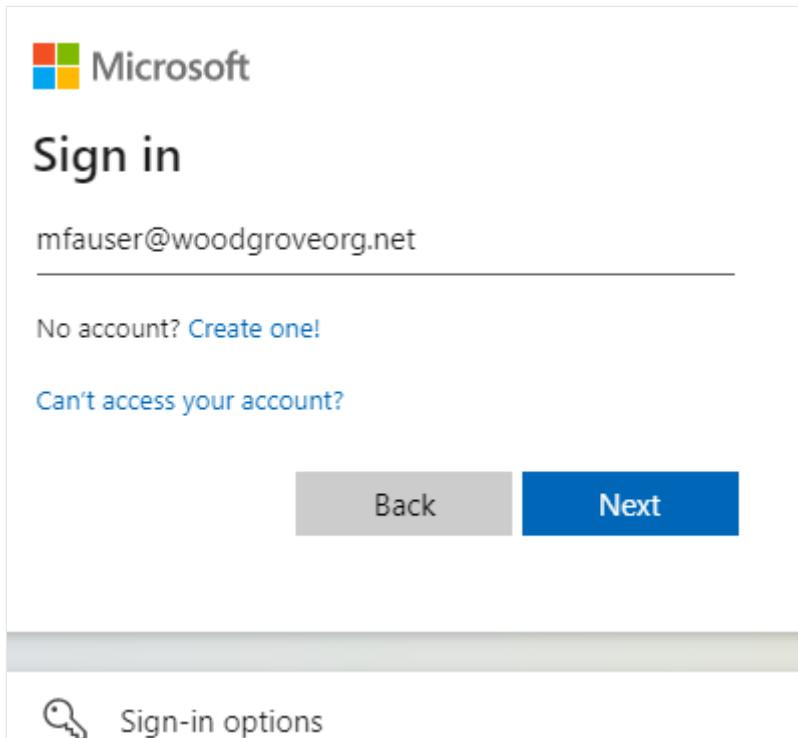
Step 5: Test your configuration

This section covers how to test your certificate and custom authentication binding rules.

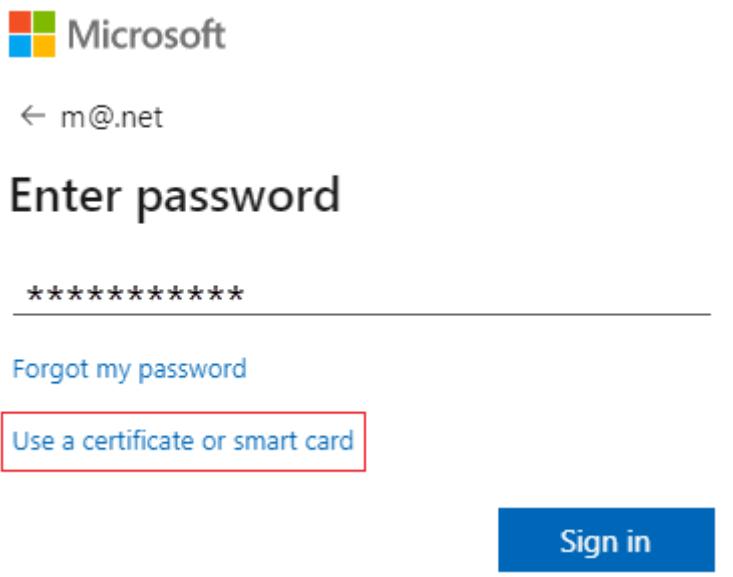
Test your certificate

As a first configuration test, you should try to sign in to the [MyApps portal](#) using your on-device browser.

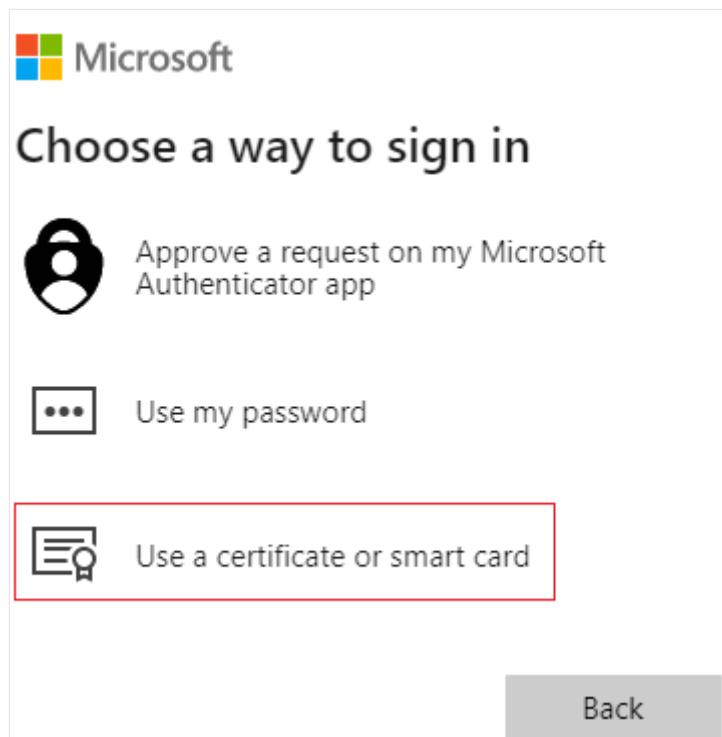
1. Enter your User Principal Name (UPN).



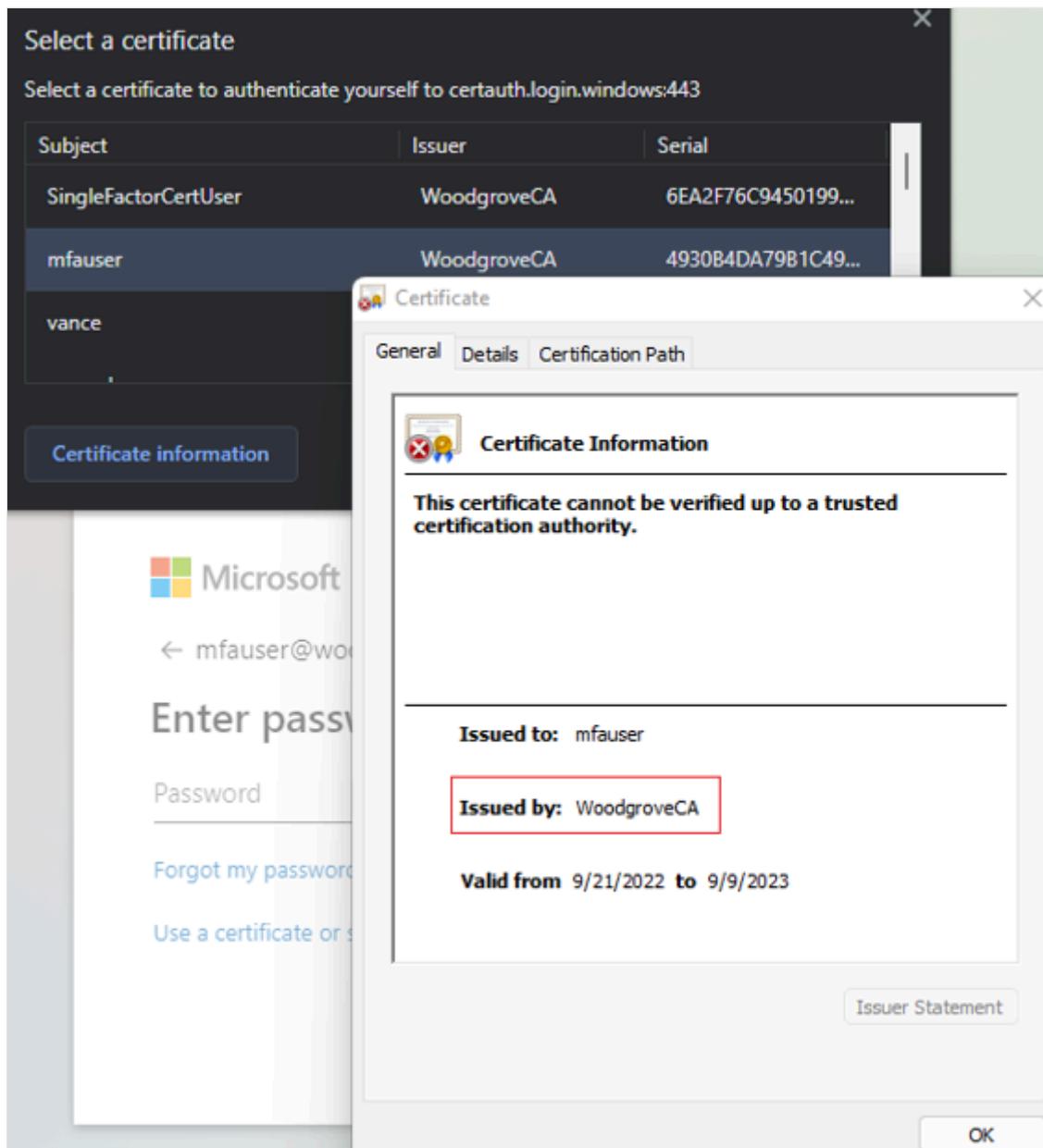
2. Select Next.



If you enabled other authentication methods like Phone sign-in or FIDO2, users might see a different sign-in screen.



3. Select **Sign in with a certificate**.
4. Pick the correct user certificate in the client certificate picker UI and select **OK**.



5. Users should be signed into [MyApps portal](#).

If your sign-in is successful, then you know that:

- The user certificate is provisioned into your test device.
- Microsoft Entra ID is configured correctly with trusted CAs.
- Username binding is configured correctly, and the user is found and authenticated.

Test custom authentication binding rules

Let's walk through a scenario where we validate strong authentication. We create two authentication policy rules, one by using issuer subject to satisfy single-factor authentication, and another by using policy OID to satisfy multifactor authentication.

1. Create an issuer Subject rule with protection level as single-factor authentication and value set to your CAs Subject value. For example:

CN = WoodgroveCA

2. Create a policy OID rule, with protection level as multifactor authentication and value set to one of the policy OIDs in your certificate. For example, 1.2.3.4.

The screenshot shows the 'Certificate-based authentication settings' page. At the top, there's a breadcrumb navigation: Dashboard > woodgrove > Security > Authentication methods > Policies >. Below the header, a note states: 'Certificate-based authentication is a passwordless, unphishable authentication method that uses x.509 certificates and an enterprise public key infrastructure (PKI) for authentication. Learn more.' A 'Configure' button is present. Under 'Authentication binding', it says: 'Select the default protection level for all certificate bindings. To override the default, create special rules.' A 'Protection Level' dropdown is set to 'Single-factor authentication'. An 'Add rule' button is available. A table lists a single rule:

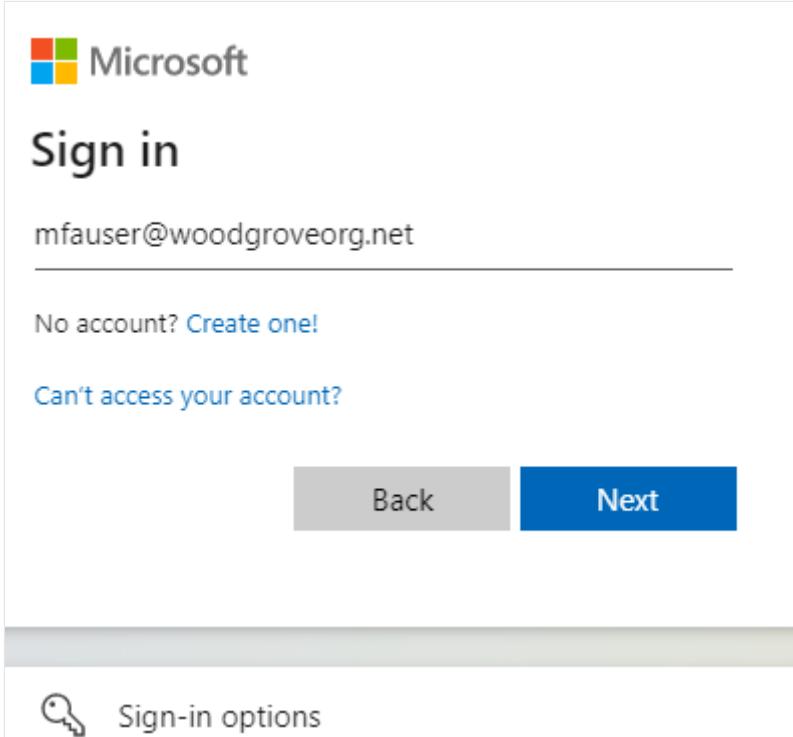
Rule type	Identifier	Protection Level
Certificate issuer	CN=WoodgroveCA	Single-factor authentication
Policy OID	1.2.3.4	Multifactor authentication

A 'Username binding' section follows, with a note: 'Select user attribute to create binding. The first certificate field has the highest priority in the username binding.' It shows a table of certificate fields and their corresponding user attributes:

Certificate field	User attribute
PrincipalName	userPrincipalName
RFC822Name	userPrincipalName
SubjectKeyIdentifier	Select user attribute
SHA1PublicKey	Select user attribute

At the bottom are 'Save' and 'Discard' buttons.

3. Create a Conditional Access policy for the user to require multifactor authentication by following steps at [Conditional Access - Require MFA](#).
4. Navigate to [MyApps portal](#). Enter your UPN and select **Next**.



5. Select **Sign in with a certificate**.



← m@.net

Enter password

[Forgot my password](#)

[Use a certificate or smart card](#)

Sign in

If you enabled other authentication methods like Phone sign-in or security keys, users might see a different sign-in screen.



Choose a way to sign in



Approve a request on my Microsoft Authenticator app



Use my password



Use a certificate or smart card

Back

6. Select the client certificate and select **Certificate Information**.

Select a certificate

Select a certificate to authenticate yourself to certauth.login.microsoftonline.com:443

Subject	Issuer	Serial
sfactor	ContosoCA	74C8F4E322A633BC4...
mfauserwithnocert	ContosoCA	251E322927597F8849...
mfauser	ContosoCA	2CC3681EBE7B0CBC4...

Certificate information OK Cancel

 Microsoft

← mfauser@contoso.com

Enter password

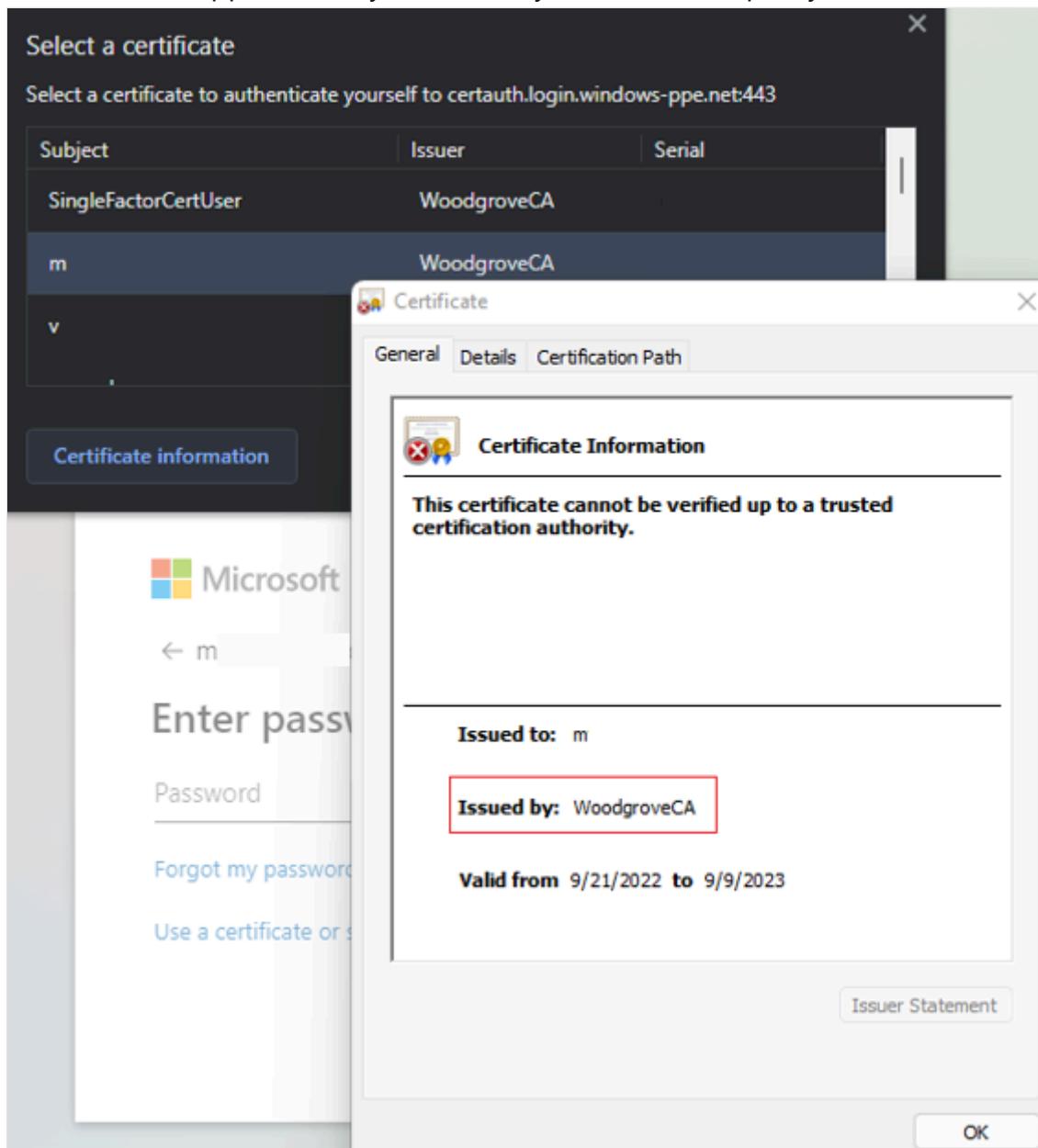
Password

[Forgot my password](#)

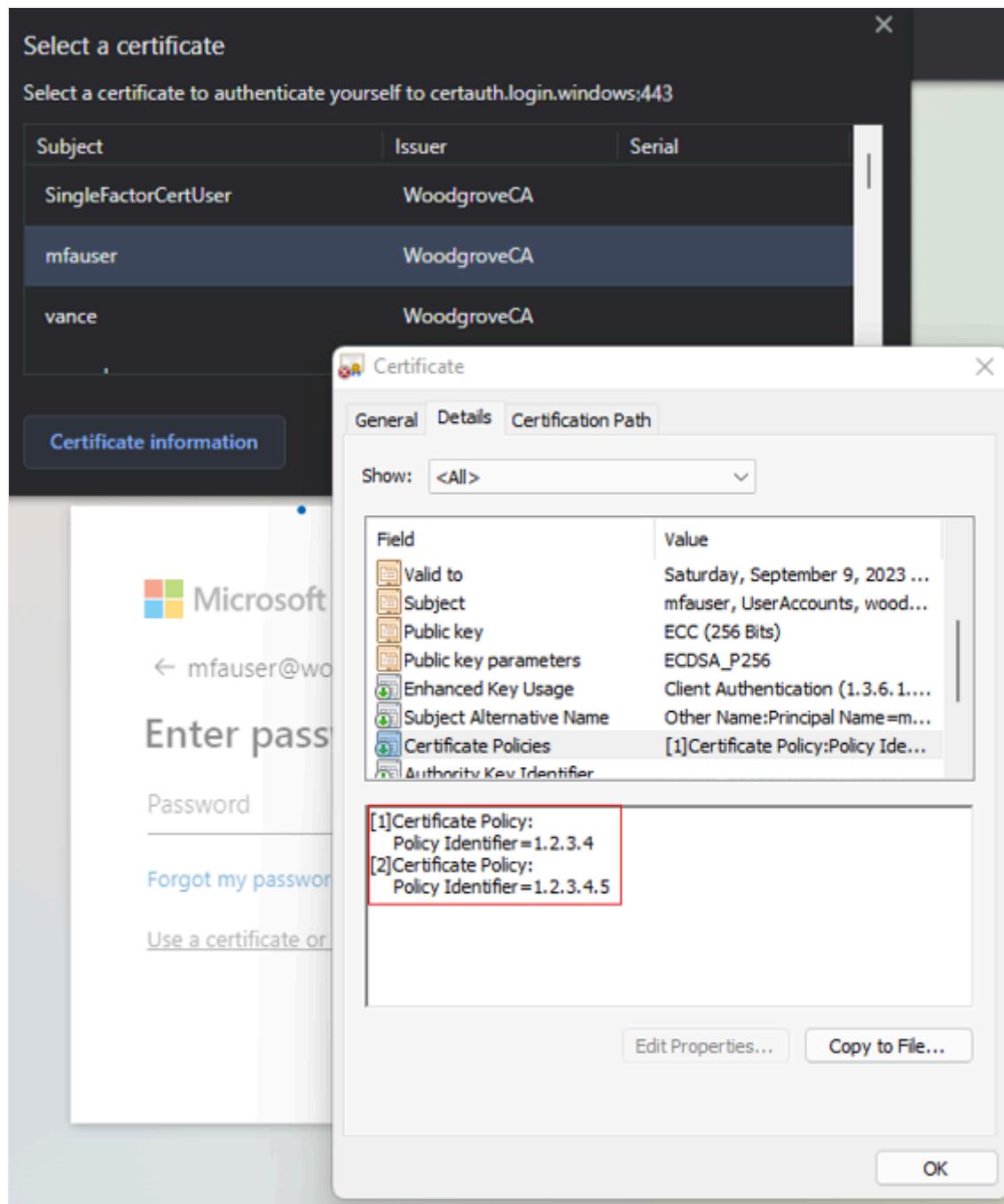
[Sign in with a certificate](#)

[Sign in](#)

7. The certificate appears, and you can verify the issuer and policy OID values.



8. To see Policy OID values, select **Details**.



9. Select the client certificate and select OK.
10. The policy OID in the certificate matches the configured value of 1.2.3.4, and satisfies multifactor authentication. Similarly, the issuer in the certificate matches the configured value of CN=WoodgroveCA, and satisfies single-factor authentication.
11. Because the policy OID rule takes precedence over the issuer rule, the certificate satisfies multifactor authentication.
12. The Conditional Access policy for the user requires MFA and the certificate satisfies multifactor, so the user can sign in to the application.

Test username binding policy

The username binding policy helps validate the certificate of the user. There are three bindings that are supported for the username binding policy:

- **IssuerAndSerialNumber > CertificateUserIds**
- **IssuerAndSubject > CertificateUserIds**
- **Subject > CertificateUserIds**

By default, Microsoft Entra ID maps **Principal Name** in the certificate to **UserPrincipalName** in the user object to determine the user. An Authentication Policy Administrator can override the default and create a custom mapping, as explained earlier.

An Authentication Policy Administrator needs to enable the new bindings. To prepare, they must make sure the correct values for the corresponding username bindings are updated in the **CertificateUserIds** attribute of the user object:

- For cloud only users, use the [Microsoft Entra admin center](#) or [Microsoft Graph APIs](#) to update the value in CertificateUserIds.
- For on-premises synced users, use Microsoft Entra Connect to sync the values from on-premises by following [Microsoft Entra Connect Rules](#) or [syncing AltSeclId value](#).

Important

The format of the values of Issuer, Subject, and SerialNumber should be in the reverse order of their format in the certificate. Don't add any space in the Issuer or Subject.

Issuer and Serial Number manual mapping

Here's an example for Issuer and Serial Number manual mapping. The Issuer value to be added is:

```
C=US,0=U.SGovernment,OU=DoD,OU=PKI,OU=CONTRACTOR,CN=CRL.BALA.SelfSignedCertificate
```

Field	Value
Version	V3
Serial number	
Signature algorithm	
Signature hash algorithm	
Issuer	SelfSignedCertificate, ...
Valid from	Wednesday 26 April 2023 13:5...
Valid to	Friday 26 April 2024 14:16:38
Subject	SelfSignedCertificate, ...

CN = SelfSignedCertificate
OU = CONTRACTOR
OU = PKI
OU = DoD
O = U.S. Government
C = US

To get the correct value for serial number, run the following command, and store the value shown in CertificateUserIds. The command syntax is:

```
Certutil -dump -v [~certificate path~] >> [~dumpFile path~]
```

For example:

```
certutil -dump -v firstusercert.cer >> firstCertDump.txt
```

Here's an example for the certutil command:

```
certutil -dump -v C:\save\CBA\certs\CBATestRootProd\mfausercer.cer
```

```
X509 Certificate:  
Version: 3  
Serial Number: 48efa06ba8127299499b069f133441b2
```

```
b2 41 34 13 9f 06 9b 49 99 72 12 a8 6b a0 ef 48
```

The SerialNumber value to be added in CertificateUserId is:

b24134139f069b49997212a86ba0ef48

CertificateUserId:

```
X509:  
<I>C=US,O=U.SGovernment,OU=DoD,OU=PKI,OU=CONTRACTOR,CN=CRL.BALA.SelfSignedCe  
rtificate<SR> b24134139f069b49997212a86ba0ef48
```

Issue and Subject manual mapping

Here's an example for Issue and Subject manual mapping. The Issuer value is:

Field	Value
Version	V3
Serial number	
Signature algorithm	
Signature hash algorithm	
Issuer	SelfSignedCertificate, ...
Valid from	Wednesday 26 April 2023 13:5...
Valid to	Friday 26 April 2024 14:16:38
Subject	SelfSignedCertificate, ...
... (2 more)	

CN = SelfSignedCertificate
OU = CONTRACTOR
OU = PKI
OU = DoD
O = U.S. Government
C = US

The Subject value is:

Subject	FirstUserATCSession, UserAcc...
Public key	FCC (256 Bits)
<p>CN = FirstUserATCSession OU = UserAccounts DC = corp DC = contoso DC = com</p>	

CertificateUserId:

X509: <I>C=US,O=U.SGovernment,OU=DoD,OU=PKI,OU=CONTRACTOR,CN=CRL.BALA.SelfSignedCe rtificate<S> DC=com,DC=contoso,DC=corp,OU=UserAccounts,CN=FirstUserATCSession

Subject manual mapping

Here's an example for Subject manual mapping. The Subject value is:

The screenshot shows a certificate details interface. The 'Subject' field contains the value 'FirstUserATCSession, UserAcc...'. Below it, the 'Public key' field shows 'ECC (256 Bits)'. A large text area below lists the following attributes:
CN = FirstUserATCSession
OU = UserAccounts
DC = corp
DC = contoso
DC = com

CertificateUserId:

The screenshot shows a certificate details interface. A large text area displays the X509 certificate string:
X509:<S>DC=com,DC=contoso,DC=corp,OU=UserAccounts,CN=FirstUserATCSession

Test affinity binding

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Policies**.
3. Under **Manage**, select **Authentication methods > Certificate-based Authentication**.
4. Select **Configure**.
5. Set **Required Affinity Binding** at the tenant level.

i Important

Be careful with the tenant-wide affinity setting. You can lock out the entire tenant if you change the **Required Affinity Binding** for the tenant and you don't have proper values in the user object. Similarly, if you create a custom rule that applies to all users and requires high affinity binding, users in the tenant can get locked out.

Home > vimrangorg1 | Security > Security | Authentication methods > Authentication methods | Policies >

Certificate-based authentication settings

Certificate-based authentication is a passwordless, phishing-resistant authentication method that uses X.509 certificates and an enterprise public key infrastructure (PKI) for authentication. [Learn more.](#)

Enable and Target [Configure](#)

Authentication binding

The authentication binding policy helps determine the strength of your certificate-based authentication method policy as single-factor or multi-factor and low affinity or high affinity. Override default settings with special rules. [Learn more.](#)

Protection Level	<input checked="" type="radio"/> Single-factor authentication <input type="radio"/> Multi-factor authentication
Required Affinity Binding	<input type="radio"/> Low <input checked="" type="radio"/> High

[+ Add rule](#)

Certificate issuer	Policy OID	Authentication strength	Affinity binding
	1.2.3.4	Multi-factor	Low
	1.2.3.4.7	Single-factor	Low

Username binding

Select one of the X.509 certificates fields to bind with one of the user attributes in the cloud. [Learn more](#)

⚠ Please add at least one username binding policy rule with high affinity to satisfy your authentication binding policy.

[+ Add rule](#)

Certificate field	Affinity binding	User attribute
PrincipalName	Low	userPrincipalName
RFC822Name	Low	userPrincipalName

[Save](#) [Discard](#)

6. To test, select Required Affinity Binding to be Low.

7. Add a high affinity binding like SKI. Select **Add rule** under **Username binding**.

8. Select SKI and select **Add**.

Home > vimrangorg1 | Security > Security | Auth

Certificate-based authentication

from using CBA as second factor or registering other users.

Certificate-based authentication is a passwordless, phishing-resistant authentication method that uses X.509 certificates and an enterprise public key infrastructure (PKI) for authentication. [Learn more.](#)

Enable and Target [Configure](#)

Authentication binding

The authentication binding policy helps determine the strength of your certificate-based authentication method policy as single-factor or multi-factor and low affinity or high affinity. Override default settings with special rules. [Learn more.](#)

Protection Level	<input checked="" type="radio"/> Single-factor authentication <input type="radio"/> Multi-factor authentication
Required Affinity Binding	<input type="radio"/> Low <input checked="" type="radio"/> High

[+ Add rule](#)

Certificate issuer	Policy OID	Authentication strength	Affinity binding
	1.2.3.4	Multi-factor	Low
	1.2.3.4.7	Single-factor	Low

Username binding

Select one of the X.509 certificates fields to bind with one of the user attributes in the cloud. [Learn more](#)

⚠ Please add at least one username binding policy rule with high affinity to satisfy your authentication binding policy.

Add username binding policy rule

Certificate field *
Select certificate field

High affinity

- SKI
- SHA1PublicKey

Low affinity

- PrincipalName
- RFC822Name

[+ Add rule](#)

Certificate field	Affinity binding	User attribute
PrincipalName	Low	userPrincipalName
RFC822Name	Low	userPrincipalName

[Save](#) [Discard](#) [Add](#) [Cancel](#)

When finished, the rule looks like this screenshot:

The screenshot shows the 'Authentication binding' section of a certificate-based authentication policy. It includes fields for 'Protection Level' (Single-factor authentication selected), 'Required Affinity Binding' (Low selected), and a table for 'Certificate issuer' mapping. The 'Username binding' section shows mappings for 'PrincipalName', 'RFC822Name', and 'SKI'. The 'SKI' row is highlighted with a red border. At the bottom are 'Save' and 'Discard' buttons.

9. Update all user objects CertificateUserIds attribute to have the correct value of SKI from the user certificate. For more information, see [Supported patterns for CertificateUserIDs](#).
10. Create a custom rule for Authentication binding.
11. Select Add.

The dialog shows options for 'Certificate attribute' (unchecked) and 'Policy OID' (checked). The 'Policy OID' field contains '9.8.7.5'. Under 'Authentication strength', 'Multi-factor authentication' is selected. Under 'Affinity binding', 'High' is selected. At the bottom are 'Add' and 'Cancel' buttons.

When finished, the rule looks like this screenshot:

Certificate-based authentication is a passwordless, phishing-resistant authentication method that uses x509 certificates and an enterprise public key infrastructure (PKI) for authentication. [Learn more](#).

Enable and Target [Configure](#)

Authentication binding

The authentication binding policy helps determine the strength of your certificate-based authentication method policy as single-factor or multi-factor and low affinity or high affinity. Override default settings with special rules. [Learn more](#)

Protection Level (i)	<input checked="" type="radio"/> Single-factor authentication <input type="radio"/> Multi-factor authentication
Required Affinity Binding (i)	<input checked="" type="radio"/> Low <input type="radio"/> High

[+ Add rule](#)

Certificate Issuer	Policy OID	Authentication strength	Affinity binding	...
	1.2.3.4	Multi-factor	Low	...
	1.2.3.4.7	Single-factor	Low	...
	9.8.7.5	Multi-factor	High	...

Username binding

Select one of the X.509 certificates fields to bind with one of the user attributes in the cloud. [Learn more](#)

[+ Add rule](#)

Certificate field	Affinity binding	User attribute	...
PrincipalName	Low	userPrincipalName	...
RFC822Name	Low	userPrincipalName	...
SKI	High	CertificateUserIds	...

[Save](#) [Discard](#)

12. Update the user CertificateUserIds with correct SKI value from the certificate with policy OID 9.8.7.5.
13. Test with a certificate with policy OID 9.8.7.5 and the user should be authenticated with SKI binding and get MFA with only the certificate.

Enable CBA using Microsoft Graph API

To enable CBA and configure username bindings using Graph API, complete the following steps.

1. Go to [Microsoft Graph Explorer](#).
2. Select **Sign into Graph Explorer** and sign in to your tenant.
3. Follow the steps to [consent to the Policy.ReadWrite.AuthenticationMethod delegated permission](#).
4. GET all authentication methods:

```
HTTP
GET
https://graph.microsoft.com/v1.0/policies/authenticationmethodspolicy
```

5. GET the configuration for the x509 Certificate authentication method:

HTTP

GET

<https://graph.microsoft.com/v1.0/policies/authenticationmethodsPolicy/authenticationMethodConfigurations/X509Certificate>

6. By default, the x509 Certificate authentication method is disabled. To allow users to sign in with a certificate, you must enable the authentication method and configure the authentication and username binding policies through an update operation. To update policy, run a PATCH request.

Request body:

HTTP

PATCH

<https://graph.microsoft.com/v1.0/policies/authenticationMethodsPolicy/authenticationMethodConfigurations/x509Certificate>

Content-Type: application/json

```
{  
    "@odata.type":  
        "#microsoft.graph.x509CertificateAuthenticationMethodConfiguration",  
    "id": "X509Certificate",  
    "state": "enabled",  
    "certificateUserBindings": [  
        {  
            "x509CertificateField": "PrincipalName",  
            "userProperty": "onPremisesUserPrincipalName",  
            "priority": 1  
        },  
        {  
            "x509CertificateField": "RFC822Name",  
            "userProperty": "userPrincipalName",  
            "priority": 2  
        },  
        {  
            "x509CertificateField": "PrincipalName",  
            "userProperty": "certificateUserIds",  
            "priority": 3  
        }  
    ],  
    "authenticationModeConfiguration": {  
        "x509CertificateAuthenticationDefaultMode":  
            "x509CertificateSingleFactor",  
        "rules": [  
            {  
                "x509CertificateRuleType": "issuerSubject",  
                "identifier": "CN=WoodgroveCA ",  
                "x509CertificateAuthenticationMode":  
                    "x509CertificateMultiFactor"  
            }  
        ]  
    }  
}
```

```

    },
    {
        "x509CertificateRuleType": "policyOID",
        "identifier": "1.2.3.4",
        "x509CertificateAuthenticationMode":
    "x509CertificateMultiFactor"
    }
]
},
"includeTargets": [
{
    "targetType": "group",
    "id": "all_users",
    "isRegistrationRequired": false
}
]
}

```

7. You get a `204 No content` response code. Rerun the GET request to make sure the policies are updated correctly.
8. Test the configuration by signing in with a certificate that satisfies the policy.

Enable CBA using Microsoft PowerShell

1. Open PowerShell.
2. Connect to Microsoft Graph:

```

PowerShell

Connect-MgGraph -Scopes "Policy.ReadWrite.AuthenticationMethod"

```

3. Create a variable for defining group for CBA users:

```

PowerShell

$group = Get-MgGroup -Filter "displayName eq 'CBATestGroup'"

```

4. Define the request body:

```

PowerShell

$body = @{
"@odata.type" =
"#microsoft.graph.x509CertificateAuthenticationMethodConfiguration"
"id" = "X509Certificate"
"state" = "enabled"

```

```

"certificateUserBindings" = @(
    @{
        "@odata.type" = "#microsoft.graph.x509CertificateUserBinding"
        "x509CertificateField" = "SubjectKeyIdentifier"
        "userProperty" = "certificateUserIds"
        "priority" = 1
    },
    @{
        "@odata.type" = "#microsoft.graph.x509CertificateUserBinding"
        "x509CertificateField" = "PrincipalName"
        "userProperty" = "UserPrincipalName"
        "priority" = 2
    },
    @{
        "@odata.type" = "#microsoft.graph.x509CertificateUserBinding"
        "x509CertificateField" = "RFC822Name"
        "userProperty" = "userPrincipalName"
        "priority" = 3
    }
)
"authenticationModeConfiguration" = @{
    "@odata.type" =
    "#microsoft.graph.x509CertificateAuthenticationModeConfiguration"
    "x509CertificateAuthenticationDefaultMode" =
    "x509CertificateMultiFactor"
    "rules" = @(
        @{
            "@odata.type" = "#microsoft.graph.x509CertificateRule"
            "x509CertificateRuleType" = "policyOID"
            "identifier" = "1.3.6.1.4.1.311.21.1"
            "x509CertificateAuthenticationMode" =
            "x509CertificateMultiFactor"
        }
    )
}
"includeTargets" = @(
    @{
        "targetType" = "group"
        "id" = $group.Id
        "isRegistrationRequired" = $false
    }
) } | ConvertTo-Json -Depth 5

```

5. Run the PATCH request:

PowerShell

```

Invoke-MgGraphRequest -Method PATCH -Uri
"https://graph.microsoft.com/v1.0/policies/authenticationMethodsPolicy/
authenticationMethodConfigurations/x509Certificate" -Body $body -
ContentType "application/json"

```

Next steps

- Overview of Microsoft Entra CBA
 - Technical deep dive for Microsoft Entra CBA
 - Limitations with Microsoft Entra CBA
 - Windows SmartCard logon using Microsoft Entra CBA
 - Microsoft Entra CBA on mobile devices (Android and iOS)
 - Certificate user IDs
 - How to migrate federated users
 - FAQ
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

How to configure certificate authorities for Microsoft Entra certificate-based authentication

Article • 03/04/2025

The best way to configure the certificate authorities (CAs) is with the PKI-based trust store (Preview). You can delegate configuration with a PKI-based trust store to least privileged roles. For more information see, [Step 1: Configure the certificate authorities with PKI-based trust store \(Preview\)](#).

As an alternative, a Global Administrator can follow steps in this topic to configure CAs by using the Microsoft Entra admin center, or Microsoft Graph REST APIs and the supported software development kits (SDKs), such as Microsoft Graph PowerShell.

i Important

Microsoft recommends that you use roles with the fewest permissions. This practice helps improve security for your organization. Global Administrator is a highly privileged role that should be limited to emergency scenarios or when you can't use an existing role.

The public key infrastructure (PKI) infrastructure or PKI admin should be able to provide the list of issuing CAs.

To make sure you configured all the CAs, open the user certificate and click **Certification path** tab. Make sure every CA until the root is uploaded to the Microsoft Entra ID trust store. Microsoft Entra certificate-based authentication (CBA) fails if there are missing CAs.

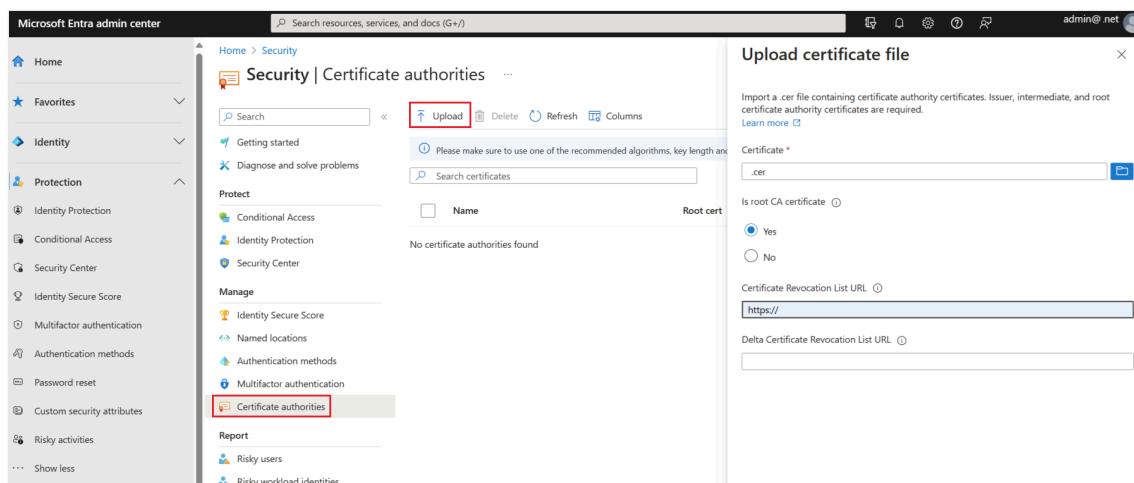
Configure certificate authorities using the Microsoft Entra admin center

To configure certificate authorities to enable CBA in the Microsoft Entra admin center, complete the following steps:

i Important

Microsoft recommends that you use roles with the fewest permissions. This practice helps improve security for your organization. Global Administrator is a highly privileged role that should be limited to emergency scenarios or when you can't use an existing role.

1. Sign in to the Microsoft Entra admin center [↗](#) as a **Global Administrator**.
2. Browse to **Protection > Show more > Security Center (or Identity Secure Score) > Certificate authorities**.
3. To upload a CA, select **Upload**:
 - a. Select the CA file.
 - b. Select **Yes** if the CA is a root certificate, otherwise select **No**.
 - c. For **Certificate Revocation List URL**, set the internet-facing URL for the CA base CRL that contains all revoked certificates. If the URL isn't set, authentication with revoked certificates doesn't fail.
 - d. For **Delta Certificate Revocation List URL**, set the internet-facing URL for the CRL that contains all revoked certificates since the last base CRL was published.
- e. Select **Add**.



4. To delete a CA certificate, select the certificate and select **Delete**.
5. Select **Columns** to add or delete columns.

ⓘ Note

Upload of a new CA fails if any existing CA expired. You should delete any expired CA, and retry to upload the new CA.

Configure certificate authorities (CA) using PowerShell

Only one CRL Distribution Point (CDP) for a trusted CA is supported. The CDP can only be HTTP URLs. Online Certificate Status Protocol (OCSP) or Lightweight Directory Access Protocol (LDAP) URLs aren't supported.

To configure your certificate authorities in Microsoft Entra ID, for each certificate authority, upload the following:

- The public portion of the certificate, in .cer format
- The internet-facing URLs where the Certificate Revocation Lists (CRLs) reside

The schema for a certificate authority looks as follows:

C#

```
class TrustedCAsForPasswordlessAuth
{
    CertificateAuthorityInformation[] certificateAuthorities;
}

class CertificateAuthorityInformation
{
    CertAuthorityType authorityType;
    X509Certificate trustedCertificate;
    string crlDistributionPoint;
    string deltaCrlDistributionPoint;
    string trustedIssuer;
    string trustedIssuerSKI;
}

enum CertAuthorityType
{
    RootAuthority = 0,
    IntermediateAuthority = 1
}
```

For the configuration, you can use [Microsoft Graph PowerShell](#):

1. Start Windows PowerShell with administrator privileges.
2. Install [Microsoft Graph PowerShell](#):

PowerShell

```
Install-Module Microsoft.Graph
```

As a first configuration step, you need to establish a connection with your tenant. As soon as a connection to your tenant exists, you can review, add, delete, and modify the trusted certificate authorities that are defined in your directory.

Connect

To establish a connection with your tenant, use [Connect-MgGraph](#):

PowerShell

```
Connect-MgGraph
```

Retrieve

To retrieve the trusted certificate authorities that are defined in your directory, use [Get-MgOrganizationCertificateBasedAuthConfiguration](#).

PowerShell

```
Get-MgOrganizationCertificateBasedAuthConfiguration
```

Add

① Note

Upload of new CAs will fail when any of the existing CAs are expired. Tenant Admin should delete the expired CAs and then upload the new CA.

Follow the preceding steps to add a CA in the Microsoft Entra admin center.

AuthorityType

- Use 0 to indicate a Root certificate authority
- Use 1 to indicate an Intermediate or Issuing certificate authority

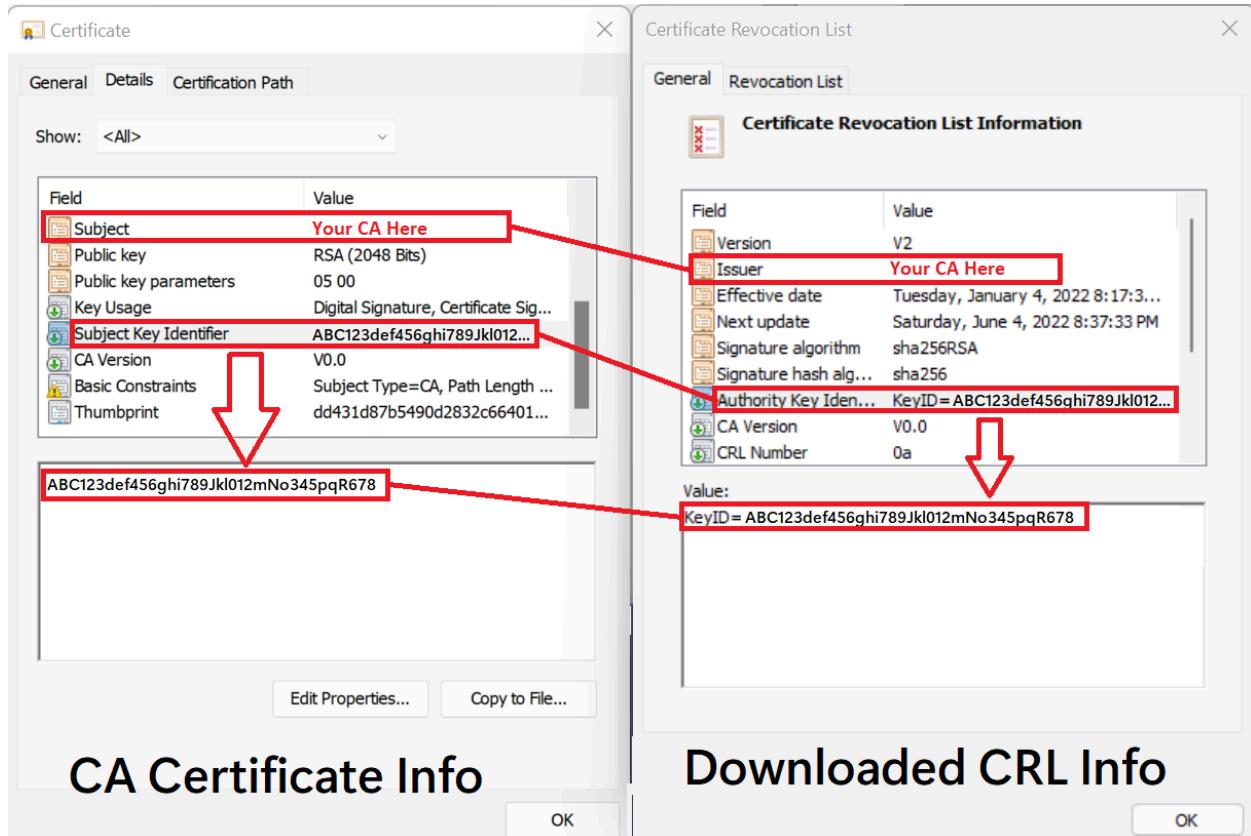
crlDistributionPoint

Download the CRL and compare the CA certificate and the CRL information. Make sure the crlDistributionPoint value in the preceding PowerShell example is valid for the CA you want to add.

The following table and graphic show how to map information from the CA certificate to the attributes of the downloaded CRL.

[Expand table](#)

CA Certificate Info	=	Downloaded CRL Info
Subject	=	Issuer
Subject Key Identifier	=	Authority Key Identifier (KeyID)



CA Certificate Info

Downloaded CRL Info

Tip

The value for `crlDistributionPoint` in the preceding example is the http location for the CA's Certificate Revocation List (CRL). This value can be found in a few places:

- In the CRL Distribution Point (CDP) attribute of a certificate issued from the CA.

If the issuing CA runs Windows Server:

- On the [Properties](#) of the CA in the certificate authority Microsoft Management Console (MMC).

- On the CA by running `certutil -cainfo cdp`. For more information, see [certutil](#).

For more information, see [Understanding the certificate revocation process](#).

Configure certificate authorities using the Microsoft Graph APIs

Microsoft Graph APIs can be used to configure certificate authorities. To update the Microsoft Entra Certificate Authority trust store, follow the steps at [certificatebasedauthconfiguration MSGraph commands](#).

Validate Certificate Authority configuration

Make sure the configuration allows Microsoft Entra CBA to:

- Validate the CA trust chain
- Get the certificate revocation list (CRL) from the configured certificate authority CRL distribution point (CDP)

To validate the CA configuration, install the [MSIdentity Tools](#) PowerShell module, and run [Test-MsIdCBATrustStoreConfiguration](#). This PowerShell cmdlet reviews the Microsoft Entra tenant CA configuration. It reports errors and warnings for common misconfigurations.

Related content

[How to configure Microsoft Entra certificate-based authentication](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Windows smart card sign-in using Microsoft Entra certificate-based authentication

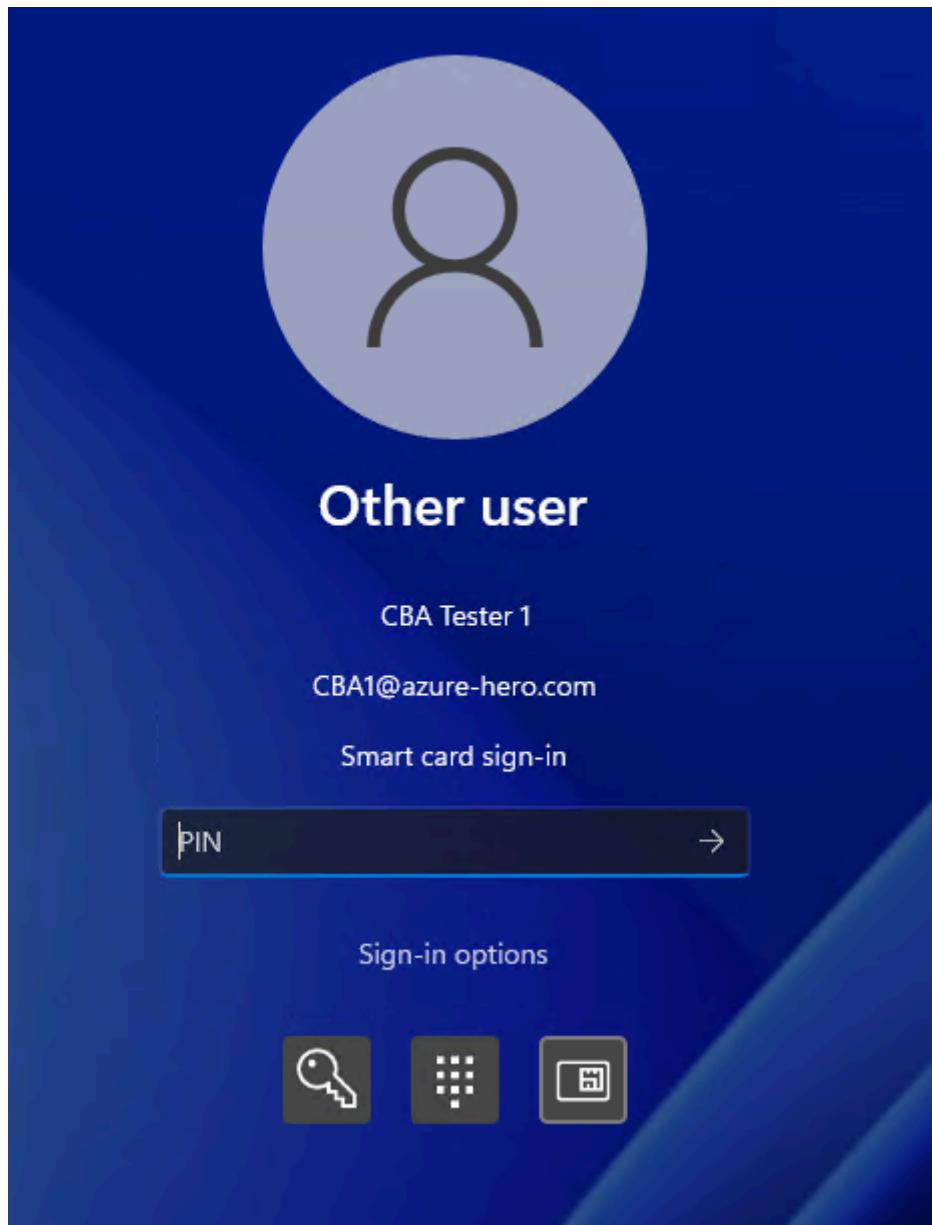
Article • 03/04/2025

Microsoft Entra users can authenticate using X.509 certificates on their smart cards directly against Microsoft Entra ID at Windows sign-in. There's no special configuration needed on the Windows client to accept the smart card authentication.

User experience

Follow these steps to set up Windows smart card sign-in:

1. Join the machine to either Microsoft Entra ID or a hybrid environment (hybrid join).
2. Configure Microsoft Entra CBA in your tenant as described in [Configure Microsoft Entra CBA](#).
3. Make sure the user is either on managed authentication or using [Staged Rollout](#).
4. Present the physical or virtual smart card to the test machine.
5. Select the smart card icon, enter the PIN, and authenticate the user.



Users will get a primary refresh token (PRT) from Microsoft Entra ID after the successful sign-in. Depending on the CBA configuration, the PRT will contain the multifactor claim.

Expected behavior of Windows sending user UPN to Microsoft Entra CBA

[Expand table](#)

Sign-in	Microsoft Entra join	Hybrid join
First sign-in	Pull from certificate	AD UPN or x509Hint
Subsequent sign-in	Pull from certificate	Cached Microsoft Entra UPN

Windows rules for sending UPN for Microsoft Entra joined devices

Windows will first use a principal name and if not present then RFC822Name from the SubjectAlternativeName (SAN) of the certificate being used to sign into Windows. If neither are present, the user must additionally supply a User Name Hint. For more information, see [User Name Hint](#)

Windows rules for sending UPN for Microsoft Entra hybrid joined devices

Hybrid Join sign-in must first successfully sign-in against the Active Directory(AD) domain. The users AD UPN is sent to Microsoft Entra ID. In most cases, the Active Directory UPN value is the same as the Microsoft Entra UPN value and is synchronized with Microsoft Entra Connect.

Some customers may maintain different and sometimes may have non-routable UPN values in Active Directory (such as user@woodgrove.local) In these cases the value sent by Windows may not match the users Microsoft Entra UPN. To support these scenarios where Microsoft Entra ID can't match the value sent by Windows, a subsequent lookup is performed for a user with a matching value in their **onPremisesUserPrincipalName** attribute. If the sign-in is successful, Windows will cache the users Microsoft Entra UPN and is sent in subsequent sign-ins.

Note

In all cases, a user supplied username login hint (X509UserNameHint) will be sent if provided. For more information, see [User Name Hint](#)

Important

If a user supplies a username login hint (X509UserNameHint), the value provided **MUST** be in UPN Format.

For more information about the Windows flow, see [Certificate Requirements and Enumeration \(Windows\)](#).

Supported Windows platforms

The Windows smart card sign-in works with the latest preview build of Windows 11. The functionality is also available for these earlier Windows versions after you apply one of the following updates [KB5017383](#):

- [Windows 11 - kb5017383](#)
- [Windows 10 - kb5017379](#)
- [Windows Server 20H2 - kb5017380](#)
- [Windows Server 2022 - kb5017381](#)
- [Windows Server 2019 - kb5017379](#)

Supported browsers

[\[+\] Expand table](#)

Edge	Chrome	Safari	Firefox

Note

Microsoft Entra CBA supports both certificates on-device as well as external storage like security keys on Windows.

Windows Out of the box experience (OOBE)

Windows OOBE should allow the user to login using an external smart card reader and authenticate against Microsoft Entra CBA. Windows OOBE by default should have the necessary smart card drivers or the smart card drivers previously added to the Windows image before OOBE setup.

Restrictions and caveats

- Microsoft Entra CBA is supported on Windows devices that are hybrid or Microsoft Entra joined.
- Users must be in a managed domain or using Staged Rollout and can't use a federated authentication model.

Next steps

- Overview of Microsoft Entra CBA
 - Technical deep dive for Microsoft Entra CBA
 - How to configure Microsoft Entra CBA
 - Microsoft Entra CBA on iOS devices
 - Microsoft Entra CBA on Android devices
 - Certificate user IDs
 - How to migrate federated users
 - FAQ
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Microsoft Entra certificate-based authentication on iOS and macOS

Article • 03/04/2025

This topic covers Microsoft Entra certificate-based authentication (CBA) support for macOS and iOS devices.

Microsoft Entra certificate-based authentication on macOS devices

Devices that run macOS can use CBA to authenticate against Microsoft Entra ID by using their X.509 client certificate. Microsoft Entra CBA is supported with certificates on-device and external hardware protected security keys. On macOS, Microsoft Entra CBA is supported on all browsers and on Microsoft first-party applications.

Browsers supported on macOS

[Expand table](#)

Edge	Chrome	Safari	Firefox
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

macOS device sign-in with Microsoft Entra CBA

Microsoft Entra CBA today isn't supported for device-based sign-in to macOS machines. The certificate used to sign in to the device can be the same certificate used to authenticate to Microsoft Entra ID from a browser or desktop application, but the device sign-in itself isn't supported against Microsoft Entra ID yet.

Microsoft Entra certificate-based authentication on iOS devices

Devices that run iOS can use certificate-based authentication (CBA) to authenticate to Microsoft Entra ID using a client certificate on their device when connecting to:

- Office mobile applications such as Microsoft Outlook and Microsoft Word
- Exchange ActiveSync (EAS) clients

Microsoft Entra CBA is supported for certificates on-device on native browsers and on Microsoft first-party applications on iOS devices.

Prerequisites

- iOS version must be iOS 9 or later.
- Microsoft Authenticator is required for Office applications and Outlook on iOS.

Support for on-device certificates and external storage

On-device certificates are provisioned on the device. Customers can use Mobile Device Management (MDM) to provision the certificates on the device. Since iOS doesn't support hardware protected keys out of the box, customers can use external storage devices for certificates.

Supported platforms

- Only native browsers are supported
- Applications using latest MSAL libraries or Microsoft Authenticator can do CBA
- Edge with profile, when users add account and logged in a profile support CBA
- Microsoft first party apps with latest MSAL libraries or Microsoft Authenticator can do CBA

Browsers

[] Expand table

Edge	Chrome	Safari	Firefox
✗	✗	✓	✗

Microsoft mobile applications support

[] Expand table

Applications	Support
Azure Information Protection app	✓
Company Portal	✓
Microsoft Teams	✓

Applications	Support
Office (mobile)	<input checked="" type="checkbox"/>
OneNote	<input checked="" type="checkbox"/>
OneDrive	<input checked="" type="checkbox"/>
Outlook	<input checked="" type="checkbox"/>
Power BI	<input checked="" type="checkbox"/>
Skype for Business	<input checked="" type="checkbox"/>
Word / Excel / PowerPoint	<input checked="" type="checkbox"/>
Yammer	<input checked="" type="checkbox"/>

Support for Exchange ActiveSync clients

On iOS 9 or later, the native iOS mail client is supported.

To determine if your email application supports Microsoft Entra CBA, contact your application developer.

Support for certificates on hardware security key

Certificates can be provisioned in external devices like hardware security keys along with a PIN to protect private key access. Microsoft's mobile certificate-based solution coupled with the hardware security keys is a simple, convenient, FIPS (Federal Information Processing Standards) certified phishing-resistant MFA method.

As for iOS 16/iPadOS 16.1, Apple devices provide native driver support for USB-C or Lightning connected CCID-compliant smart cards. This means Apple devices on iOS 16/iPadOS 16.1 see a USB-C or Lightning connected CCID-compliant device as a smart card without the use of additional drivers or third-party apps. Microsoft Entra CBA works on these USB-A, USB-C, or Lightning connected CCID-compliant smart cards.

Advantages of certificates on hardware security key

Security keys with certificates:

- Can be used on any device, and don't need a certificate to be provisioned on every device the user has
- Are hardware-secured with a PIN, which makes them phishing-resistant
- Provide multifactor authentication with a PIN as second factor to access the private key of the certificate
- Satisfy the industry requirement to have MFA on separate device
- Help in future proofing where multiple credentials can be stored including Fast Identity Online 2 (FIDO2) keys

Microsoft Entra CBA on iOS mobile with YubiKey

Even though the native Smartcard/CCID driver is available on iOS/iPadOS for Lightning connected CCID-compliant smart cards, the YubiKey 5Ci Lightning connector isn't seen as a connected smart card on these devices without the use of PIV (Personal Identity Verification) middleware like the Yubico Authenticator.

One-time registration prerequisite

- Have a PIV-enabled YubiKey with a smartcard certificate provisioned on it
- Download the [Yubico Authenticator for iOS app](#) on your iPhone with v14.2 or later
- Open the app, insert the YubiKey or tap over near field communication (NFC) and follow steps to upload the certificate to iOS keychain

Steps to test YubiKey on Microsoft apps on iOS mobile

1. Install the latest Microsoft Authenticator app.
2. Open Outlook and plug in your YubiKey.
3. Select **Add account** and enter your user principal name (UPN).
4. Select **Continue** and the iOS certificate picker appears.
5. Select the public certificate copied from YubiKey that is associated with the user's account.
6. Select **YubiKey required** to open the YubiKey authenticator app.
7. Enter the PIN to access YubiKey and select the back button at the top left corner.

The user should be successfully logged in and redirected to the Outlook homepage.

Troubleshoot certificates on hardware security key

What happens if the user has certificates both on the iOS device and YubiKey?

The iOS certificate picker shows all the certificates on both iOS device and the ones copied from YubiKey into iOS device. Depending on the certificate user picks, they may be taken to YubiKey authenticator to enter a PIN, or directly authenticated.

My YubiKey is locked after incorrectly typing PIN 3 times. How do I fix it?

- Users should see a dialog informing you that too many PIN attempts have been made. This dialog also pops up during subsequent attempts to select **Use Certificate or smart card**.
- [YubiKey Manager](#) can reset a YubiKey's PIN.

After CBA fails, the CBA option in the 'Other ways to sign in' link also fails. Is there a workaround?

This issue happens because of certificate caching. We're working on an update to clear the cache. As a workaround, select **Cancel**, retry sign-in, and choose a new certificate.

Microsoft Entra CBA with YubiKey is failing. What information would help debug the issue?

1. Open Microsoft Authenticator app, select the three dots icon in the top right corner and select **Send Feedback**.
2. Select **Having Trouble?**.
3. For **Select an option**, select **Add or sign into an account**.
4. Describe any details you want to add.
5. Select the send arrow in the top right corner. Note the code provided in the dialog that appears.

How can I enforce phishing-resistant MFA using a hardware security key on browser-based applications on mobile?

Certificate-based authentication and Conditional Access authentication strength capability makes it powerful for customers to enforce authentication needs. Edge as a profile (add an account) works with a hardware security key like YubiKey and a Conditional Access policy with authentication strength capability can enforce phishing-resistant authentication with CBA.

CBA support for YubiKey is available in the latest Microsoft Authentication Library (MSAL) libraries, and any third-party application that integrates the latest MSAL. All Microsoft first-party applications can use CBA and Conditional Access authentication strength.

Supported operating systems

[\[+\] Expand table](#)

Operating system	Certificate on-device/Derived PIV	Smart cards/Security keys
iOS	✓	Supported vendors only

Supported browsers

[\[+\] Expand table](#)

Operating system	Chrome certificate on-device	Chrome smart card/security key	Safari certificate on-device	Safari smart card/security key	Edge certificate on-device	Edge smart card/security key
iOS	✗	✗	✓	✓	✗	✗

Security key providers

[\[+\] Expand table](#)

Provider	iOS
YubiKey	✓

Known issues

- On iOS, users with certificate-based authentication will see a "double prompt", where they must select the option to use certificate-based authentication twice.
- On iOS, users with Microsoft Authenticator App will also see hourly login prompt to authenticate with CBA if there's an Authentication Strength policy enforcing CBA, or if they use CBA as the second factor.
- On iOS, an auth strength policy requiring CBA and an MAM app protection policy will end up in a loop between device registration and MFA satisfaction. Due to the

bug on iOS, when a user uses CBA to satisfy MFA requirement, the MAM policy is not satisfied with error being thrown by server saying device registration is required, even though the device is registered. This incorrect error causes re-registration and the request is stuck in loop of using CBA to sign in and device need registration. Due to the above issues, CBA as a second factor is blocked on iOS and will be unblocked as soon as the fixes are fixed.

Next steps

- Overview of Microsoft Entra CBA
- Technical deep dive for Microsoft Entra CBA
- How to configure Microsoft Entra CBA
- Microsoft Entra CBA on Android devices
- Windows smart card logon using Microsoft Entra CBA
- Certificate user IDs
- How to migrate federated users
- FAQ

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Microsoft Entra certificate-based authentication on Android devices

Article • 03/04/2025

Microsoft Entra Certificate-based authentication is supported with certificates provisioned on the device and with external security keys like YubiKeys.

Prerequisites

- Android version must be Android 5.0 (Lollipop) or later.
- Microsoft first-party apps with latest MSAL libraries or Microsoft Authenticator can do CBA.
- Third party applications using latest MSAL libraries or integrated with Microsoft Authenticator can do CBA.

CBA with on-device certificates

Customers can use their choice of Mobile Device Management (MDM) to provision the certificates on the device. End users must first register their devices with MDM and get the certificate provisioned on the device. Once the certificate is provisioned on the device, users can authenticate using CBA.

Steps to test YubiKey on Microsoft apps on Android:

1. Open Outlook.
2. Select **Add account** and enter your user principal name (UPN).
3. Click **Continue**.
4. Select **Use Certificate or smart card**.
5. Select **Certificate on the device** in the dialog**.**
6. The certificate picker appears.
7. Select the certificate associated with the user's account. Click **Continue**.
8. User is allowed to access the Outlook resource if the authentication is successful.

CBA with certificates on hardware security key

Certificates can be provisioned in external devices like hardware security keys along with a PIN to protect private key access. Microsoft Entra ID supports CBA with YubiKey.

Advantages of certificates on hardware security key

Security keys with certificates:

- Have the roaming nature of a security key, which allows users to use the same certificate on different devices.
- Are hardware-secured with a PIN, which makes them phishing-resistant.
- Provide multifactor authentication with a PIN as second factor to access the private key of the certificate.
- Satisfy the industry requirement to have MFA on separate device.
- Help in future proofing where multiple credentials can be stored including Fast Identity Online 2 (FIDO2) keys.

Microsoft Entra CBA on Android mobile with YubiKey

Android needs a middleware application to be able to support smartcard or security keys with certificates. To support YubiKeys with Microsoft Entra CBA, YubiKey Android SDK has been integrated into the Microsoft broker code which can be leveraged through the latest Microsoft Authentication Library (MSAL).

Because Microsoft Entra CBA with YubiKey on Android mobile is enabled by using the latest MSAL, YubiKey Authenticator app isn't required for Android support.

Steps to test YubiKey on Microsoft apps on Android:

1. Install Microsoft Authenticator.
2. If your YubiKey has USB-C, open Outlook and plug in your YubiKey.
3. Select **Add account** and enter your user principal name (UPN).
4. Click **Continue**, and when asked for permission to access your YubiKey, click **OK**.
5. Select **Use Certificate or smart card**.
6. If you're using an NFC-enabled Yubikey, hold the Yubikey to the back of the device.
7. A custom certificate picker appears.
8. Select the certificate associated with the user's account, and click **Continue**.
9. Enter the PIN to access YubiKey and select **Unlock**.
10. If you're using a Yubikey with NFC, hold the Yubikey to the back of the phone again to validate the PIN.
11. After authentication succeeds, you can access Outlook.

Note

For a smooth CBA flow, plug in YubiKey as soon as the application is opened and accept the consent dialog from YubiKey before selecting the link **Use Certificate or smart card**. If you want to experience only a single connection, consider having users plug in the YubiKey by using USB instead of NFC, which only needs to be done once at the beginning of login.

Support for Exchange ActiveSync clients

Certain Exchange ActiveSync applications on Android 5.0 (Lollipop) or later are supported. To determine if your email application supports Microsoft Entra CBA, contact your application developer.

Supported Microsoft Entra use cases

Microsoft mobile application support

[+] Expand table

Applications	Support
Azure Information Protection app	✓
Company Portal	✓
Microsoft Teams	✓
Office (mobile)	✓
OneNote	✓
OneDrive	✓
Outlook	✓
Power BI	✓
Skype for Business	✓
Word / Excel / PowerPoint	✓
Yammer	✓
Edge browser with profile login	✓
Managed Home Screen	✓

Browsers

[+] Expand table

Operating system	Chrome certificate on-device	Chrome smart card/security key	Safari certificate on-device	Safari smart card/security key	Edge certificate on-device	Edge smart card/security key
Android	✓	✗	N/A	N/A	✓	✗

① Note

Although Edge as a browser isn't supported, Edge as a profile (for account login) is an MSAL app that supports CBA on Android.

Operating systems

[+] Expand table

Operating system	Certificate on-device/Derived PIV	Smart cards/Security keys
Android	✓	Supported vendors only

Security key providers

[+] Expand table

Provider	Android
YubiKey	✓

Troubleshoot certificates on hardware security key

What will happen if the user has certificates both on the Android device and YubiKey?

- If the user has certificates both on the android device and YubiKey, then if the YubiKey is plugged in before user clicks Use Certificate or smart card, the user will be shown the certificates in the YubiKey.

- If the YubiKey isn't plugged in before user clicks **Use Certificate or smart card**, the user will be asked to select between certificates on device or physical smart card. If the user chooses **Certificate on device**, the user will be shown the certificates on the device. If the user chooses **Certificates on physical smart card**, plug in or hold the YubiKey to the back, and the user will be shown the certificates in the YubiKey.

My YubiKey is locked after incorrectly typing PIN three times. How do I fix it?

- Users should see a dialog informing you that too many PIN attempts have been made. This dialog also pops up during subsequent attempts to select **Use Certificate or smart card**.
- Users should contact the admin to reset a YubiKey PIN.

I have installed Microsoft authenticator but still don't see an option to do Certificate based authentication with YubiKey.

Before installing Microsoft Authenticator, uninstall Company Portal and install it after Microsoft Authenticator installation.

Does Microsoft Entra CBA support YubiKey via NFC?

Microsoft Entra CBA supports using YubiKey with USB and NFC.

Once CBA fails, clicking on the CBA option again in the 'Other ways to sign in' link on the error page fails.

This issue happens because of certificate caching. As a workaround, clicking cancel and restarting the login flow will let the user choose a new certificate and successfully login.

Microsoft Entra CBA with YubiKey is failing. What information would help debug the issue?

1. Open Microsoft Authenticator app, click the three dots icon in the top right corner and select **Send Feedback**.
2. Click **Having Trouble?**
3. For **Select an option**, select **Add or sign into an account**.
4. Describe any details you want to add.
5. Click the send arrow in the top right corner. Note the code provided in the dialog that appears.

Next steps

- Overview of Microsoft Entra CBA
 - Technical deep dive for Microsoft Entra CBA
 - How to configure Microsoft Entra CBA
 - Microsoft Entra CBA on iOS devices
 - Windows SmartCard logon using Microsoft Entra CBA
 - Certificate user IDs
 - How to migrate federated users
 - FAQ
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Mapping to the certificateUserIds attribute in Microsoft Entra ID

Article • 04/21/2025

User objects in Microsoft Entra ID have an attribute named certificateUserIds.

- The certificateUserIds attribute is multivalued and can hold up to 10 values.
- Each value can be no more than 1024 characters.
- Each value must be unique. Once a value is present on one user account, it can't be written to any other user account in the same Microsoft Entra tenant.
- The value doesn't need to be in email ID format. The certificateUserIds attribute can store nonroutable user principal names (UPNs) like *bob@woodgrove* or *bob@local*.

! Note

Although each value must be unique in Microsoft Entra ID, you can map a single certificate to multiple accounts by implementing multiple username bindings. For more information, see [Multiple username bindings](#).

Supported patterns for certificate user IDs

The values stored in certificateUserIds should be in the format described in the following table. The X509:<Mapping> prefixes are case-sensitive.

[] Expand table

Certificate mapping Field	Examples of values in certificateUserIds
PrincipalName	X509:<PN>bob@woodgrove.com
PrincipalName	X509:<PN>bob@woodgrove
RFC822Name	X509:<RFC822>user@woodgrove.com
IssuerAndSubject	X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-CA<S>DC=com,DC=contoso,OU=UserAccounts,CN=mfatest
Subject	X509:<S>DC=com,DC=contoso,OU=UserAccounts,CN=mfatest
SKI	X509:<SKI>aB1cD2eF3gH4iJ5kL6mN7oP8qR
SHA1PublicKey	X509:<SHA1-PUKEY>cD2eF3gH4iJ5kL6mN7oP8qR9sT

Certificate mapping	Examples of values in certificateUserIds
Field	
IssuerAndSerialNumber	<p>X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-CA<SR>eF3gH4iJ5KL6mN7oP8qR9sT0uV</p> <p>To get the correct value for serial number, run this command and store the value shown in certificateUserIds:</p> <p>Syntax:</p> <pre>Certutil -dump -v [~certificate path~] >> [~dumpFile path~]</pre> <p>Example:</p> <pre>certutil -dump -v firstusercert.cer >> firstCertDump.txt</pre>

Roles to update certificateUserIds

Cloud-only users must have at least **Privileged Authentication Administrator** role to update certificateUserIds. Cloud-only users can use either the Microsoft Entra admin center or Microsoft Graph to update certificateUserIds.

Synchronized users must have at least **Hybrid Identity Administrator** role to update certificateUserIds. Only Microsoft Entra Connect can be used to update certificateUserIds by synchronizing the value from on-premises.

(!) Note

Active Directory administrators can make changes that impact the certificateUserIds value in Microsoft Entra ID for any synchronized account. Administrators can include accounts with delegated administrative privilege over synchronized user accounts, or administrative rights over the Microsoft Entra Connect servers.

How to find the correct CertificateUserIds values for a user from the end user certificate using PowerShell module

Certificate UserIds follow a certain pattern for its values as per the UserName binding configurations on the tenant. The following PowerShell command helps an admin to retrieve the exact values for Certificate UserIds attribute for a user from an end user certificate. Admin can also get the current values in Certificate UserIds attribute for a user for a given username binding and set the value of the Certificate UserIds attribute.

More information at [Microsoft Entra PowerShell Installation](#) and [Microsoft Graph PowerShell](#).

1. Start PowerShell.

2. Install and import the Microsoft Graph PowerShell SDK.

PowerShell

```
Install-Module Microsoft.Graph -Scope CurrentUser
Import-Module Microsoft.Graph.Authentication
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

3. Install Microsoft Entra PowerShell module (Minimum required version is 1.0.6)

PowerShell

```
Install-Module -Name Microsoft.Entra
```

More information on CertificateBasedAuthentication module [here](#)

Get-EntraUserCBAuthorizationInfo

Get-EntraUserCBAuthorizationInfo helps retrieve authorization information for a Microsoft Entra ID user, including certificate-based authentication identifiers.

Syntax: Get-EntraUserCBAuthorizationInfo [-UserId] <String> [-Raw] [<CommonParameters>]

Example 1: Get authorization information for a user by User Principal Name

PowerShell

```
Connect-Entra -Scopes 'User.Read.All'
Get-EntraUserCBAuthorizationInfo -UserId 'user@contoso.com'
```

Response:

[] [Expand table](#)

Attribute	Value
Id	aaaaaaaa-0000-1111-2222-bbbbbbbbbb
DisplayName	Contoso User
UserPrincipalName	user@contoso.com
UserType	Member
AuthorizationInfo	<pre>@{CertificateUserIds=System.Object[]; RawAuthorizationInfo=System.Collections.Hashtable}</pre>

This command retrieves the authorization information for the user with the specified User Principal Name.

Example 2: Retrieve authorization information for a user

PowerShell

```
Connect-Entra -Scopes 'User.Read.All'  
$userInfo = Get-EntraUserCBAuthorizationInfo -UserId 'user@contoso.com'  
$userInfo.AuthorizationInfo.CertificateUserIds | Format-Table Type, TypeName,  
Value
```

Response:

 Expand table

Type	TypeName	Value
PN	PrincipalName	user@contoso.com
S	Subject	CN=user@contoso.com
SKI	SubjectKeyIdentifier	1111112222333344445555

This example retrieves the authorization information.

Example 3: Extract specific certificate user IDs

PowerShell

```
Connect-Entra -Scopes 'User.Read.All'  
$userInfo = Get-EntraUserCBAuthorizationInfo -UserId user@contoso.com'  
$userInfo.AuthorizationInfo.CertificateUserIds | Where-Object Type -eq "PN" |  
Select-Object -ExpandProperty Value
```

Response: user@contoso.com

This example retrieves the authorization information and then filters to display only the Principal Name certificate values.

Get-EntraUserCertificateUserIdsFromCertificate

Returns an object with the certificate values needed to configure CertificateUserIDs for Certificate-Based Authentication in Microsoft Entra ID.

Syntax: Get-EntraUserCertificateUserIdsFromCertificate [-Path] <string> [[-Certificate] <System.Security.Cryptography.X509Certificates.X509Certificate2> [-CertificateMapping] <string>] [<CommonParameters>]

Example 1: Retrieve certificate object from a certificate path

PowerShell

```
Get-EntraUserCertificateUserIdsFromCertificate -Path 'C:\path\to\certificate.cer'
```

Response:

[] [Expand table](#)

Name	Value
Subject	X509:<S>DC=com,DC=contoso,OU=UserAccounts,CN=user
IssuerAndSerialNumber	X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-CA<SR>eF3gH4iJ5kL6mN7oP8qR9sV0uD
RFC822Name	X509:<RFC822>user@contoso.com
SHA1PublicKey	X509:<SHA1-PUKEY>cA2eB3gH4iJ5kL6mN7oP8qR9sT
IssuerAndSubject	X509:<I>DC=com,DC=contoso,CN=CONTOSO-DC-<S>DC=com,DC=contoso,OU=UserAccounts,CN=user
SKI	X509:<SKI>aB1cD2eF3gH4iJ5kL6mN7oP8qR
PrincipalName	X509:<PN>user@contoso.com

This example shows how to get all possible certificate mappings as an object.

Example 2: Retrieve certificate object from a certificate path and certificate mapping

PowerShell

```
Get-EntraUserCertificateUserIdsFromCertificate -Path 'C:\path\to\certificate.cer' -CertificateMapping 'Subject'
```

Response: X509:<S>DC=com,DC=contoso,OU=UserAccounts,CN=user

This command returns the PrincipalName property.

Example 3: Retrieve certificate object from a certificate

PowerShell

```
$text = "-----BEGIN CERTIFICATE-----  
MIIDiz...=  
-----END CERTIFICATE-----"  
$bytes = [System.Text.Encoding]::UTF8.GetBytes($text)  
$certificate =  
[System.Security.Cryptography.X509Certificates.X509Certificate2]::new($bytes)  
Get-EntraUserCertificateUserIdsFromCertificate -Certificate $certificate -  
CertificateMapping 'Subject'
```

Response: X509:<S>DC=com,DC=contoso,OU=UserAccounts,CN=user

This command returns the PrincipalName property.

Set-EntraUserCBAutoCertificateUserId

Sets certificate-based authentication user IDs for a user in Microsoft Entra ID using a certificate file or object.

Syntax Set-EntraUserCBAutoCertificateUserId `-UserId <string>` `[-CertPath <string>]` `[-Cert <System.Security.Cryptography.X509Certificates.X509Certificate2>]` `[-CertificateMapping <string[]>]` `[<CommonParameters>]`

Example 1: Update user's certificate authorization information using certificate path

PowerShell

```
Connect-Entra -Scopes 'Directory.ReadWrite.All', 'User.ReadWrite.All'  
Set-EntraUserCBAutoCertificateUserId -UserId 'user@contoso.com' -CertPath  
'C:\path\to\certificate.cer' -CertificateMapping @('Subject', 'PrincipalName')
```

This example sets the certificate user IDs for the specified user using a certificate file, mapping both the Subject and PrincipalName fields. You can use Get-EntraUserCBAutoAuthorizationInfo command to view updated details.

Example 2: Update user's certificate authorization information using a certificate

PowerShell

```
Connect-Entra -Scopes 'Directory.ReadWrite.All', 'User.ReadWrite.All'  
$text = "-----BEGIN CERTIFICATE-----  
MIIDiz...=  
-----END CERTIFICATE-----"  
$bytes = [System.Text.Encoding]::UTF8.GetBytes($text)  
$certificate =  
[System.Security.Cryptography.X509Certificates.X509Certificate2]::new($bytes)
```

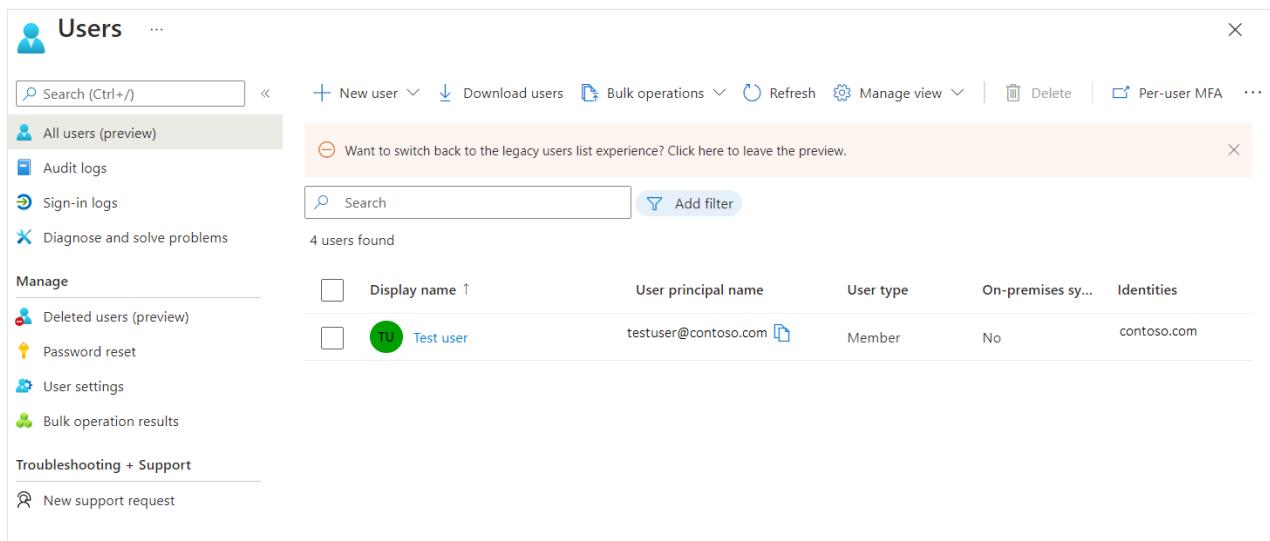
```
Set-EntraUserCBACertificateUserId -UserId user@contoso.com' -Cert $certificate -  
CertificateMapping @('RFC822Name', 'SKI')
```

This example sets the certificate user IDs for the specified user using a certificate object, mapping the RFC822Name and SKI fields. You can use Get-EntraUserCBAuthorizationInfo command to view updated details.

Update certificateUserIds using Microsoft Entra admin center

Use the following steps to update certificateUserIds for users:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Privileged Authentication Administrator** for cloud-only users or as at least a **Hybrid Identity Administrator** for synchronized users.
2. Search for and select **All users**.



The screenshot shows the 'Users' page in the Microsoft Entra admin center. The left sidebar includes links for Audit logs, Sign-in logs, Diagnose and solve problems, Deleted users (preview), Password reset, User settings, Bulk operation results, New support request, and Troubleshooting + Support. The main area displays a table with four users found. The columns are: Display name ↑, User principal name, User type, On-premises sy..., and Identities. One user, 'Test user', is highlighted with a green circular icon containing 'TU'. The user's principal name is 'testuser@contoso.com', type is 'Member', and identity is 'contoso.com'.

3. Select a user, and select **Edit Properties**.
4. Next to **Authorization info**, select **View**.

Test user ...

User

Search (Ctrl+ /) < Edit properties Delete Refresh Reset password Revoke sessions Got feedback?

Overview Audit logs Sign-in logs Diagnose and solve problems

Manage

- Custom security attributes (preview)
- Assigned roles
- Administrative units
- Groups
- Applications
- Licenses
- Devices
- Azure role assignments
- Authentication methods

Troubleshooting + Support

New support request

Identity

Display name	Test user	Contact Information
First name		Street address
Last name		City
User principal name	testuser@ 	State or province
Object ID	aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb 	ZIP or postal code
Identities	.com 	Country or region
User type	Member	Business phone
Creation type		Mobile phone
Created date time	Sep 6, 2022, 4:40 PM	Email
Last password change date time	Sep 6, 2022, 4:40 PM	Other emails
External user state		Proxy addresses
External user state change date ti...		Fax number
Assigned licenses	View	IM addresses
Password policies		Mail nickname
Password profile	View	testuser
Preferred language		Parental controls
Sign in sessions valid from date t...	Sep 6, 2022, 4:40 PM	
Authorization info	View	Age group
Job Information		Consent provided for minor
Job title		Legal age group classification
		Settings
		
		Account enabled
		Yes

Authorization info

Job Information

5. Select **Edit certificate user IDs**.

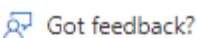
Test user

...

Properties



Refresh



Got feedback?

First name

Last name

User principal name

Object ID

User type

▼

Creation type

Created date time

Last password change date time

External user state

External user state change date time

Sign in sessions valid from date time

Authorization info

[Edit Certificate user IDs](#)

Job title

Company name

Department

Employee ID

Employee type

Employee hire date



Office location

Manager

[+ Add manager](#)

Street address

City

State or province

6. Select Add.

Edit Certificate user IDs

+ Add

Save

Cancel

7. Enter the value and select Save. You can add up to four values, each of 120 characters.

Edit Certificate user IDs

X509:<PN>testuser@contoso.com



+ Add

Save

Cancel

Update certificateUserIds using Microsoft Graph queries

The following examples show how to use Microsoft Graph to look up certificateUserIds and update them.

Look up certificateUserIds

Authorized callers can run Microsoft Graph queries to find all the users with a given certificateUserId value. On the Microsoft Graph [user](#) object, the collection of certificateUserIds is stored in the **authorizationInfo** property.

To retrieve certificateUserIds of all user objects:

```
msgraph
```

```
GET https://graph.microsoft.com/v1.0/users?$select=authorizationinfo  
ConsistencyLevel: eventual
```

To retrieve certificateUserIds for a given user by user's ObjectId:

```
msgraph
```

```
GET https://graph.microsoft.com/v1.0/users/{user-object-id}?  
$select=authorizationinfo
```

```
ConsistencyLevel: eventual
```

To retrieve the user object with a specific value in certificateUserIds:

```
msgraph
```

```
GET https://graph.microsoft.com/v1.0/users?  
$select=authorizationinfo&$filter=authorizationInfo/certificateUserIds/any(x:x eq  
'X509:<PN>user@contoso.com')&$count=true  
ConsistencyLevel: eventual
```

You can also use the `not` and `startsWith` operators to match the filter condition. To filter against the certificateUserIds object, the request must include the `$count=true` query string, and the `ConsistencyLevel` header must be set to `eventual`.

Update certificateUserIds

Run a PATCH request to update the certificateUserIds for a given user.

Request body

```
HTTP
```

```
PATCH https://graph.microsoft.com/v1.0/users/{user-object-id}  
Content-Type: application/json  
{  
    "authorizationInfo": {  
        "certificateUserIds": [  
            "X509:<PN>123456789098765@mil"  
        ]  
    }  
}
```

Update certificateUserIds using Microsoft Graph PowerShell commands

For this configuration, you can use [Microsoft Graph PowerShell](#).

1. Start PowerShell with administrator privileges.
2. Install and import the Microsoft Graph PowerShell SDK.

```
PowerShell
```

```
Install-Module Microsoft.Graph -Scope CurrentUser
Import-Module Microsoft.Graph.Authentication
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

3. Connect to the tenant and accept all.

PowerShell

```
Connect-MGGraph -Scopes "Directory.ReadWrite.All", "User.ReadWrite.All" -
TenantId <tenantId>
```

4. List certificateUserIds attribute of a given user.

PowerShell

```
$results = Invoke-MGGraphRequest -Method get -Uri
'https://graph.microsoft.com/v1.0/users/<userId>?$select=authorizationinfo' -
OutputType PSObject -Headers @{'ConsistencyLevel' = 'eventual'}
#list certificateUserIds
$results.authorizationInfo
```

5. Create a variable with certificateUserIds values.

PowerShell

```
#Create a new variable to prepare the change. Ensure that you list any
existing values you want to keep as this operation will overwrite the
existing value
$params = @{
    authorizationInfo = @{
        certificateUserIds = @(
            "X509:<SKI>gH4iJ5kL6mN7oP8qR9sT0uV1wX",
            "X509:<PN>user@contoso.com"
        )
    }
}
```

6. Update the certificateUserIds attribute.

PowerShell

```
$results = Invoke-MGGraphRequest -Method patch -Uri
'https://graph.microsoft.com/v1.0/users/<UserId>/?$select=authorizationinfo' -
OutputType PSObject -Headers @{'ConsistencyLevel' = 'eventual'} -Body
$params
```

Update certificateUserIds using user object

1. Get the user object.

```
PowerShell
```

```
$userObjectId = "aaaaaaaa-0000-1111-2222-bbbbbbbbbbb"  
$user = Get-MgUser -UserId $userObjectId -Property AuthorizationInfo
```

2. Update the certificateUserIds attribute of the user object.

```
PowerShell
```

```
$user.AuthorizationInfo.certificateUserIds = @("X509:  
<SKI>iJ5kL6mN7oP8qR9sT0uV1wX2yZ", "X509:<PN>user1@contoso.com")  
Update-MgUser -UserId $userObjectId -AuthorizationInfo  
$user.AuthorizationInfo
```

Update certificateUserIds using Microsoft Entra Connect

Microsoft Entra Connect supports synchronizing values to certificateUserIds from an on-premises Active Directory environment. On-premises Active Directory supports certificate-based authentication and multiple username bindings. Make sure you use the latest version of [Microsoft Entra Connect](#).

To use these mapping methods, you need to populate the altSecurityIdentities attribute of user objects in the on-premises Active Directory. In addition, after you apply certificate-based authentication changes on Windows domain controllers as described in [KB5014754](#), you may have implemented some of the nonreusable mapping methods (Type=strong) mapping methods to meet the on-premises Active Directory strong certificate binding enforcement requirements.

To prevent synchronization errors, make sure the values being synchronized follow one of the supported formats for the certificateUserIds.

Before you begin, make sure all user accounts that are synchronized from on-premises Active Directory have:

- 10 or fewer values in their altSecurityIdentities attributes
- No value with more than 1,024 characters
- No duplicate values

Carefully consider if a duplicate value is meant to map a single certificate to multiple on-premises Active Directory accounts. For more information, see [Multiple username bindings](#).

! Note

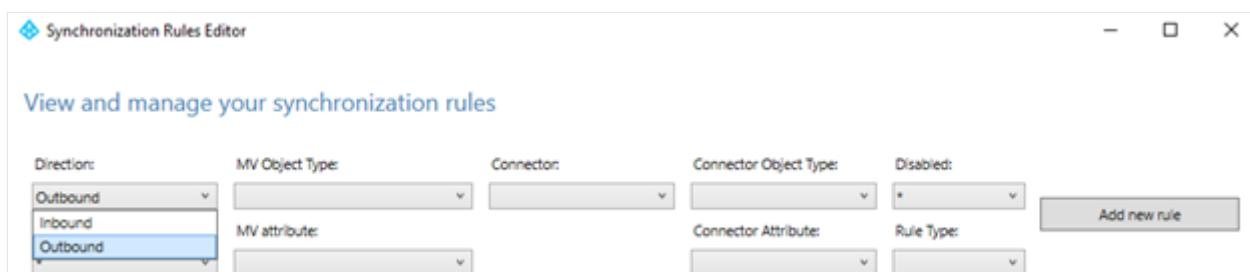
In specific scenarios, a subset of users might have a valid business justification to map a single certificate to more than one on-premises Active Directory account. Review these scenarios and where needed, implement separate mapping methods to map to more than one account in both the on-premises Active Directory and Microsoft Entra ID.

Considerations for ongoing synchronization of certificateUserIds

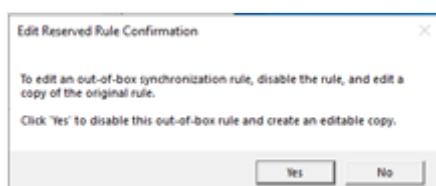
- Ensure that the provisioning process for populating the values in on-premises Active Directory implements proper hygiene. Only values associated with current valid certificates are populated.
- Values are removed when the corresponding certificate is expired or revoked.
- Values larger than 1024 characters aren't populated.
- Duplicate values aren't provisioned.
- Use Microsoft Entra Connect Health to monitor synchronization.

Follow these steps to configure Microsoft Entra Connect to synchronize userPrincipalName to certificateUserIds:

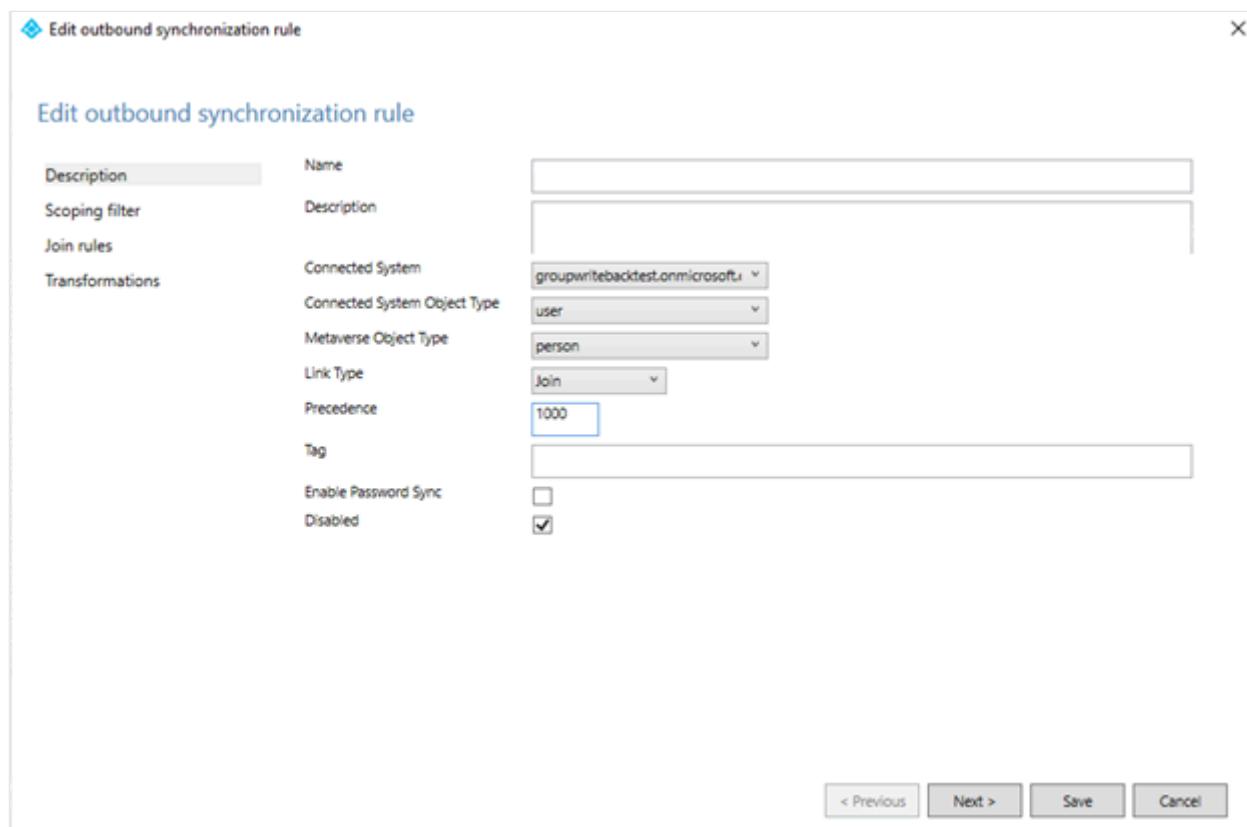
1. On the Microsoft Entra Connect server, find and start the **Synchronization Rules Editor**.
2. Select **Direction**, and select **Outbound**.



3. Find the rule **Out to Microsoft Entra ID – User Identity**, select **Edit**, and select **Yes** to confirm.



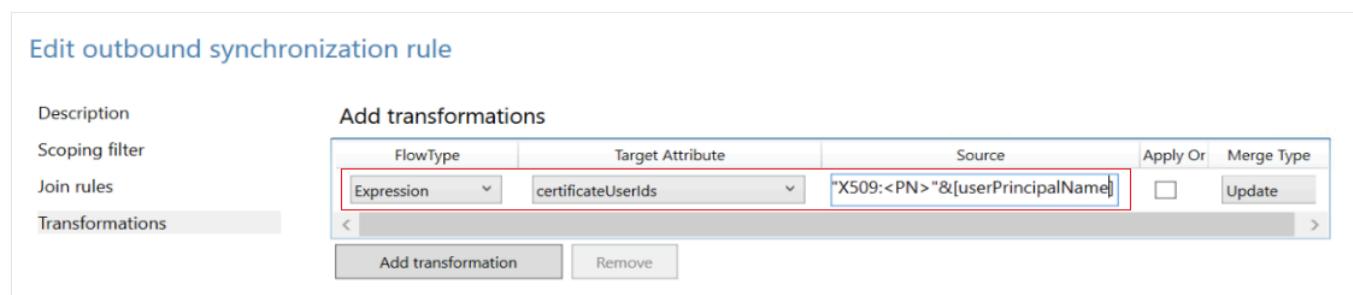
4. Enter a high number in the **Precedence** field, and then select **Next**.



5. Select **Transformations > Add transformation**. You may need to scroll down the list of transformations before you can create a new one.

Synchronize X509:<PN>PrincipalNameValue

To synchronize X509:<PN>PrincipalNameValue, create an outbound synchronization rule, and choose **Expression** in the flow type. Choose the target attribute as **certificateUserIds**, and in the source field, add the following expression. If your source attribute isn't **userPrincipalName**, you can change the expression accordingly.



Synchronize X509:<RFC822>RFC822Name

To synchronize X509:<RFC822>RFC822Name, create an outbound synchronization rule and choose **Expression** in the flow type. Choose the target attribute as **certificateUserIds**, and in the source field, add the following expression. If your source attribute isn't userPrincipalName, you can change the expression accordingly.

The screenshot shows the 'View outbound synchronization rule' interface. On the left, there are tabs for 'Description', 'Scoping filter', 'Join rules', and 'Transformations'. The 'Transformations' tab is selected. In the center, there is a table titled 'Add transformations' with columns: FlowType, Target Attribute, Source, Apply Or, and Merge Type. A single row is present in the table:

FlowType	Target Attribute	Source	Apply Or	Merge Type
Expression	certificateUserIds	"X509:<RFC822>"&[userPrincipalName]	<input type="checkbox"/>	Update

Below the table are 'Add transformation' and 'Remove' buttons.

1. Select **Target Attribute**, select **certificateUserIds**, select **Source**, select **userPrincipalName**, and then select **Save**.

The screenshot shows the 'Edit outbound synchronization rule' dialog. On the left, there are tabs for 'Description', 'Scoping filter', 'Join rules', and 'Transformations'. The 'Transformations' tab is selected. In the center, there is a table titled 'Add transformations' with columns: FlowType, Target Attribute, Source, Apply Or, and Merge Type. Multiple rows are listed in the table:

FlowType	Target Attribute	Source	Apply Or	Merge Type
Direct	displayName	displayName	<input type="checkbox"/>	Update
Direct	employeeID	employeeID	<input type="checkbox"/>	Update
Direct	givenName	givenName	<input type="checkbox"/>	Update
Direct	mail	mail	<input type="checkbox"/>	Update
Direct	onPremisesUserPrincipalName	onPremisesUserPrincipalName	<input type="checkbox"/>	Update
Direct	proxyAddresses	proxyAddresses	<input type="checkbox"/>	Update
Direct	surname	sn	<input type="checkbox"/>	Update
Direct	userPrincipalName	userPrincipalName	<input type="checkbox"/>	Update
Direct	certificateUserIds	userPrincipalName	<input type="checkbox"/>	Update

Below the table are 'Add transformation' and 'Remove' buttons. At the bottom right, there are buttons for '< Previous', 'Next >', 'Save', and 'Cancel'.

2. Select **OK** to confirm.

Important

The preceding examples use **userPrincipalName** attribute as a source attribute in the transform rule. You can use any available attribute with the appropriate value. For

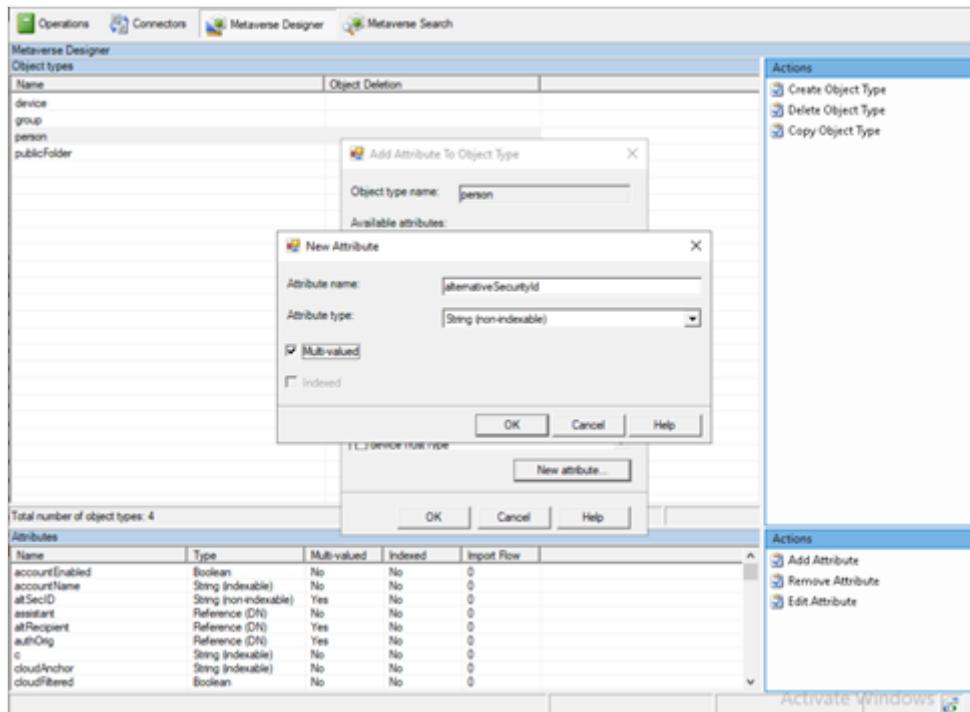
example, some organizations use the mail attribute. For more complex transform rules, see [Microsoft Entra Connect Sync: Understanding Declarative Provisioning Expressions](#)

For more information about declarative provisioning expressions, see [Microsoft Entra Connect: Declarative Provisioning Expressions](#).

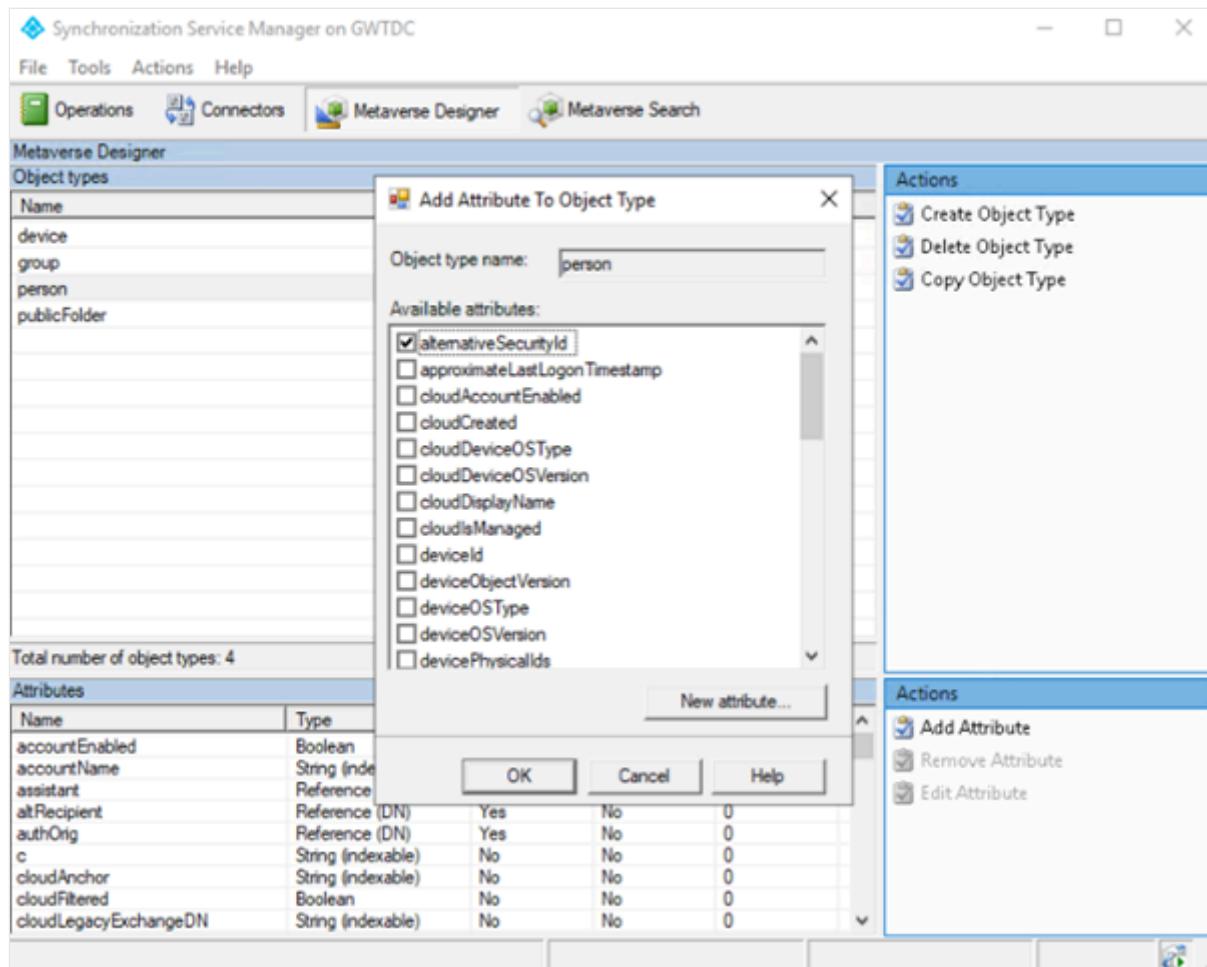
Synchronize altSecurityIdentities attribute from Active Directory to Microsoft Entra certificateUserIds

The altSecurityIdentities attribute isn't part of the default attributes set. An administrator needs to add a new attribute to the person object in the Metaverse, and then create the appropriate synchronization rules to relay this data to certificateUserIds in Microsoft Entra ID.

1. Open Metaverse Designer and select the person object. To create the alternativeSecurityId attribute, select **New attribute**. Select **String (non-indexable)** to create an attribute size up to 1024 characters, which is the maximum supported length for certificateUserIds. If you select **String (indexable)**, the maximum size of an attribute value is 448 characters. Make sure you select **Multi-valued**.



2. Open Metaverse Designer, and select alternativeSecurityId to add it to the person object.



3. Create an inbound synchronization rule to transform from altSecurityIdentities to alternativeSecurityId attribute.

In the inbound rule, use the following options.

[Expand table](#)

Option	Value
Name	Descriptive name of the rule, such as: In from Active Directory - altSecurityIdentities
Connected System	Your on-premises Active Directory domain
Connected System Object Type	user
Metaverse Object Type	person
Precedence	Choose a number under 100 that isn't currently used

Then select **Transformations** and create a direct mapping to the target attribute alternativeSecurityId from the source attribute altSecurityIdentities, as shown in the following screenshot.

View inbound synchronization rule

FlowType	Target Attribute	Source	Apply Or	Merge Type
Direct	alternativeSecurityId	altSecurityIdentities	<input type="checkbox"/>	Update

Add transformation Remove

4. Create an outbound synchronization rule to transform from the alternativeSecurityId attribute to the certificateUserIds attribute in Microsoft Entra ID.

[Expand table](#)

Option	Value
Name	Descriptive name of the rule, such as: Out to Microsoft Entra ID - certificateUserIds
Connected System	Your Microsoft Entra domain
Connected System Object Type	user
Metaverse Object Type	person
Precedence	Choose a high number not currently used above all default rules, such as 150

Then select **Transformations** and create a direct mapping to the target attribute certificateUserIds from the source attribute alternativeSecurityId, as shown in the following screenshot.

FlowType	Target Attribute	Source	Apply Or	Merge Type
Direct	certificateUserIds	alternativeSecurityId	<input type="checkbox"/>	Update

Add transformation Remove

5. Run the synchronization to populate data to the certificateUserIds attribute.
6. To verify success, view the Authorization info of a user in Microsoft Entra ID.

To map a subset of values from the altSecurityIdentities attribute, replace the Transformation in step 4 with an Expression. To use an Expression, proceed to the **Transformations** tab and change your FlowType option to Expression, the target attribute to certificateUserIds, and then input the expression into the Source field. The following example filters only values that align to the SKI and SHA1PublicKey Certificate mapping fields:

Expression code:

```
PowerShell
```

```
IIF(IsPresent([alternativeSecurityId]),  
    Where($item,[alternativeSecurityId],BitOr(InStr($item, "X509:  
<SKI>"),InStr($item, "X509:<SHA1-PUKEY>"))>0),[alternativeSecurityId]  
)
```

Administrators can filter values from altSecurityIdentities that align with the supported patterns. Ensure that the CBA configuration is updated to support the username bindings that are synchronized to certificateUserIds and enable authentication using these values.

Next steps

- [Overview of Microsoft Entra CBA](#)
- [Technical deep dive for Microsoft Entra CBA](#)
- [How to configure Microsoft Entra CBA](#)
- [Microsoft Entra CBA on iOS devices](#)
- [Microsoft Entra CBA on Android devices](#)
- [Windows smart card logon using Microsoft Entra CBA](#)
- [How to migrate federated users](#)
- [FAQ](#)

Migrate from federation to Microsoft Entra certificate-based authentication (CBA)

Article • 02/28/2025

This article explains how to migrate from running federated servers such as Active Directory Federation Services (AD FS) on-premises to cloud authentication using Microsoft Entra certificate-based authentication (CBA).

Staged Rollout

A tenant admin could cut the federated domain fully over to Microsoft Entra CBA without pilot testing. This is done by enabling the CBA auth method in Microsoft Entra ID and converting the entire domain to managed authentication. However, if customer wants to test a small batch of users authenticate against Microsoft Entra CBA before the full domain cutover to managed, they can make use of staged rollout feature.

[Staged Rollout](#) for Certificate-based Authentication (CBA) helps customers transition from performing CBA at a federated IdP to Microsoft Entra ID by selectively moving small set of users to use CBA at Microsoft Entra ID (no longer being redirected to the federated IdP) with selected groups of users before then converting the domain configuration in Microsoft Entra ID from federated to managed. Staged rollout isn't designed for the domain to remain federated for long periods of time or for large amounts of users.

Watch this quick video demonstrating the migration from ADFS certificate-based authentication to Microsoft Entra CBA

<https://www.youtube-nocookie.com/embed/jsKQxo-xGgA>

Note

When Staged rollout is enabled for a user, the user is considered a managed user and all authentication happens at Microsoft Entra ID. For a federated Tenant, if CBA is enabled on Staged Rollout, password authentication only works if PHS is enabled too. Otherwise, password authentication fails.

Enable Staged Rollout for certificate-based authentication on your tenant

To configure Staged Rollout, follow these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an **User Administrator**.
2. Search for and select **Microsoft Entra Connect**.
3. On the Microsoft Entra Connect page, under the Staged Rollout of cloud authentication, select **Enable Staged Rollout for managed user sign-in**.
4. On the **Enable Staged Rollout** feature page, select **On** for the option [Certificate-based authentication](#)
5. Select **Manage groups** and add groups you want to be part of cloud authentication. To avoid a time-out, ensure that the security groups contain no more than 200 members initially.

Note

Microsoft recommends using separate groups to manage staged rollout for Entra certificate-based authentication and the certificate-based authentication method policy

For more information, see [Staged Rollout](#).

Use Microsoft Entra Connect to update certificateUserIds attribute

An AD FS admin can use **Synchronization Rules Editor** to create rules to sync the values of attributes from AD FS to Microsoft Entra user objects. For more information, see [Sync rules for certificateUserIds](#).

Microsoft Entra Connect requires a special role named **Hybrid Identity Administrator**, which grants the necessary permissions. You need this role for permission to write to the new cloud attribute.

Note

If a user is using synchronized attributes, such as the `onPremisesUserPrincipalName` attribute in the user object for username binding, then any user that has administrative access to the Microsoft Entra Connect server can change the synchronized attribute mapping, and change the value of the synchronized

attribute. The user doesn't need to be a cloud admin. The AD FS admin should make sure the administrative access to the Microsoft Entra Connect server should be limited, and privileged accounts should be cloud-only accounts.

Frequently asked questions about migrating from AD FS to Microsoft Entra ID

Can we have privileged accounts with a federated AD FS server?

Although it's possible, Microsoft recommends privileged accounts be cloud-only accounts. Using cloud-only accounts for privileged access limits exposure in Microsoft Entra ID from a compromised on-premises environment. For more information, see [Protecting Microsoft 365 from on-premises attacks](#).

If an organization is a hybrid running both AD FS and Azure CBA, are they still vulnerable to the AD FS compromise?

Microsoft recommends privileged accounts be cloud-only accounts. This practice limits the exposure in Microsoft Entra ID from a compromised on-premises environment. Maintaining privileged accounts a cloud-only is foundational to this goal.

For synchronized accounts:

- If they're in a managed domain (not federated), there's no risk from the federated IdP.
- If they're in a federated domain, but a subset of accounts is being moved to Microsoft Entra CBA by Staged Rollout, they're subject to risks related to the federated IdP until the federated domain is fully switched to cloud authentication.

Should organizations eliminate federated servers like AD FS to prevent the capability to pivot from AD FS to Azure?

With federation, an attacker could impersonate anyone, such as a CIO, even if they can't obtain a cloud-only role like a highly privileged administrator account.

When a domain is federated in Microsoft Entra ID, a high level of trust is being placed on the Federated IdP. AD FS is one example, but the notion holds true for *any* federated

IdP. Many organizations deploy a federated IdP such as AD FS exclusively to accomplish certificate based authentication. Microsoft Entra CBA completely removes the AD FS dependency in this case. With Microsoft Entra CBA, customers can move their application estate to Microsoft Entra ID to modernize their IAM infrastructure and reduce costs with increased security.

From a security perspective, there's no change to the credential, including the X.509 certificate, CACs, PIVs, and so on, or to the PKI being used. The PKI owners retain complete control of the certificate issuance and revocation lifecycle and policy. The revocation check and the authentication happen at Microsoft Entra ID instead of federated Idp. These checks enable passwordless, phishing-resistant authentication directly to Microsoft Entra ID for all users.

How does authentication work with Federated AD FS and Microsoft Entra cloud authentication with Windows?

Microsoft Entra CBA requires the user or application to supply the Microsoft Entra UPN of the user who signs in.

In the browser example, the user most often types in their Microsoft Entra UPN. The Microsoft Entra UPN is used for realm and user discovery. The certificate used then must match this user by using one of the configured username bindings in the policy.

In Windows sign-in, the match depends on if the device is hybrid or Microsoft Entra joined. But in both cases, if username hint is provided, Windows sends the hint as a Microsoft Entra UPN. The certificate used then must match this user by using one of the configured username bindings in the policy.

Next steps

- [Overview of Microsoft Entra CBA](#)
- [Technical deep dive for Microsoft Entra CBA](#)
- [How to configure Microsoft Entra CBA](#)
- [Microsoft Entra CBA on iOS devices](#)
- [Microsoft Entra CBA on Android devices](#)
- [Windows smart card sign-in using Microsoft Entra CBA](#)
- [Certificate user IDs](#)
- [FAQ](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Frequently asked questions about Microsoft Entra certificate-based authentication (CBA)

FAQ

This article addresses frequently asked questions about how Microsoft Entra certificate-based authentication (CBA) works. Keep checking back for updated content.

Why don't I see an option to sign in to Microsoft Entra ID by using certificates after I enter my username?

An administrator needs to enable CBA for the tenant to make the sign-in with certificate option available for users. For more information, see [Step 3: Configure authentication binding policy](#).

Where can I get more diagnostic information after a user sign-in failed?

On the error page, select **More Details** for more information to help your tenant admin. The tenant admin can check the **sign-in logs** to investigate further. For example, if a user certificate is revoked and is part of a Certificate Revocation List, then authentication fails correctly. To get more diagnostic information, check the **sign-in logs**.

How can an administrator enable Microsoft Entra CBA?

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Policy Administrator**.
2. Browse to **Entra ID > Authentication methods > Policies**.
3. Select the **Certificate-based Authentication** policy.
4. On the **Enable and Target** tab, select **Enable**.

Is Microsoft Entra CBA a free feature?

Certificate-based authentication is a free feature. Every edition of Microsoft Entra ID includes Microsoft Entra CBA. For more information about features in each Microsoft Entra edition, see [Microsoft Entra pricing](#).

Does Microsoft Entra CBA support Alternate ID as the username instead of userPrincipalName?

No, sign-in using a non-UPN value, such as an alternate email, isn't supported now.

Can I have more than one CRL Distribution Point (CDP) for a Certificate Authority (CA)?

No, only one CDP is supported per CA.

Can I have non-http URLs for CDP?

No, CDP supports only HTTP URLs.

How do I find the CRL for a Certificate Authority or how do I troubleshoot the error AADSTS2205015: The Certificate Revocation List (CRL) failed signature validation?

Download the CRL and compare the CA certificate and the CRL information to validate the crlDistributionPoint value is valid for the CA you want to add. You can configure the CRL to the corresponding CA by matching CA's Issuer SKI to the AKI of the CRL (CA Issuer SKI == CRL AKI). The following table and graphic show how to map information from the CA certificate to the attributes of the downloaded CRL.

[] Expand table

CA Certificate Info	=	Downloaded CRL Info
Subject	=	Issuer
Subject Key Identifier	=	Authority Key Identifier (KeyID)

The screenshot displays two windows side-by-side. On the left is a 'Certificate' window titled 'General' showing certificate details. On the right is a 'Certificate Revocation List' window titled 'General'. Red arrows point from specific fields in the left window to corresponding fields in the right window, illustrating how they map.

Field	Value
Subject	Your CA Here
Public key	RSA (2048 Bits)
Public key parameters	05 00
Key Usage	Digital Signature, Certificate Sig...
Subject Key Identifier	ABC123def456ghi789Jkl012...
CA Version	V0.0
Basic Constraints	Subject Type=CA, Path Length ...
Thumbprint	dd431d87b5490d2832c66401...

Field	Value
Version	V2
Issuer	Your CA Here
Effective date	Tuesday, January 4, 2022 8:17:3...
Next update	Saturday, June 4, 2022 8:37:33 PM
Signature algorithm	sha256RSA
Signature hash alg...	sha256
Authority Key Iden...	KeyID=ABC123def456ghi789Jkl012...
CA Version	V0.0
CRL Number	0a

Red boxes highlight the 'Subject' field in both windows, and red arrows point from the 'Subject Key Identifier' field in the CA Certificate Info window to the 'Authority Key Identifier' field in the Downloaded CRL Info window, and from the 'Subject Key Identifier' value in the CA Certificate Info window to the 'KeyID' value in the Downloaded CRL Info window.

How do I validate the Certificate Authority configuration?

It's important to ensure that the Certificate Authority configuration in the trust store results in Microsoft Entra's ability to both validate the certificate authority trust chain. Additionally, it should successfully acquire the certificate revocation list (CRL) from the configured certificate authority CRL distribution point (CDP). To assist with this task, it's recommended to install the [MSIdentity Tools](#) PowerShell module and run `Test-MsIdCBATrustStoreConfiguration`. This PowerShell cmdlet will review the Microsoft Entra tenant certificate authority configuration and surface errors/warnings for common mis-configuration issues.

How can I turn certificate revocation checking on or off for a particular CA?

We highly recommend against disabling certificate revocation list (CRL) checking as you won't be able to revoke certificates. However, if you need to investigate issues with CRL checking, you can exempt a CA from CRL checking in the Microsoft Entra admin center. In the CBA Authentication methods policy, click **Configure** and then click **Add exemption**. Choose the CA that you want to exempt, and click **Add**.

Is there a limit for CRL size?

The following CRL size limits apply:

- Interactive sign in download limit: 20 MB (Azure Global includes GCC), 45 MB for (Azure US government, includes GCC High, Dept. of Defense)
- Service download limit: 65 MB (Azure Global includes GCC), 150 MB for (Azure US government, includes GCC High, Dept. of Defense)

When a CRL download fails, the following message appears:

"The Certificate Revocation List (CRL) downloaded from {uri} has exceeded the maximum allowed size ({size} bytes) for CRLs in Microsoft Entra ID. Try again in few minutes. If the issue persists, contact your tenant administrators."

Download remains in the background with higher limits.

We're reviewing the impact of these limits and have plans to remove them.

I see a valid Certificate Revocation List (CRL) endpoint set, but why don't I see any CRL revocation?

- Make sure the CRL distribution point is set to a valid HTTP URL.
- Make sure the CRL distribution point is accessible via an internet-facing URL.
- Make sure the CRL sizes are within limits.

How do I instantly revoke a certificate?

Follow the steps to [manually revoke a certificate](#).

Will the changes to the Authentication methods policy take effect immediately?

The policy is cached. After a policy update, it might take up to an hour for the changes to take effect.

Why do I see the certificate-based authentication option after it fails?

The Authentication method policy always shows all available authentication methods to the user so they can retry sign-in using any method they prefer. Microsoft Entra ID doesn't hide available methods based on success or failure of a sign-in.

Why does certificate-based auth (CBA) loops once it fails?

The browser caches the certificate after the certificate picker appears. If the user retries, the cached certificate is used automatically. The user should close the browser, and reopen a new session to try CBA again.

Why doesn't proof up for registering other auth methods come up when I use single factor certificates?

A user is considered capable for MFA when the user is in scope for **Certificate-based authentication** in the Authentication methods policy. This policy requirement means a user can't use proof up as part of their authentication to register other available methods.

How can I use single-factor certificates to complete MFA?

We have support for single factor CBA to get MFA. CBA SF + passwordless phone sign-in (PSI) and CBA SF + FIDO2 are the two supported combinations to get MFA using single factor certificates. [MFA with single factor certificates](#)

CertificateUserIds update fails with value already there. How can an admin query all the user objects with the same value?

Tenant admins can run Microsoft Graph queries to find all the users with a given certificateUserId value. For more information, see [CertificateUserIds graph queries](#).

This command returns all user objects that have the value 'bob@contoso.com' value in certificateUserIds:

HTTP

```
GET https://graph.microsoft.com/v1.0/users?$filter=certificateUserIds/any(x:x eq 'bob@contoso.com')
```

After a CRL endpoint is configured, end users aren't able to sign in and they see the following diagnostic message: `http AADSTS500173: Unable to download CRL. Invalid status code Forbidden from CRL distribution point errorCode: 500173`

This is commonly seen when a firewall rule setting blocks access to the CRL endpoint.

Can Microsoft Entra CBA be used on Surface Hub?

Yes. CBA works out-of-the-box for most smart card and smart card reader combinations. If the smart card and smart card reader combination requires other drivers, they must be installed before you can use the smart card and smart card reader combination on the Surface Hub.

Next steps

If your question isn't answered here, see the following related topics:

- Overview of Microsoft Entra CBA
- Technical deep dive for Microsoft Entra CBA
- Microsoft Entra CBA on iOS devices
- Microsoft Entra CBA on Android devices
- How to configure Microsoft Entra CBA
- Windows smart card logon using Microsoft Entra CBA
- Certificate user IDs
- How to migrate federated users

Get started with certificate-based authentication in Microsoft Entra ID with federation

Article • 01/28/2025

Certificate-based authentication (CBA) with federation enables Microsoft Entra ID to authenticate you with a client certificate on a Windows, Android, or iOS device when connecting your Exchange online account to:

- Microsoft mobile applications such as Microsoft Outlook and Microsoft Word
- Exchange ActiveSync (EAS) clients

Configuring this feature eliminates the need to enter a username and password combination into certain mail and Microsoft Office applications on your mobile device.

ⓘ Note

As an alternative, organizations can deploy Microsoft Entra CBA without needing federation. For more information, see [Overview of Microsoft Entra certificate-based authentication against Microsoft Entra ID](#).

This topic:

- Provides steps to configure and utilize CBA for users of tenants in Office 365 Enterprise, Business, Education, and US Government plans.
- Assumes that you already have a [public key infrastructure \(PKI\)](#) and [AD FS](#) configured.

Requirements

To configure CBA with federation, the following statements must be true:

- CBA with federation is only supported for Federated environments for browser applications, native clients using modern authentication, or MSAL libraries. The one exception is Exchange Active Sync (EAS) for Exchange Online (EXO), which can be used for federated and managed accounts. To configure Microsoft Entra CBA without needing federation, see [How to configure Microsoft Entra certificate-based authentication](#).

- The root certificate authority and any intermediate certificate authorities must be configured in Microsoft Entra ID.
- Each certificate authority must have a certificate revocation list (CRL) that can be referenced via an internet-facing URL.
- You must have at least one certificate authority configured in Microsoft Entra ID. You can find related steps in the [Configure the certificate authorities](#) section.
- For Exchange ActiveSync clients, the client certificate must have the user's routable email address in Exchange online in either the Principal Name or the RFC822 Name value of the Subject Alternative Name field. Microsoft Entra ID maps the RFC822 value to the Proxy Address attribute in the directory.
- Your client device must have access to at least one certificate authority that issues client certificates.
- A client certificate for client authentication must have been issued to your client.

 **Important**

The maximum size of a CRL for Microsoft Entra ID to successfully download and cache is 20MB, and the time required to download the CRL must not exceed 10 seconds. If Microsoft Entra ID can't download a CRL, certificate based authentications using certificates issued by the corresponding CA will fail. Best practices to ensure CRL files are within size constraints are to keep certificate lifetimes to within reasonable limits and to clean up expired certificates.

Step 1: Select your device platform

As a first step, for the device platform you care about, you need to review the following:

- The Office mobile applications support
- The specific implementation requirements

The related information exists for the following device platforms:

- [Android](#)
- [iOS](#)

Step 2: Configure the certificate authorities

To configure your certificate authorities in Microsoft Entra ID, for each certificate authority, upload the following:

- The public portion of the certificate, in .cer format

- The internet-facing URLs where the Certificate Revocation Lists (CRLs) reside

The schema for a certificate authority looks as follows:

C#

```
class TrustedCAsForPasswordlessAuth
{
    CertificateAuthorityInformation[] certificateAuthorities;
}

class CertificateAuthorityInformation
{
    CertAuthorityType authorityType;
    X509Certificate trustedCertificate;
    string crlDistributionPoint;
    string deltaCrlDistributionPoint;
    string trustedIssuer;
    string trustedIssuerSKI;
}

enum CertAuthorityType
{
    RootAuthority = 0,
    IntermediateAuthority = 1
}
```

For the configuration, you can use [Microsoft Graph PowerShell](#):

1. Start Windows PowerShell with administrator privileges.
2. Install [Microsoft Graph PowerShell](#):

PowerShell

```
Install-Module Microsoft.Graph
```

As a first configuration step, you need to establish a connection with your tenant. As soon as a connection to your tenant exists, you can review, add, delete, and modify the trusted certificate authorities that are defined in your directory.

Connect

To establish a connection with your tenant, use [Connect-MgGraph](#):

PowerShell

Retrieve

To retrieve the trusted certificate authorities that are defined in your directory, use [Get-MgOrganizationCertificateBasedAuthConfiguration](#).

PowerShell

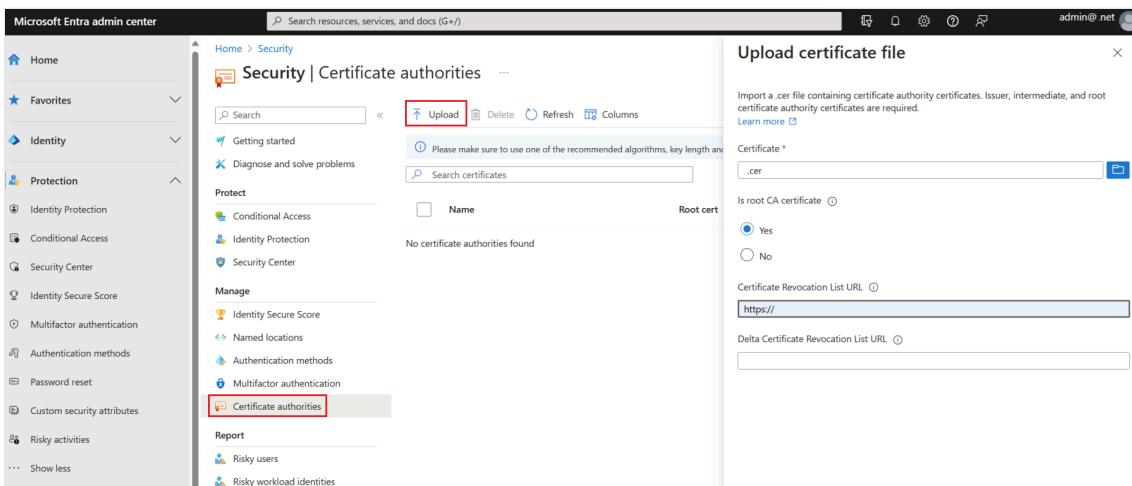
```
Get-MgOrganizationCertificateBasedAuthConfiguration
```

Important

Microsoft recommends that you use roles with the fewest permissions. This practice helps improve security for your organization. Global Administrator is a highly privileged role that should be limited to emergency scenarios or when you can't use an existing role.

To add, modify, or remove a CA, use the Microsoft Entra admin center:

1. Sign in to the [Microsoft Entra admin center](#) as a [Global Administrator](#).
2. Browse to **Protection > Show more > Security Center (or Identity Secure Score) > Certificate authorities**.
3. To upload a CA, select **Upload**:
 - a. Select the CA file.
 - b. Select **Yes** if the CA is a root certificate, otherwise select **No**.
 - c. For **Certificate Revocation List URL**, set the internet-facing URL for the CA base CRL that contains all revoked certificates. If the URL isn't set, authentication with revoked certificates won't fail.
 - d. For **Delta Certificate Revocation List URL**, set the internet-facing URL for the CRL that contains all revoked certificates since the last base CRL was published.
 - e. Select **Add**.



4. To delete a CA certificate, select the certificate and select **Delete**.

5. Select **Columns** to add or delete columns.

Step 3: Configure revocation

To revoke a client certificate, Microsoft Entra ID fetches the certificate revocation list (CRL) from the URLs uploaded as part of certificate authority information and caches it. The last publish timestamp (**Effective Date** property) in the CRL is used to ensure the CRL is still valid. The CRL is periodically referenced to revoke access to certificates that are a part of the list.

If a more instant revocation is required (for example, if a user loses a device), the authorization token of the user can be invalidated. To invalidate the authorization token, set the **StsRefreshTokensValidFrom** field for this particular user using Windows PowerShell. You must update the **StsRefreshTokensValidFrom** field for each user you want to revoke access for.

To ensure that the revocation persists, you must set the **Effective Date** of the CRL to a date after the value set by **StsRefreshTokensValidFrom** and ensure the certificate in question is in the CRL.

The following steps outline the process for updating and invalidating the authorization token by setting the **StsRefreshTokensValidFrom** field.

```
https

# Authenticate to Microsoft Graph
Connect-MgGraph -Scopes "User.Read.All"

# Get the user
$user = Get-MgUser -UserPrincipalName "test@yourdomain.com"
```

```
# Get the StsRefreshTokensValidFrom property  
$user.StsRefreshTokensValidFrom
```

The date you set must be in the future. If the date is not in the future, the **StsRefreshTokensValidFrom** property is not set. If the date is in the future, **StsRefreshTokensValidFrom** is set to the current time (not the date indicated by Set-MsolUser command).

Step 4: Test your configuration

Testing your certificate

As a first configuration test, you should try to sign in to [Outlook Web Access](#) or [SharePoint Online](#) using your **on-device browser**.

If your sign-in is successful, then you know that:

- The user certificate has been provisioned to your test device
- AD FS is configured correctly

Testing Office mobile applications

1. On your test device, install an Office mobile application (for example, OneDrive).
2. Launch the application.
3. Enter your username, and then select the user certificate you want to use.

You should be successfully signed in.

Testing Exchange ActiveSync client applications

To access Exchange ActiveSync (EAS) via certificate-based authentication, an EAS profile containing the client certificate must be available to the application.

The EAS profile must contain the following information:

- The user certificate to be used for authentication
- The EAS endpoint (for example, outlook.office365.com)

An EAS profile can be configured and placed on the device through the utilization of Mobile device management (MDM) such as Microsoft Intune or by manually placing the certificate in the EAS profile on the device.

Testing EAS client applications on Android

1. Configure an EAS profile in the application that satisfies the requirements in the prior section.
2. Open the application, and verify that mail is synchronizing.

Next steps

[Additional information about certificate-based authentication on Android devices.](#)

[Additional information about certificate-based authentication on iOS devices.](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Microsoft Entra certificate-based authentication with federation on Android

Article • 03/04/2025

Android devices can use certificate-based authentication (CBA) to authenticate to Microsoft Entra ID using a client certificate on their device when connecting to:

- Office mobile applications such as Microsoft Outlook and Microsoft Word
- Exchange ActiveSync (EAS) clients

Configuring this feature eliminates the need to enter a username and password combination into certain mail and Microsoft Office applications on your mobile device.

Microsoft mobile applications support

[\[...\] Expand table](#)

Apps	Support
Azure Information Protection app	✓
Intune Company Portal	✓
Microsoft Teams	✓
OneNote	✓
OneDrive	✓
Outlook	✓
Power BI	✓
Skype for Business	✓
Word / Excel / PowerPoint	✓
Yammer	✓

Implementation requirements

The device OS version must be Android 5.0 (Lollipop) and above.

A federation server must be configured.

For Microsoft Entra ID to revoke a client certificate, the AD FS token must have the following claims:

- `http://schemas.microsoft.com/ws/2008/06/identity/claims/<serialnumber>` (The serial number of the client certificate)
- `http://schemas.microsoft.com/2012/12/certificatecontext/field/<issuer>` (The string for the issuer of the client certificate)

Microsoft Entra ID adds these claims to the refresh token if they're available in the AD FS token (or any other SAML token). When the refresh token needs to be validated, this information is used to check the revocation.

As a best practice, you should update your organization's AD FS error pages with the following information:

- The requirement for installing the Microsoft Authenticator on Android.
- Instructions on how to get a user certificate.

For more information, see [Customizing the AD FS Sign-in Pages](#).

Office apps with modern authentication enabled send '*'prompt=login'*' to Microsoft Entra ID in their request. By default, Microsoft Entra ID translates '*'prompt=login'*' in the request to AD FS as '*'wauth=usernamepassworduri'*' (asks AD FS to do U/P Auth) and '*'wfresh=0'*' (asks AD FS to ignore SSO state and do a fresh authentication). If you want to enable certificate-based authentication for these apps, you need to modify the default Microsoft Entra behavior. Set the '*'PromptLoginBehavior'*' in your federated domain settings to '*'Disabled'*'. You can use [New-MgDomainFederationConfiguration](#) to perform this task:

PowerShell

```
New-MgDomainFederationConfiguration -DomainId <domain> -PromptLoginBehavior "disabled"
```

Exchange ActiveSync clients support

Certain Exchange ActiveSync applications on Android 5.0 (Lollipop) or later are supported. To determine if your email application does support this feature, contact your application developer.

Next steps

If you want to configure certificate-based authentication in your environment, see [Get started with certificate-based authentication on Android](#) for instructions.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Microsoft Entra certificate-based authentication with federation on iOS

Article • 03/04/2025

To improve security, iOS devices can use certificate-based authentication (CBA) to authenticate to Microsoft Entra ID using a client certificate on their device when connecting to the following applications or services:

- Office mobile applications such as Microsoft Outlook and Microsoft Word
- Exchange ActiveSync (EAS) clients

Using certificates eliminates the need to enter a username and password combination into certain mail and Microsoft Office applications on your mobile device.

Microsoft mobile applications support

[\[+\] Expand table](#)

Apps	Support
Azure Information Protection app	✓
Company Portal	✓
Microsoft Teams	✓
Office (mobile)	✓
OneNote	✓
OneDrive	✓
Outlook	✓
Power BI	✓
Skype for Business	✓
Word / Excel / PowerPoint	✓
Yammer	✓

Requirements

To use CBA with iOS, the following requirements and considerations apply:

- The device OS version must be iOS 9 or above.
- Microsoft Authenticator is required for Office applications on iOS.
- An identity preference must be created in the macOS Keychain that includes the authentication URL of the AD FS server. For more information, see [Create an identity preference in Keychain Access on Mac](#).

The following Active Directory Federation Services (AD FS) requirements and considerations apply:

- The AD FS server must be enabled for certificate authentication and use federated authentication.
- The certificate needs to have to use Enhanced Key Usage (EKU) and contain the UPN of the user in the *Subject Alternative Name (NT Principal Name)*.

Configure AD FS

For Microsoft Entra ID to revoke a client certificate, the AD FS token must have the following claims. Microsoft Entra ID adds these claims to the refresh token if they're available in the AD FS token (or any other SAML token). When the refresh token needs to be validated, this information is used to check the revocation:

- `http://schemas.microsoft.com/ws/2008/06/identity/claims/<serialnumber>` - add the serial number of your client certificate
- `http://schemas.microsoft.com/2012/12/certificatecontext/field/<issuer>` - add the string for the issuer of your client certificate

As a best practice, you also should update your organization's AD FS error pages with the following information:

- The requirement for installing the Microsoft Authenticator on iOS.
- Instructions on how to get a user certificate.

For more information, see [Customizing the AD FS sign in page](#).

Use modern authentication with Office apps

Some Office apps with modern authentication enabled send `prompt=login` to Microsoft Entra ID in their request. By default, Microsoft Entra ID translates `prompt=login` in the request to AD FS as `wauth=usernamepassworduri` (asks AD FS to do U/P Auth) and `wfresh=0` (asks AD FS to ignore SSO state and do a fresh authentication). If you want to

enable certificate-based authentication for these apps, modify the default Microsoft Entra behavior.

To update the default behavior, set the '*PromptLoginBehavior*' in your federated domain settings to *Disabled*. You can use the [New-MgDomainFederationConfiguration](#) cmdlet to perform this task, as shown in the following example:

PowerShell

```
New-MgDomainFederationConfiguration -DomainId <domain> -PromptLoginBehavior "disabled"
```

Support for Exchange ActiveSync clients

On iOS 9 or later, the native iOS mail client is supported. To determine if this feature is supported for all other Exchange ActiveSync applications, contact your application developer.

Next steps

To configure certificate-based authentication in your environment, see [Get started with certificate-based authentication](#) for instructions.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Configure and enable users for SMS-based authentication using Microsoft Entra ID

Article • 03/04/2025

To simplify and secure sign-in to applications and services, Microsoft Entra ID provides multiple authentication options. SMS-based authentication lets users sign-in without providing, or even knowing, their user name and password. After their account is created by an identity administrator, they can enter their phone number at the sign-in prompt. They receive an SMS authentication code that they can provide to complete the sign-in. This authentication method simplifies access to applications and services, especially for Frontline workers.

This article shows you how to enable SMS-based authentication for select users or groups in Microsoft Entra ID. For a list of apps that support using SMS-based sign-in, see [App support for SMS-based authentication](#).

Before you begin

To complete this article, you need the following resources and privileges:

- An active Azure subscription.
 - If you don't have an Azure subscription, [create an account](#).
- A Microsoft Entra tenant associated with your subscription.
 - If needed, [create a Microsoft Entra tenant](#) or [associate an Azure subscription with your account](#).
- You need at least the [Authentication Policy Administrator](#) role in your Microsoft Entra tenant to enable SMS-based authentication.
- Each user that's enabled in the SMS authentication method policy must be licensed, even if they don't use it. Each enabled user must have one of the following Microsoft Entra ID, EMS, Microsoft 365 licenses:
 - [Microsoft 365 F1 or F3](#)
 - [Microsoft Entra ID P1 or P2](#)
 - [Enterprise Mobility + Security \(EMS\) E3 or E5](#) or [Microsoft 365 E3 or E5](#)
 - [Office 365 F3](#)

Known issues

Here are some known issues:

- SMS-based authentication isn't currently compatible with Microsoft Entra multifactor authentication.
- Except for Teams, SMS-based authentication isn't compatible with native Office applications.
- SMS-based authentication isn't supported for B2B accounts.
- Federated users won't authenticate in the home tenant. They only authenticate in the cloud.
- If a user's default sign-in method is a text or call to your phone number, then the SMS code or voice call is sent automatically during multifactor authentication. As of June 2021, some apps will ask users to choose **Text** or **Call** first. This option prevents sending too many security codes for different apps. If the default sign-in method is the Microsoft Authenticator app ([which we highly recommend](#) ↗), then the app notification is sent automatically.
- [Cross-tenant synchronization](#) does not support users with SMS sign-in enabled.

Enable the SMS-based authentication method

There are three main steps to enable and use SMS-based authentication in your organization:

- Enable the authentication method policy.
- Select users or groups that can use the SMS-based authentication method.
- Assign a phone number for each user account.
 - This phone number can be assigned in the Microsoft Entra admin center (which is shown in this article), and in *My Staff* or *My Account*.

First, let's enable SMS-based authentication for your Microsoft Entra tenant.

1. Sign in to the [Microsoft Entra admin center](#) ↗ as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Policies**.
3. From the list of available authentication methods, select **SMS**.

The screenshot shows the Microsoft Entra ID Security interface. On the left, there's a navigation pane with sections like Manage, Policies, Password protection, Registration campaign, Authentication strengths, and Settings. The main area is titled "Manage migration" with a note about deprecating legacy policies. A table lists various authentication methods with columns for Method, Target, and Enabled. The "SMS" method is highlighted with a red box.

Method	Target	Enabled
FIDO2 security key		No
Microsoft Authenticator		No
SMS		No
Temporary Access Pass		No
Hardware OATH tokens (Preview)		No
Third-party software OATH tokens		No
Voice call		No
Email OTP		Yes
Certificate-based authentication		No

4. Select **Enable** and select **Target users**. You can choose to enable SMS-based authentication for *All users* or *Select users and groups*.

➊ Note

To configure SMS-based authentication for first-factor (that is, to allow users to sign in with this method), check the **Use for sign-in** checkbox. Leaving this unchecked makes SMS-based authentication available for multifactor authentication and Self-Service Password Reset only.

This screenshot shows the "SMS settings" window. It includes a note about SMS being used for multi-factor authentication and self-service password reset. The "Enable and Target" section has a toggle switch for "Enable" set to "On". Under "Include" and "Exclude", "All users" is selected. In the table below, "All users" is listed as a "Group" type with the "Use for sign-in" checkbox checked and "Registration" set to "Optional".

Name	Type	Use for sign-in	Registration
All users	Group	<input checked="" type="checkbox"/>	Optional

Assign the authentication method to users and groups

With SMS-based authentication enabled in your Microsoft Entra tenant, now select some users or groups to be allowed to use this authentication method.

1. In the SMS authentication policy window, set **Target** to *Select users*.
2. Choose to **Add users or groups**, then select a test user or group, such as *Contoso User* or *Contoso SMS Users*.

3. When you've selected your users or groups, choose **Select**, then **Save** the updated authentication method policy.

Each user that's enabled in SMS authentication method policy must be licensed, even if they don't use it. Make sure you have the appropriate licenses for the users you enable in the authentication method policy, especially when you enable the feature for large groups of users.

Set a phone number for user accounts

Users are now enabled for SMS-based authentication, but their phone number must be associated with the user profile in Microsoft Entra ID before they can sign-in. The user can [set this phone number themselves](#) in *My Account*, or you can assign the phone number using the Microsoft Entra admin center. Phone numbers can be set by those with at least the [Authentication Administrator](#) role.

When a phone number is set for SMS-based sign-in, it's also then available for use with [Microsoft Entra multifactor authentication](#) and [self-service password reset](#).

1. Search for and select **Microsoft Entra ID**.
2. From the navigation menu on the left-hand side of the Microsoft Entra window, select **Users**.
3. Select the user you enabled for SMS-based authentication in the previous section, such as *Contoso User*, then select **Authentication methods**.
4. Select **+ Add authentication method**, then in the *Choose method* drop-down menu, choose **Phone number**.

Enter the user's phone number, including the country code, such as +1 xxxxxxxxx. The Microsoft Entra admin center validates the phone number is in the correct format.

Then, from the *Phone type* drop-down menu, select *Mobile*, *Alternate mobile*, or *Other* as needed.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various user management options like Profile, Assigned roles, Groups, Applications, Licenses, Devices, and Azure role assignments. The 'Authentication methods' option is selected and highlighted with a red box. On the right, a modal window titled 'Add authentication method' is open. Inside, it says 'Choose method' with a dropdown set to 'Phone number'. Below that, there's a note about adding a phone number for authentication. A form is present with 'Phone number *' containing '+1 4251234567' and 'Phone type' set to 'Mobile'. At the bottom of the modal is a blue 'Add' button.

The phone number must be unique in your tenant. If you try to use the same phone number for multiple users, an error message is shown.

5. To apply the phone number to a user's account, select **Add**.

When successfully provisioned, a check mark appears for *SMS Sign-in enabled*.

Test SMS-based sign-in

To test the user account that's now enabled for SMS-based sign-in, complete the following steps:

1. Open a new InPrivate or Incognito web browser window to <https://www.office.com>
2. In the top right-hand corner, select **Sign in**.
3. At the sign-in prompt, enter the phone number associated with the user in the previous section, then select **Next**.



Sign in

+14251234567

No account? [Create one!](#)

[Can't access your account?](#)

[Sign-in options](#)

[Next](#)

4. An SMS message is sent to the phone number provided. To complete the sign-in process, enter the 6-digit code provided in the SMS message at the sign-in prompt.



← [+1 425-123-4567](#)

Enter code

We just sent a code to [+1 425-123-4567](#)

Enter code

[Sign in](#)

5. The user is now signed in without the need to provide a username or password.

Troubleshoot SMS-based sign-in

You can use the following scenarios and troubleshooting steps if you have problems with enabling and using SMS-based sign-in. For a list of apps that support using SMS-based sign-in, see [App support for SMS-based authentication](#).

Phone number already set for a user account

If a user has already registered for Microsoft Entra multifactor authentication and / or self-service password reset (SSPR), they already have a phone number associated with their account. This phone number isn't automatically available for use with SMS-based sign-in.

A user that has a phone number already set for their account is displayed a button to *Enable for SMS sign-in* in their **My Profile** page. Select this button, and the account is enabled for use with SMS-based sign-in and the previous Microsoft Entra multifactor authentication or SSPR registration.

For more information on the end-user experience, see [SMS sign-in user experience for phone number](#).

Error when trying to set a phone number on a user's account

If you receive an error when you try to set a phone number for a user account in the Microsoft Entra admin center, review the following troubleshooting steps:

1. Make sure that you're enabled for the SMS-based sign-in.
2. Confirm that the user account is enabled in the **SMS** authentication method policy.
3. Make sure you set the phone number with the proper formatting, as validated in the Microsoft Entra admin center (such as +1 4251234567).
4. Make sure that the phone number isn't used elsewhere in your tenant.
5. Check there's no voice number set on the account. If a voice number is set, delete and try to the phone number again.

Next steps

- For a list of apps that support using SMS-based sign-in, see [App support for SMS-based authentication](#).
- For more ways to sign-in to Microsoft Entra ID without a password, such as the Microsoft Authenticator App or FIDO2 security keys, see [Passwordless authentication options for Microsoft Entra ID](#).
- You can also use the Microsoft Graph REST API to [enable](#) or [disable](#) SMS-based sign-in.

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

App support for SMS-based authentication

Article • 03/04/2025

SMS-based authentication is available to Microsoft apps integrated with the Microsoft identity platform (Microsoft Entra ID). The table lists some of the web and mobile apps that support SMS-based authentication. If you would like to add or validate any app, [contact us](#).

[+] Expand table

App	Web/browser app	Native mobile app
Office 365- Microsoft Online Services*	•	
Microsoft One Note	•	
Microsoft Teams	•	•
Company portal	•	•
My Apps portal	•	Not available
Microsoft Forms	•	Not available
Microsoft Edge	•	
Microsoft Power BI	•	
Microsoft Stream	•	
Microsoft Power Apps	•	
Microsoft Azure	•	•
Azure Virtual Desktop	•	

*SMS sign-in isn't available for office applications, such as Word, Excel, etc., when accessed directly on the web, but is available when accessed through the [Office 365 web app](#).

The above mentioned Microsoft apps support SMS sign-in is because they use the Microsoft Identity login (<https://login.microsoftonline.com/>), which allows users to enter phone number and SMS code.

Unsupported Microsoft apps

Microsoft 365 desktop (Windows or Mac) apps and Microsoft 365 web apps (except MS One Note) that are accessed directly on the web don't support SMS sign-in. These apps use the Microsoft Office login (https://office.live.com/start/*) that requires a password to sign in. For the same reason, Microsoft Office mobile apps (except Microsoft Teams, Company portal, and Microsoft Azure) don't support SMS sign-in.

[+] [Expand table](#)

Unsupported Microsoft apps	Examples
Native desktop Microsoft apps	Microsoft Teams, Microsoft 365 apps, Word, Excel, and so on.
Native mobile Microsoft apps (except Microsoft Teams, Company portal, and Microsoft Azure)	Outlook, Edge, Power BI, Stream, SharePoint, Power Apps, Word, and so on.
Microsoft 365 web apps (accessed directly on web)	Outlook , Word , Excel , PowerPoint

Support for Non-Microsoft apps

To make Non-Microsoft apps compatible with the SMS sign-in feature:

- Integrate Non-Microsoft web apps with Microsoft Entra ID and use Microsoft Entra authentication. Use Security Assertion Markup Language [SAML](#) or OpenID Connect [OIDC](#) to integrate with Microsoft Entra SSO.
- Integrate Non-Microsoft on-premises apps with Microsoft Entra ID using [Microsoft Entra application proxy](#)
- Integrate Non-Microsoft client apps with [Microsoft identity platform](#) for authentication
 - [Sample app iOS](#)
 - [Sample app Android](#)

Next steps

- [How to enable SMS-based sign-in for users](#)
- See the following links to enable SMS sign-in for native mobile apps using MSAL Libraries:
 - [iOS](#)
 - [Android](#)
- [Integrate SAAS application with Microsoft Entra ID](#)

Recommended content

- Add an application to your Microsoft Entra ID
 - Overview of MSAL libraries to acquire token from Microsoft identity platform to authenticate users
 - Configure Microsoft Managed Home Screen with Microsoft Entra ID
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

Two-way SMS unsupported

Article • 03/04/2025

Two-way SMS for Azure Multi-Factor Authentication Server was originally deprecated in 2018, and no longer supported after February 24, 2021, except for organizations that received a support extension until August 2, 2021. Administrators should enable another method for users who still use two-way SMS.

Email notifications and Service Health notifications (portal toasts) were sent to affected admins on December 8, 2020 and January 28, 2021. The alerts went to the Owner, Co-Owner, Admin, and Service Admin RBAC roles tied to the subscriptions. If you've already completed the following steps, no action is necessary.

Required actions

1. Enable the mobile app for your users, if you haven't done so already. For more information, see [Enable mobile app authentication with MFA Server](#).
2. Notify your end users to visit your MFA Server [User portal](#) to activate the mobile app. The [Microsoft Authenticator app](#) is the recommended verification option since it's more secure than two-way SMS. For more information, see [It's Time to Hang Up on Phone Transports for Authentication](#).
3. Change the user settings from two-way text message to mobile app as the default method.

FAQ

What if I don't change the default method from two-way SMS to the mobile app?

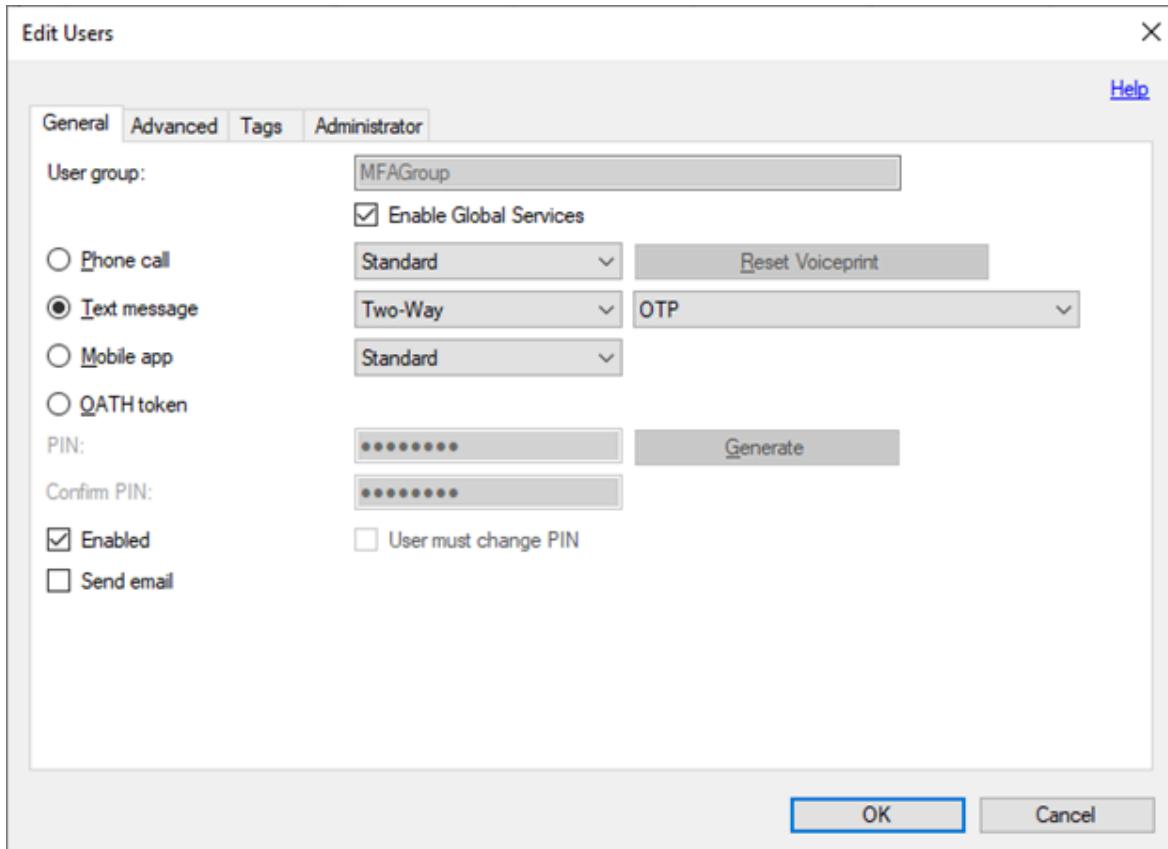
Two-way SMS fails after February 24, 2021. Users will see an error when they try to sign in and pass MFA.

How do I change the user settings from two-way text message to mobile app?

You should change the user settings by following these steps:

1. In MFA Server, filter the user list for two-way text message.

2. Select all users.
3. Open the Edit Users dialog.
4. Change users from Text message to Mobile app.



Do my users need to take any action? If yes, how?

Yes. Your end users need to visit your specific MFA Server User portal to activate the mobile app, if they haven't done so already. After you've done Step 3, any users that didn't visit the User Portal to set up the mobile app will start failing sign in until they visit the User portal to re-register.

What if my users can't install the mobile app? What other options do they have?

The alternative to two-way SMS or the mobile app is a phone call. However, the Microsoft Authenticator app is the recommended verification method.

Will one-way SMS be deprecated as well?

No, just two-way SMS is being deprecated. For MFA Server, one-way SMS works for a subset of scenarios:

- AD FS Adapter
- IIS Authentication (requires User Portal and configuration)
- RADIUS (requires that RADIUS clients support access challenge and that PAP protocol is used)

There are limitations to when one-way SMS can be used that make the mobile app a better alternative because it doesn't require the verification code prompt. If you still want to use one-way SMS in some scenarios, then you could leave these checked, but change the **Company Settings** section, **General** tab **User Defaults Text Message** to **One-Way** instead of **Two-Way**. Lastly, if you use Directory Synchronization that defaults to SMS, you'd need to change it to One-Way instead of Two-Way.

How can I check which users are still using two-way SMS?

To list these users, start **MFA Server**, select the **Users** section, click **Filter User List**, and filter for **Text Message Two-Way**.

How do we hide two-way SMS as an option in the MFA portal to prevent users from selecting it in the future?

In **MFA Server** User portal, click **Settings**, you can clear **Text Message** so that it's not available. The same is true in the **AD FS** section if you're using AD FS for user enrollment.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Sign-in to Microsoft Entra ID with email as an alternate login ID (Preview)

Article • 04/17/2025

! Note

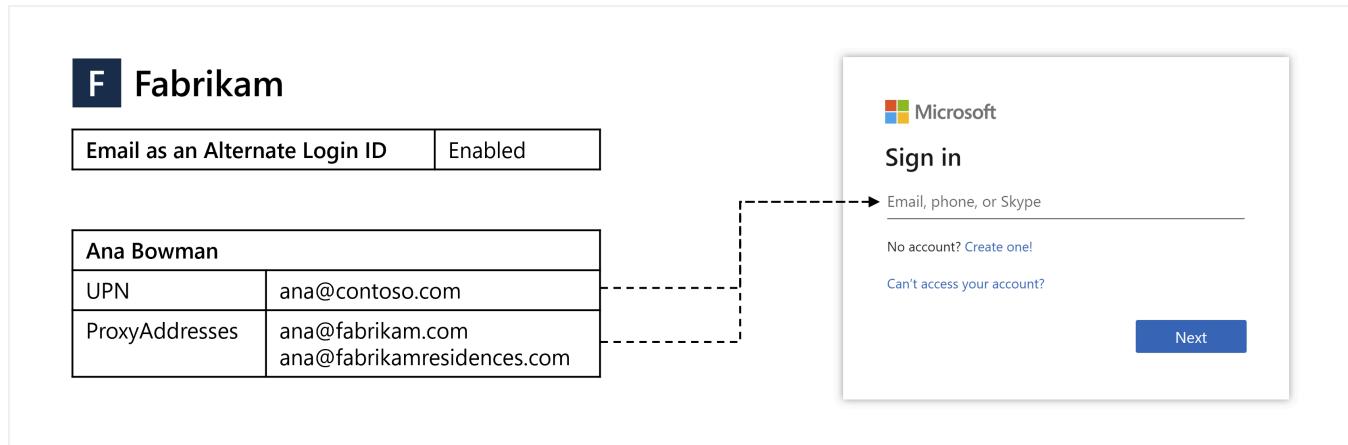
Sign-in to Microsoft Entra ID with email as an alternate login ID is a public preview feature of Microsoft Entra ID. For more information about previews, see [Supplemental Terms of Use for Microsoft Azure Previews](#).

Many organizations want to let users sign in to Microsoft Entra ID using the same credentials as their on-premises directory environment. With this approach, known as hybrid authentication, users only need to remember one set of credentials.

Some organizations haven't moved to hybrid authentication for the following reasons:

- By default, the Microsoft Entra user Principal Name (UPN) is set to the same value as the on-premises UPN.
- Changing the Microsoft Entra UPN creates a mismatch between on-premises and Microsoft Entra environments that could cause problems with certain applications and services.
- Due to business or compliance reasons, the organization doesn't want to use the on-premises UPN to sign in to Microsoft Entra ID.

To move toward hybrid authentication, you can configure Microsoft Entra ID to let users sign in with their email as an alternate login ID. For example, if *Contoso* rebranded to *Fabrikam*, rather than continuing to sign in with the legacy `ana@contoso.com` UPN, email as an alternate login ID can be used. To access an application or service, users would sign in to Microsoft Entra ID using their non-UPN email, such as `ana@fabrikam.com`.



This article shows you how to enable and use email as an alternate login ID.

Before you begin

Here's what you need to know about email as an alternate login ID:

- The feature is available in Microsoft Entra ID Free edition and higher.
- The feature enables sign-in with *ProxyAddresses*, in addition to UPN, for cloud-authenticated Microsoft Entra users. More on how this applies to Microsoft Entra business-to-business (B2B) collaboration in the [B2B](#) section.
- When a user signs in with a non-UPN email, the `unique_name` and `preferred_username` claims (if present) in the [ID token](#) will return the non-UPN email.
 - If the non-UPN email in use becomes stale (no longer belongs to the user), these claims will return the UPN instead.
- The feature supports managed authentication with Password Hash Sync (PHS) or Pass-Through Authentication (PTA).
- There are two options for configuring the feature:
 - [Home Realm Discovery \(HRD\) policy](#) - Use this option to enable the feature for the entire tenant. At least the [Application Administrator](#) role is required.
 - [Staged rollout policy](#) - Use this option to test the feature with specific Microsoft Entra groups. When you first add a security group for staged rollout, you're limited to 200 users to avoid a UX time-out. After you've added the group, you can add more users directly to it, as required.

Preview limitations

In the current preview state, the following limitations apply to email as an alternate login ID:

- **User experience** - Users may see their UPN, even when they signed-in with their non-UPN email. The following example behavior may be seen:
 - User is prompted to sign in with UPN when directed to Microsoft Entra sign-in with `login_hint=<non-UPN_email>`.
 - When a user signs-in with a non-UPN email and enters an incorrect password, the "Enter your password" page changes to display the UPN.
 - On some Microsoft sites and apps, such as Microsoft Office, the *Account Manager* control typically displayed in the upper right may display the user's UPN instead of the non-UPN email used to sign in.
- **Unsupported flows** - Some flows are currently not compatible with non-UPN emails, such as the following:
 - Microsoft Entra ID Protection doesn't match non-UPN emails with *Leaked Credentials* risk detection. This risk detection uses the UPN to match credentials that have been leaked. For more information, see [How To: Investigate risk](#).

- When a user is signed-in with a non-UPN email, they cannot change their password. Microsoft Entra self-service password reset (SSPR) should work as expected. During SSPR, the user may see their UPN if they verify their identity using a non-UPN email.
- **Unsupported scenarios** - The following scenarios are not supported. Sign-in with non-UPN email for:
 - [Microsoft Entra hybrid joined devices](#)
 - [Microsoft Entra joined devices](#)
 - [Microsoft Entra registered devices](#)
 - [Single Sign-On and App Protection Policies on Mobile Platform](#)
 - Legacy authentication such as POP3 and SMTP
- **Unsupported apps** - Some third-party applications may not work as expected if they assume that the `unique_name` or `preferred_username` claims are immutable or will always match a specific user attribute, such as UPN.
- **Logging** - Changes made to the feature's configuration in HRD policy are not explicitly shown in the audit logs.
- **Staged rollout policy** - The following limitations apply only when the feature is enabled using staged rollout policy:
 - The feature does not work as expected for users that are included in other staged rollout policies.
 - Staged rollout policy supports a maximum of 10 groups per feature.
 - Staged rollout policy does not support nested groups.
 - Staged rollout policy does not support dynamic membership groups.
 - Contact objects inside the group will block the group from being added to a staged rollout policy.
- **Duplicate values** - Within a tenant, a cloud-only user's UPN can be the same value as another user's proxy address synced from the on-premises directory. In this scenario, with the feature enabled, the cloud-only user will not be able to sign in with their UPN. More on this issue in the [Troubleshoot](#) section.

Overview of alternate login ID options

To sign in to Microsoft Entra ID, users enter a value that uniquely identifies their account. Historically, you could only use the Microsoft Entra UPN as the sign-in identifier.

For organizations where the on-premises UPN is the user's preferred sign-in email, this approach was great. Those organizations would set the Microsoft Entra UPN to the exact same value as the on-premises UPN, and users would have a consistent sign-in experience.

Alternate Login ID for AD FS

However, in some organizations the on-premises UPN isn't used as a sign-in identifier. In the on-premises environments, you would configure the local AD DS to allow sign-in with an alternate login ID. Setting the Microsoft Entra UPN to the same value as the on-premises UPN isn't an option as Microsoft Entra ID would then require users to sign in with that value.

Alternate Login ID in Microsoft Entra Connect

The typical workaround to this issue was to set the Microsoft Entra UPN to the email address the user expects to sign in with. This approach works, though results in different UPNs between the on-premises AD and Microsoft Entra ID, and this configuration isn't compatible with all Microsoft 365 workloads.

Email as an Alternate Login ID

A different approach is to synchronize the Microsoft Entra ID and on-premises UPNs to the same value and then configure Microsoft Entra ID to allow users to sign in to Microsoft Entra ID with a verified email. To provide this ability, you define one or more email addresses in the user's *ProxyAddresses* attribute in the on-premises directory. *ProxyAddresses* are then synchronized to Microsoft Entra ID automatically using Microsoft Entra Connect.

 Expand table

Option	Description
Alternate Login ID for AD FS	Enable sign-in with an alternate attribute (such as Mail) for AD FS users.
Alternate Login ID in Microsoft Entra Connect	Synchronize an alternate attribute (such as Mail) as the Microsoft Entra UPN.
Email as an Alternate Login ID	Enable sign-in with verified domain <i>ProxyAddresses</i> for Microsoft Entra users.

Synchronize sign-in email addresses to Microsoft Entra ID

Traditional Active Directory Domain Services (AD DS) or Active Directory Federation Services (AD FS) authentication happens directly on your network and is handled by your AD DS infrastructure. With hybrid authentication, users can instead sign in directly to Microsoft Entra ID.

To support this hybrid authentication approach, you synchronize your on-premises AD DS environment to Microsoft Entra ID using [Microsoft Entra Connect](#) and configure it to use PHS or PTA. For more information, see [Choose the right authentication method for your Microsoft Entra hybrid identity solution](#).

In both configuration options, the user submits their username and password to Microsoft Entra ID, which validates the credentials and issues a ticket. When users sign in to Microsoft Entra ID, it removes the need for your organization to host and manage an AD FS infrastructure.

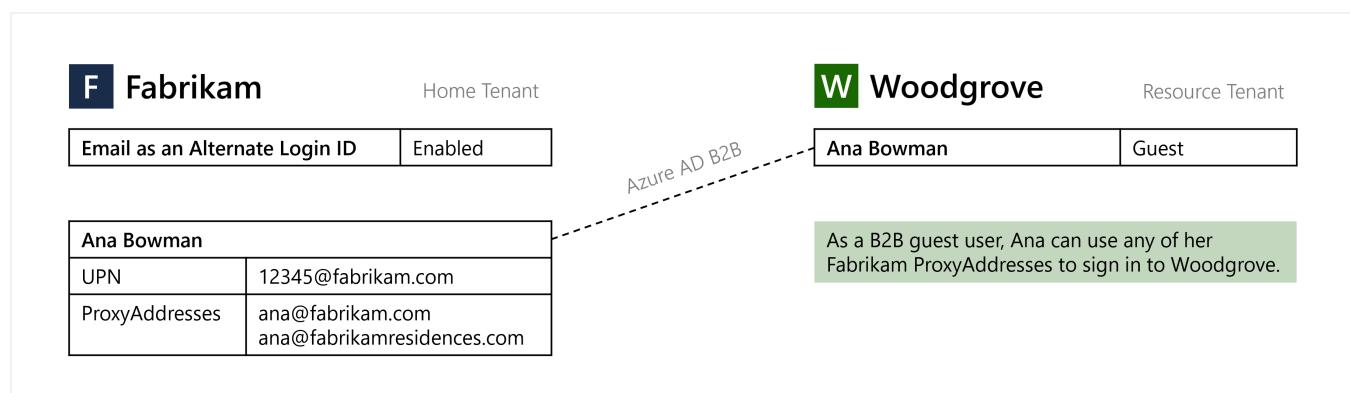
One of the user attributes that's automatically synchronized by Microsoft Entra Connect is *ProxyAddresses*. If users have an email address defined in the on-premises AD DS environment as part of the *ProxyAddresses* attribute, it's automatically synchronized to Microsoft Entra ID. This email address can then be used directly in the Microsoft Entra sign-in process as an alternate login ID.

i Important

Only emails in verified domains for the tenant are synchronized to Microsoft Entra ID. Each Microsoft Entra tenant has one or more verified domains, for which you have proven ownership, and are uniquely bound to your tenant.

For more information, see [Add and verify a custom domain name in Microsoft Entra ID](#).

B2B guest user sign-in with an email address



Email as an alternate login ID applies to [Microsoft Entra B2B collaboration](#) under a "bring your own sign-in identifiers" model. When email as an alternate login ID is enabled in the home tenant, Microsoft Entra users can perform guest sign in with non-UPN email on the resource tenant endpoint. No action is required from the resource tenant to enable this functionality.

! Note

When an alternate login ID is used on a resource tenant endpoint that does not have the functionality enabled, the sign-in process will work seamlessly, but SSO will be interrupted.

Enable user sign-in with an email address

! Note

This configuration option uses HRD policy. For more information, see [homeRealmDiscoveryPolicy resource type](#).

Once users with the *ProxyAddresses* attribute applied are synchronized to Microsoft Entra ID using Microsoft Entra Connect, you need to enable the feature for users to sign in with email as an alternate login ID for your tenant. This feature tells the Microsoft Entra login servers to not only check the sign-in identifier against UPN values, but also against *ProxyAddresses* values for the email address.

You can use either Microsoft Entra admin center or Graph PowerShell to set up the feature.

Microsoft Entra admin center

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Hybrid Identity Administrator](#).
2. Browse to **Entra ID > Entra Connect > Connect Sync**
3. Select **Email as alternate login ID****.

The screenshot shows the Microsoft Entra Connect | Connect Sync page. The left sidebar has a tree view with 'Microsoft Entra Connect' selected under 'User experiences'. The main content area has a breadcrumb trail: Home > Authentication methods > Authentication strengths > Devices > All devices > Microsoft Entra Connect | Cloud Sync > Cloud sync | Configurations > Microsoft Entra Connect. The main content area has several sections: 'USER SIGN-IN' (with 'Email as alternate login ID' checked), 'STAGED ROLLOUT OF CLOUD AUTHENTICATION' (with a note to enable staged rollout for managed user sign-in), 'ON-PREMISES APPLICATIONS' (with a note to configure remote access for on-premises applications), and 'HEALTH AND ANALYTICS' (with a note to monitor on-premises identity infrastructure and synchronization services in the cloud).

4. Click the checkbox next to *Email as an alternate login ID*.

5. Click Save.

The screenshot shows a configuration page for enabling email as an alternate login ID. At the top, there's a breadcrumb navigation: Home > Contoso | Microsoft Entra Connect > Microsoft Entra Connect | Connect Sync >. The main title is "Email as an alternate login ID". Below the title, a subtitle reads: "This feature allows cloud-authenticated users to sign in to Microsoft Entra ID with any of their proxy addresses, in addition to UPN." A "Learn more" link is provided. The next section states: "The option below controls the tenant-wide feature setting in Home Realm Discovery policy." Another "Learn more" link is available. A checkbox labeled "Email as an alternate login ID" is checked and highlighted with a red border. At the bottom, there are "Save" and "Cancel" buttons, with the "Save" button also highlighted with a red border.

With the policy applied, it can take up to one hour to propagate and for users to be able to sign in using their alternate login ID.

PowerShell

! Note

This configuration option uses HRD policy. For more information, see [homeRealmDiscoveryPolicy resource type](#).

Once users with the *ProxyAddresses* attribute applied are synchronized to Microsoft Entra ID using Microsoft Entra Connect, you need to enable the feature for users to sign-in with email as an alternate login ID for your tenant. This feature tells the Microsoft Entra login servers to not only check the sign-in identifier against UPN values, but also against *ProxyAddresses* values for the email address.

1. Open a PowerShell session as an administrator, then install the *Microsoft.Graph* module using the `Install-Module` cmdlet:

```
PowerShell  
Install-Module Microsoft.Graph
```

For more information on installation, see [Install the Microsoft Graph PowerShell SDK](#).

2. Sign-in to your Microsoft Entra tenant using the `Connect-MgGraph` cmdlet:

```
PowerShell  
Connect-MgGraph -Scopes "Policy.ReadWrite.ApplicationConfiguration" -TenantId organizations
```

The command will ask you to authenticate using a web browser.

3. Check if a *HomeRealmDiscoveryPolicy* already exists in your tenant using the `Get-MgPolicyHomeRealmDiscoveryPolicy` cmdlet as follows:

```
PowerShell  
Get-MgPolicyHomeRealmDiscoveryPolicy
```

4. If there's no policy currently configured, the command returns nothing. If a policy is returned, skip this step and move on to the next step to update an existing policy.

To add the *HomeRealmDiscoveryPolicy* to the tenant, use the `New-MgPolicyHomeRealmDiscoveryPolicy` cmdlet and set the *AlternateIdLogin* attribute to "*Enabled*": *true* as shown in the following example:

```
PowerShell
```

```

$AzureADPolicyDefinition = @(
    @{
        "HomeRealmDiscoveryPolicy" = @{
            "AlternateIdLogin" = @{
                "Enabled" = $true
            }
        }
    } | ConvertTo-JSON -Compress
)

$AzureADPolicyParameters = @{
    Definition          = $AzureADPolicyDefinition
    DisplayName         = "BasicAutoAccelerationPolicy"
    AdditionalProperties = @{ IsOrganizationDefault = $true }
}

New-MgPolicyHomeRealmDiscoveryPolicy @AzureADPolicyParameters

```

When the policy has been successfully created, the command returns the policy ID, as shown in the following example output:

PowerShell
<pre> Definition DeletedDateTime Description DisplayName Id IsOrganizationDefault ----- ----- ----- {{"HomeRealmDiscoveryPolicy": {"AlternateIdLogin": {"Enabled": true}}}} BasicAutoAccelerationPolicy HRD_POLICY_ID True </pre>

5. If there's already a configured policy, check if the *AlternateIdLogin* attribute is enabled, as shown in the following example policy output:

PowerShell
<pre> Definition DeletedDateTime Description DisplayName Id IsOrganizationDefault ----- ----- ----- {{"HomeRealmDiscoveryPolicy": {"AlternateIdLogin": {"Enabled": true}}}} BasicAutoAccelerationPolicy HRD_POLICY_ID True </pre>

If the policy exists but the *AlternateIdLogin* attribute that isn't present or enabled, or if other attributes exist on the policy you wish to preserve, update the existing policy using

the `Update-MgPolicyHomeRealmDiscoveryPolicy` cmdlet.

ⓘ Important

When you update the policy, make sure you include any old settings and the new *AlternateIdLogin* attribute.

The following example adds the *AlternateIdLogin* attribute and preserves the *AllowCloudPasswordValidation* attribute that was previously set:

PowerShell

```
$AzureADPolicyDefinition = @(
    @{
        "HomeRealmDiscoveryPolicy" = @{
            "AllowCloudPasswordValidation" = $true
            "AlternateIdLogin" = @{
                "Enabled" = $true
            }
        }
    } | ConvertTo-JSON -Compress
)

$AzureADPolicyParameters = @{
    HomeRealmDiscoveryPolicyId = "HRD_POLICY_ID"
    Definition                 = $AzureADPolicyDefinition
    DisplayName                = "BasicAutoAccelerationPolicy"
    AdditionalProperties       = @{
        "IsOrganizationDefault" = $true
    }
}

Update-MgPolicyHomeRealmDiscoveryPolicy @AzureADPolicyParameters
```

Confirm that the updated policy shows your changes and that the *AlternateIdLogin* attribute is now enabled:

PowerShell

```
Get-MgPolicyHomeRealmDiscoveryPolicy
```

ⓘ Note

With the policy applied, it can take up to an hour to propagate and for users to be able to sign-in using email as an alternate login ID.

Removing policies

To remove an HRD policy, use the `Remove-MgPolicyHomeRealmDiscoveryPolicy` cmdlet:

PowerShell

```
Remove-MgPolicyHomeRealmDiscoveryPolicy -HomeRealmDiscoveryPolicyId  
"HRD_POLICY_ID"
```

Enable staged rollout to test user sign-in with an email address

! Note

This configuration option uses staged rollout policy. For more information, see [featureRolloutPolicy resource type](#).

Staged rollout policy allows tenant administrators to enable features for specific Microsoft Entra groups. It is recommended that tenant administrators use staged rollout to test user sign-in with an email address. When administrators are ready to deploy this feature to their entire tenant, they should use [HRD policy](#).

1. Open a PowerShell session as an administrator, then install the *Microsoft.Graph.Beta* module using the `Install-Module` cmdlet:

PowerShell

```
Install-Module Microsoft.Graph.Beta
```

If prompted, select Y to install NuGet or to install from an untrusted repository.

2. Sign in to your Microsoft Entra tenant using the `Connect-MgGraph` cmdlet:

PowerShell

```
Connect-MgGraph -Scopes "Directory.ReadWrite.All"
```

The command returns information about your account, environment, and tenant ID.

3. List all existing staged rollout policies using the following cmdlet:

PowerShell

Get-MgBetaPolicyFeatureRolloutPolicy

4. If there are no existing staged rollout policies for this feature, create a new staged rollout policy and take note of the policy ID:

PowerShell

```
$MgPolicyFeatureRolloutPolicy = @{
    Feature      = "EmailAsAlternateId"
    DisplayName  = "EmailAsAlternateId Rollout Policy"
    IsEnabled    = $true
}
New-MgBetaPolicyFeatureRolloutPolicy @MgPolicyFeatureRolloutPolicy
```

5. Find the directoryObject ID for the group to be added to the staged rollout policy. Note the value returned for the *Id* parameter, because it will be used in the next step.

PowerShell

```
Get-MgBetaGroup -Filter "DisplayName eq 'Name of group to be added to the
staged rollout policy'"
```

6. Add the group to the staged rollout policy as shown in the following example. Replace the value in the *-FeatureRolloutPolicyId* parameter with the value returned for the policy ID in step 4 and replace the value in the *-OdataId* parameter with the *Id* noted in step 5. It may take up to 1 hour before users in the group can sign in to Microsoft Entra ID with email as an alternate login ID.

PowerShell

```
New-MgBetaDirectoryFeatureRolloutPolicyApplyToByRef ` 
    -FeatureRolloutPolicyId "ROLLOUT_POLICY_ID" ` 
    -OdataId
"https://graph.microsoft.com/v1.0/directoryObjects/{GROUP_OBJECT_ID}"
```

For new members added to the group, it may take up to 24 hours before they can sign in to Microsoft Entra ID with email as an alternate login ID.

Removing groups

To remove a group from a staged rollout policy, run the following command:

PowerShell

```
Remove-MgBetaPolicyFeatureRolloutPolicyApplyToByRef -FeatureRolloutPolicyId  
"ROLLOUT_POLICY_ID" -DirectoryObjectId "GROUP_OBJECT_ID"
```

Removing policies

To remove a staged rollout policy, first disable the policy then remove it from the system:

PowerShell

```
Update-MgBetaPolicyFeatureRolloutPolicy -FeatureRolloutPolicyId  
"ROLLOUT_POLICY_ID" -IsEnabled:$false  
Remove-MgBetaPolicyFeatureRolloutPolicy -FeatureRolloutPolicyId  
"ROLLOUT_POLICY_ID"
```

Test user sign-in with an email address

To test that users can sign in with email, go to <https://myprofile.microsoft.com> and sign in with a non-UPN email, such as `balas@fabrikam.com`. The sign-in experience should look and feel the same as signing-in with the UPN.

Troubleshoot

If users have trouble signing in with their email address, review the following troubleshooting steps:

1. Make sure it's been at least 1 hour since email as an alternate login ID was enabled. If the user was recently added to a group for staged rollout policy, make sure it's been at least 24 hours since they were added to the group.
2. If using HRD policy, confirm that the Microsoft Entra ID *HomeRealmDiscoveryPolicy* has the *AlternateIdLogin* definition property set to "*Enabled*": *true* and the *IsOrganizationDefault* property set to *True*:

PowerShell

```
Get-MgBetaPolicyHomeRealmDiscoveryPolicy | Format-List *
```

If using staged rollout policy, confirm that the Microsoft Entra ID *FeatureRolloutPolicy* has the *IsEnabled* property set to *True*:

PowerShell

Get-MgBetaPolicyFeatureRolloutPolicy

3. Make sure the user account has their email address set in the *ProxyAddresses* attribute in Microsoft Entra ID.

Sign-in logs

Activity Details: Sign-ins X

Basic info Location Device info Authentication Details Conditional Access Report-only ...

Date
Request ID
Correlation ID
Authentication requirement
Status
Continuous access evaluation
Additional Details

Troubleshoot Event

User
Username
User ID
Sign-in identifier
User type
Sign-in identifier type proxyAddress
Cross tenant access type
Application

You can review the [sign-in logs in Microsoft Entra ID](#) for more information. Sign-ins with email as an alternate login ID will emit `proxyAddress` in the *Sign-in identifier type* field and the inputted username in the *Sign-in identifier* field.

Conflicting values between cloud-only and synced users

Within a tenant, a cloud-only user's UPN may take on the same value as another user's proxy address synced from the on-premises directory. In this scenario, with the feature enabled, the

cloud-only user will not be able to sign in with their UPN. Here are the steps for detecting instances of this issue.

1. Open a PowerShell session as an administrator, then install Microsoft Graph by using the [Install-Module](#) cmdlet:

```
PowerShell  
  
Install-Module Microsoft.Graph.Authentication
```

If prompted, select Y to install NuGet or to install from an untrusted repository.

2. Connect to Microsoft Graph:

```
PowerShell  
  
Connect-MgGraph -Scopes "User.Read.All"
```

3. Get affected users.

```
PowerShell  
  
# Get all users  
$allUsers = Get-MgUser -All  
  
# Get list of proxy addresses from all synced users  
$syncedProxyAddresses = $allUsers |  
    Where-Object {$_['ImmutableId']} |  
    Select-Object -ExpandProperty ProxyAddresses |  
    ForEach-Object {$_.Replace "smtp:", ""}  
  
# Get list of user principal names from all cloud-only users  
$cloudOnlyUserPrincipalNames = $allUsers |  
    Where-Object {!$_['ImmutableId']} |  
    Select-Object -ExpandProperty UserPrincipalName  
  
# Get intersection of two lists  
$duplicateValues = $syncedProxyAddresses |  
    Where-Object {$cloudOnlyUserPrincipalNames -Contains $_}
```

4. To output affected users:

```
PowerShell  
  
# Output affected synced users  
$allUsers |  
    Where-Object {$_['ImmutableId'] -And ($_.ProxyAddresses | Where-Object  
    {($duplicateValues | ForEach-Object {"smtp:$_"}) -Contains $_}).Length -GT 0}
```

```
|  
    Select-Object ObjectId, DisplayName, UserPrincipalName, ProxyAddresses,  
    ImmutableId, UserType  
  
# Output affected cloud-only users  
$allUsers |  
    Where-Object {!$_.ImmutableId -And $duplicateValues -Contains  
    $_.UserPrincipalName} |  
    Select-Object ObjectId, DisplayName, UserPrincipalName, ProxyAddresses,  
    ImmutableId, UserType
```

5. To output affected users to CSV:

PowerShell

```
# Output affected users to CSV  
$allUsers |  
    Where-Object {  
        ($_.ImmutableId -And ($_.ProxyAddresses | Where-Object  
        {($duplicateValues | ForEach-Object {"smtp:$_"}) -Contains $_}).Length -GT 0)  
        -Or  
        (!$.ImmutableId -And $duplicateValues -Contains  
        $_.UserPrincipalName)  
    } |  
    Select-Object ObjectId, DisplayName, UserPrincipalName,  
    @{n="ProxyAddresses"; e={$_.ProxyAddresses -Join ','}}, @{n="IsSyncedUser";  
e={$_.ImmutableId.Length -GT 0}}, UserType |  
    Export-Csv -Path .\AffectedUsers.csv -NoTypeInformation
```

Next steps

To learn more about hybrid identity, such as Microsoft Entra application proxy or Microsoft Entra Domain Services, see [Microsoft Entra hybrid identity for access and management of on-prem workloads](#).

For more information on hybrid identity operations, see [how password hash sync or pass-through authentication synchronization work](#).

Protecting authentication methods in Microsoft Entra ID

Article • 04/29/2025

(!) Note

The Microsoft managed value for Authenticator Lite will move from disabled to enabled on June 26th, 2023. All tenants left in the default state **Microsoft managed** will be enabled for the feature on June 26th.

Microsoft Entra ID adds and improves security features to better protect customers against increasing attacks. As new attack vectors become known, Microsoft Entra ID can respond by enabling protection by default to help customers stay ahead of emerging security threats.

For example, in response to increasing MFA fatigue attacks, Microsoft recommended ways for customers to [defend users](#). One recommendation to prevent users from accidental multifactor authentication (MFA) approvals is to enable [number matching](#). As a result, default behavior for number matching will be explicitly **Enabled** for all Microsoft Authenticator users. You can learn more about new security features like number matching in our blog post [Advanced Microsoft Authenticator security features are now generally available!](#).

There are two ways for protection of a security feature to be enabled by default:

- After a security feature is released, customers can use the Microsoft Entra admin center or Graph API to test and roll out the change on their own schedule. To help defend against new attack vectors, Microsoft Entra ID can enable protection of a security feature by default for all tenants on a certain date, and there won't be an option to disable protection. Microsoft schedules default protection far in advance to give customers time to prepare for the change. Customers can't opt out if Microsoft schedules protection by default.
- Protection can be **Microsoft managed**, which means Microsoft Entra ID can enable or disable protection based upon the current landscape of security threats. Customers can choose whether to allow Microsoft to manage the protection. They can change from **Microsoft managed** to explicitly make the protection **Enabled** or **Disabled** at any time.

(!) Note

Only a critical security feature will have protection enabled by default.

Default protection enabled by Microsoft Entra ID

Number matching is a good example of protection for an authentication method that is currently optional for push notifications in Microsoft Authenticator in all tenants. Customers could choose to enable number matching for push notifications in Microsoft Authenticator for users and groups, or they could leave it disabled. Number matching is already the default behavior for passwordless notifications in Microsoft Authenticator, and users can't opt out.

As MFA fatigue attacks rise, number matching becomes more critical to sign-in security. As a result, Microsoft will change the default behavior for push notifications in Microsoft Authenticator.

Microsoft managed settings

In addition to configuring Authentication methods policy settings to be either **Enabled** or **Disabled**, IT admins can configure some settings in the Authentication methods policy to be **Microsoft managed**. A setting that is configured as **Microsoft managed** allows Microsoft Entra ID to enable or disable the setting.

The option to let Microsoft Entra ID manage the setting is a convenient way for an organization to allow Microsoft to enable or disable a feature by default. Organizations can more easily improve their security posture by trusting Microsoft to manage when a feature should be enabled by default. By configuring a setting as **Microsoft managed** (named *default* in Graph APIs), IT admins can trust Microsoft to enable a security feature they haven't explicitly disabled.

For example, an admin can enable [location and application name](#) in push notifications to give users more context when they approve MFA requests with Microsoft Authenticator. The additional context can also be explicitly disabled, or set as **Microsoft managed**. Today, the **Microsoft managed** configuration for location and application name is **Disabled**, which effectively disables the option for any environment where an admin chooses to let Microsoft Entra ID manage the setting.

As the security threat landscape changes over time, Microsoft can change the **Microsoft managed** configuration for location and application name to **Enabled**. For customers who want to rely upon Microsoft to improve their security posture, setting security features to **Microsoft managed** is an easy way stay ahead of security threats. They can trust Microsoft to determine the best way to configure security settings based on the current threat landscape.

The following table lists each setting that can be set to Microsoft managed and whether that setting is enabled or disabled by default.

Setting	Configuration
Registration campaign	Enabled for text message and voice call users
Location in Microsoft Authenticator notifications	Disabled
Application name in Microsoft Authenticator notifications	Disabled
System-preferred MFA	Enabled
Authenticator Lite	Enabled
Report suspicious activity	Disabled

As threat vectors change, Microsoft Entra ID can announce default protection for a **Microsoft managed** setting in [release notes](#) and on commonly read forums like [Tech Community](#).

For more information, see our blog post [It's Time to Hang Up on Phone Transports for Authentication](#) which discusses moving away from using text message and voice calls. This change leads to default enablement for the registration campaign to help users set up Authenticator for modern authentication.

Next steps

[Authentication methods in Microsoft Entra ID - Microsoft Authenticator](#)

Enable combined security information registration in Microsoft Entra ID

Article • 03/04/2025

Before combined registration, users registered authentication methods for Microsoft Entra multifactor authentication and self-service password reset (SSPR) separately. Users were confused that similar methods were used for Microsoft Entra multifactor authentication and SSPR but they had to register for both features. Now, with combined registration, users can register once and get the benefits of both Microsoft Entra multifactor authentication and SSPR.

To help you understand the functionality and effects of the new experience, see the [Combined security information registration concepts](#).

The screenshot shows a wizard titled "Keep your account secure" with the sub-step "Method 1 of 2: App". It displays two options: "App" (selected, indicated by a blue circle with a white checkmark) and "Phone" (indicated by a grey circle with the number "2"). Below this, the "Microsoft Authenticator" step is shown with the sub-step "Set up your account". It includes an icon of a smartphone displaying a QR code, instructions to "If prompted, allow notifications. Then add an account, and select 'Work or school'.", and navigation buttons "Back" and "Next".

Conditional Access policies for combined registration

To secure when and how users register for Microsoft Entra multifactor authentication and self-service password reset, you can use user actions in Conditional Access policy. This functionality may be enabled in organizations that want users to register for Microsoft Entra multifactor authentication and SSPR from a central location, such as a trusted network location during HR onboarding.

Note

This policy applies only when a user accesses a combined registration page. This policy doesn't enforce MFA enrollment when a user accesses other applications.

You can create an MFA registration policy by using [Microsoft Entra ID Protection - Configure MFA Policy](#).

For more information about creating trusted locations in Conditional Access, see [What is the location condition in Microsoft Entra Conditional Access?](#).

Create a policy to require registration from a trusted location

Complete the following steps to create a policy that applies to all selected users that attempt to register using the combined registration experience, and requires users connected on a non-trusted network to either perform MFA or sign in using a Temporary Access Pass (TAP) to register for MFA or reset their password using SSPR:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Conditional Access Administrator](#).
2. Browse to **Protection > Conditional Access**.
3. Select **+ New policy**.
4. Enter a name for this policy, such as *Combined Security Info Registration on Trusted Networks*.
5. Under **Assignments**, select **Users**. Choose the users and groups you want this policy to apply to.

Warning

Users must be enabled for combined registration.

6. Under **Cloud apps or actions**, select **User actions**. Check **Register security information**, then select **Done**.

New ...

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies.

[Learn more](#)

Name *

device compliance test 

Assignments

Users 

[All users](#)

Cloud apps or actions 

[1 user action included](#)

Conditions 

[0 conditions selected](#)

Access controls

Grant 

[0 controls selected](#)

Session 

[0 controls selected](#)

Enable policy

Report-only On Off

Create

7. Under **Conditions > Locations**, configure the following options:
 - a. Configure **Yes**.
 - b. Include **Any location**.
 - c. Exclude **All trusted locations**.
8. Under **Access controls > Grant**, choose **Require multifactor authentication**, then **Select**.
9. Set **Enable policy** to **On**.

10. To finalize the policy, select **Create**.

Next steps

If you need help, see [troubleshoot combined security info registration](#) or learn [What is the location condition in Microsoft Entra Conditional Access?](#)

Review how you can [enable self-service password reset](#) and [enable Microsoft Entra multifactor authentication](#) in your tenant.

If needed, learn how to [force users to re-register authentication methods](#).

Feedback

Was this page helpful?



[Provide product feedback](#) ↗

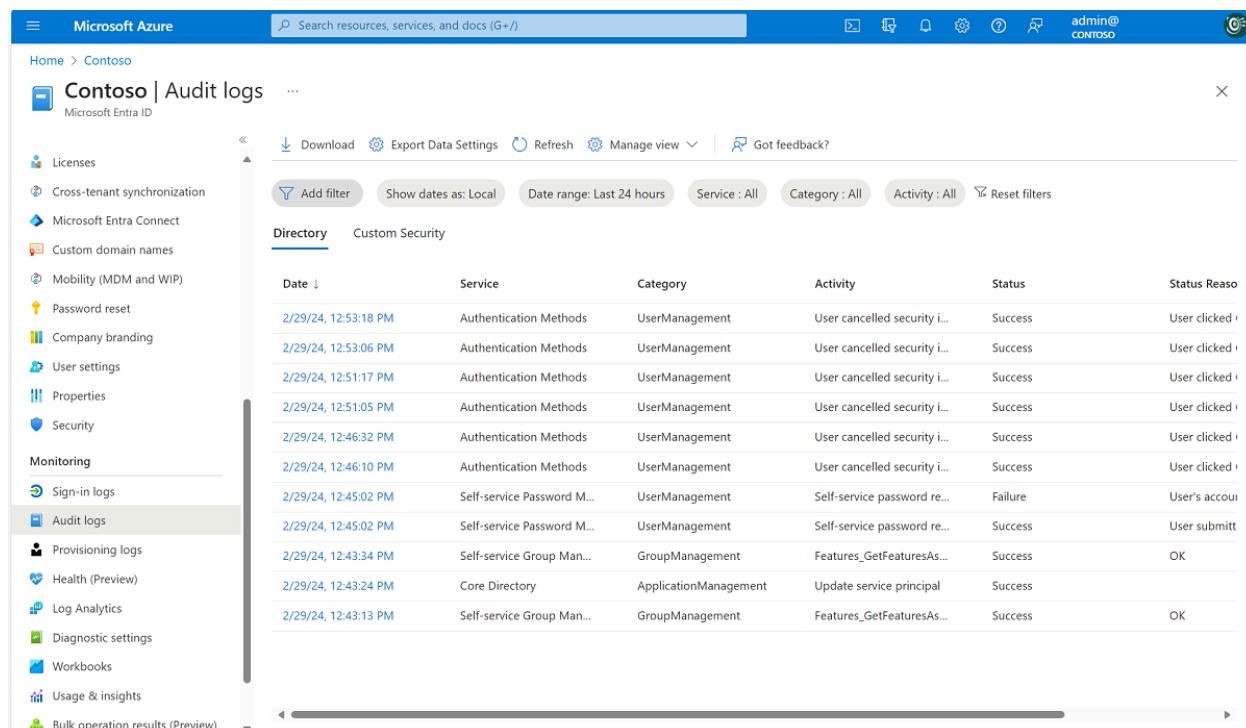
Troubleshooting combined security information registration

Article • 02/13/2025

The information in this article is meant to guide admins who are troubleshooting issues reported by users of the combined registration experience.

Audit logs

The events logged for combined registration are in the Authentication Methods service in the Microsoft Entra audit logs.



The screenshot shows the Microsoft Azure portal with the search bar at the top. The URL in the address bar is 'https://contoso.aad.portal.azure.com/#blade/Microsoft_Azure_AuditLogs/LogsBlade'. The left sidebar is titled 'Contoso | Audit logs' and includes sections for Licenses, Cross-tenant synchronization, Microsoft Entra Connect, Custom domain names, Mobility (MDM and WiP), Password reset, Company branding, User settings, Properties, Security, Monitoring, Sign-in logs, Audit logs (which is selected), Provisioning logs, Health (Preview), Log Analytics, Diagnostic settings, Workbooks, Usage & insights, and Bulk operation results (Preview). The main content area displays a table of audit events. The table has columns for Date, Service, Category, Activity, Status, and Status Reason. The data shows several successful events for Authentication Methods and Self-service Password Management, along with one failure for Self-service Password Management due to a user account lockout.

Date	Service	Category	Activity	Status	Status Reason
2/29/24, 12:53:18 PM	Authentication Methods	UserManagement	User cancelled security i...	Success	User clicked ↗
2/29/24, 12:53:06 PM	Authentication Methods	UserManagement	User cancelled security i...	Success	User clicked ↗
2/29/24, 12:51:17 PM	Authentication Methods	UserManagement	User cancelled security i...	Success	User clicked ↗
2/29/24, 12:51:05 PM	Authentication Methods	UserManagement	User cancelled security i...	Success	User clicked ↗
2/29/24, 12:46:32 PM	Authentication Methods	UserManagement	User cancelled security i...	Success	User clicked ↗
2/29/24, 12:46:10 PM	Authentication Methods	UserManagement	User cancelled security i...	Success	User clicked ↗
2/29/24, 12:45:02 PM	Self-service Password M...	UserManagement	Self-service password re...	Failure	User's account lockout ↗
2/29/24, 12:45:02 PM	Self-service Password M...	UserManagement	Self-service password re...	Success	User submitted ↗
2/29/24, 12:43:34 PM	Self-service Group Man...	GroupManagement	Features_GetFeaturesAs...	Success	OK ↗
2/29/24, 12:43:24 PM	Core Directory	ApplicationManagement	Update service principal	Success	
2/29/24, 12:43:13 PM	Self-service Group Man...	GroupManagement	Features_GetFeaturesAs...	Success	OK ↗

The following table lists all audit events generated by combined registration:

[Expand table](#)

Activity	Status	Reason	Description
User registered all required security info	Success	User registered all required security info.	This event occurs when a user has successfully completed registration.
User registered all	Failure	User canceled security info registration.	This event occurs when a user cancels registration from interrupt mode.

Activity	Status	Reason	Description
required security info			
User registered security info	Success	User registered <i>method</i> .	This event occurs when a user registers an individual method. <i>Method</i> can be Authenticator app, Phone, Email, Security questions, App password, Alternate phone, and so on.
User reviewed security info	Success	User successfully reviewed security info.	This event occurs when a user selects Looks good on the security info review page.
User reviewed security info	Failure	User failed to review security info.	This event occurs when a user selects Looks good on the security info review page but something fails on the backend.
User deleted security info	Success	User deleted <i>method</i> .	This event occurs when a user deletes an individual method. <i>Method</i> can be Authenticator app, Phone, Email, Security questions, App password, Alternate phone, and so on.
User deleted security info	Failure	User failed to delete <i>method</i> .	This event occurs when a user tries to delete a method but the attempt fails for some reason. <i>Method</i> can be Authenticator app, Phone, Email, Security questions, App password, Alternate phone, and so on.
User changed default security info	Success	User changed the default security info for <i>method</i> .	This event occurs when a user changes the default method. <i>Method</i> can be Authenticator app notification, A code from my authenticator app or token, Call +X XXXXXXXXXX, Text a code to +X XXXXXXXXXX, and so on.
User changed default security info	Failure	User failed to change the default security info for <i>method</i> .	This event occurs when a user tries to change the default method but the attempt fails for some reason. <i>Method</i> can be Authenticator app notification, A code from my authenticator app or token, Call +X XXXXXXXXXX, Text a code to +X XXXXXXXXXX, and so on.

Troubleshooting interrupt mode

[Expand table](#)

Symptom	Troubleshooting steps
I'm not seeing the methods I expected to see.	<ol style="list-style-type: none"> 1. Check if the user has a Microsoft Entra admin role. If yes, view the SSPR admin policy differences. 2. Determine whether the user is being interrupted because of multifactor authentication (MFA) registration enforcement or SSPR registration enforcement. See the flowchart under "Combined registration modes" to determine which methods should be shown. 3. Determine how recently the MFA or SSPR policy was changed. If the change was recent, it might take some time for the updated policy to propagate.

Troubleshooting manage mode

[Expand table](#)

Symptom	Troubleshooting steps
I don't have the option to add a particular method.	<ol style="list-style-type: none"> 1. Determine whether the method is enabled for MFA or for SSPR. 2. If the method is enabled, save the policies again and wait 1-2 hours before testing again. 3. If the method is enabled, ensure that the user hasn't already set up the maximum number of that method that they're allowed to set up.

How to require user to re-register for multifactor authentication

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Users**, and select the user whom you want to re-register MFA.
3. Click **Authentication methods**, and click **Require re-register for multifactor authentication**.
4. Click **OK** to confirm.

Next steps

- Learn more about combined registration for self-service password reset and Microsoft Entra multifactor authentication

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

How to run a registration campaign to set up Microsoft Authenticator

Article • 03/04/2025

You can nudge users to set up Microsoft Authenticator during sign-in. Users go through their regular sign-in, perform multifactor authentication as usual, and then get prompted to set up Microsoft Authenticator. You can include or exclude users or groups to control who gets nudged to set up the app. This allows targeted campaigns to move users from less secure authentication methods to Authenticator.

You can also define how many days a user can postpone, or "snooze," the nudge. If a user taps **Skip for now** to postpone the app setup, they get nudged again on the next MFA attempt after the snooze duration has elapsed. You can decide whether the user can snooze indefinitely or up to three times (after which registration is required).

ⓘ Note

As users go through their regular sign-in, Conditional Access policies that govern security info registration apply before the user is prompted to set up Authenticator. For example, if a Conditional Access policy requires security info updates can only occur on an internal network, then users won't be prompted to set up Authenticator unless they are on the internal network.

Prerequisites

- Your organization must have enabled Microsoft Entra multifactor authentication. Every edition of Microsoft Entra ID includes Microsoft Entra multifactor authentication. No other license is needed for a registration campaign.
- Users can't have already set up the Authenticator app for push notifications on their account.
- Admins need to enable users for the Authenticator app using one of these policies:
 - MFA Registration Policy: Users will need to be enabled for **Notification through mobile app**.
 - Authentication Methods Policy: Users will need to be enabled for the Authenticator app and the Authentication mode set to **Any or Push**. If the policy is set to **Passwordless**, the user won't be eligible for the nudge. For more information about how to set the Authentication mode, see [Enable passwordless sign-in with Microsoft Authenticator](#).

User experience

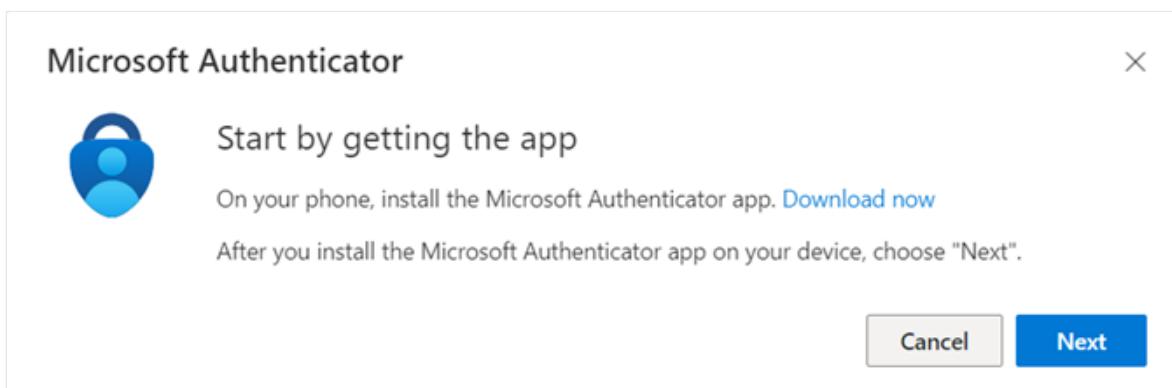
1. First, you need to successfully authenticate using Microsoft Entra multifactor authentication (MFA).
2. If you've enabled for Authenticator push notifications and don't have it already set up, you'll get prompted to set up Authenticator to improve your sign-in experience.

 **Note**

Other security features, such as passwordless passkey, self-service password reset or security defaults, might also prompt you for setup.



3. Tap **Next** and step through the Authenticator app setup.
4. First download the app.



- a. See how to set up the Authenticator app.

Microsoft Authenticator

X



Set up your account

If prompted, allow notifications. Then add an account, and select "Work or school".

Back

Next

- b. Scan the QR Code.

Microsoft Authenticator

X

Scan the QR code

Use the Microsoft Authenticator app to scan the QR code. This will connect the Microsoft Authenticator app with your account.

After you scan the QR code, choose "Next".



Can't scan image?

- c. Verify your identity.



cjensen@contoso.com

Verify your identity



Use a certificate or smart card



Approve a request on my Microsoft
Authenticator app



Call +X XXX-XXX-XX63

[More information](#)

Are your verification methods current? Check at

<https://aka.ms/mfasetup>

Cancel

d. Approve the test notification on your device.



cjensen@contoso.com

Approve sign in request



Open your Authenticator app, and enter the
number shown to sign in.

22

No numbers in your app? Make sure to upgrade to
the latest version.

e. Authenticator app is now successfully set up.

The screenshot shows the Microsoft My Sign-Ins interface. On the left, a sidebar lists options: Overview, Security info (which is selected and highlighted in blue), Devices, Password, Organizations, Settings & Privacy, and Recent activity. The main content area is titled "Security info" and contains a sub-header: "These are the methods you use to sign into your account or reset your password." It shows the "Default sign-in method" as "Phone - call +1 800-555-0199" with a "Change" link. Below this is a table titled "Add sign-in method" with three rows: "Phone" (number +1 800-555-0199, "Change" and "Delete" links), "Password" (last updated 3 months ago, "Change" link), and "Microsoft Authenticator" (device iPhone 12, "Delete" link). At the bottom is a link "Lost device? Sign out everywhere".

5. If you don't want to install the Authenticator app, you can tap **Skip for now** to snooze the prompt for up to 14 days, which can be set by an admin. Users with free and trial subscriptions can snooze the prompt up to three times.



Enable the registration campaign policy using the Microsoft Entra admin center

To enable a registration campaign in the Microsoft Entra admin center, complete the following steps:

1. Sign in to the Microsoft Entra admin center as at least an **Authentication Policy Administrator**.

2. Browse to **Protection > Authentication methods > Registration campaign** and click **Edit**.

3. For **State**:

- Select **Enabled** to enable the registration campaign for all users.
- Select **Microsoft managed** to enable the registration campaign only for voice call or text message users. The **Microsoft managed** setting allows Microsoft to set the default value. For more information, see [Protecting authentication methods in Microsoft Entra ID](#).

If the registration campaign state is set to **Enabled** or **Microsoft managed**, you can configure the experience for end users by using **Limited number of snoozes**:

- If **Limited number of snoozes** is Enabled, users can skip the interrupt prompt 3 times, after which they're forced to register Authenticator.
- If **Limited number of snoozes** is Disabled, users can snooze an unlimited number of times and avoid registering Authenticator.

Days allowed to snooze sets the period between two successive interrupt prompts. For example, if it's set to 3 days, users who skipped registration don't get prompted again until after 3 days.

The screenshot shows the Microsoft Entra admin center interface. The left sidebar has sections like Home, Favorites, Identity, Protection (which is expanded), and others. The main area shows the 'Authentication methods | Registration campaign' page for 'Contoso - Microsoft Entra ID Security'. The 'Manage' section includes 'Policies', 'Registration campaign' (which is selected and highlighted in grey), 'Password protection', and 'Authentication strengths'. The 'Settings' section shows the following configuration:

Setting	Value
State	Microsoft managed
Days allowed to snooze	1 day
Limited number of snoozes	Enabled
Excluded users and groups	1 Group

Below this, under 'Authentication method', there is one entry:

Method	Included users and groups
Microsoft Authenticator	All users

4. Select any users or groups to exclude from the registration campaign, and then click **Save**.

Enable the registration campaign policy using Graph Explorer

In addition to using the Microsoft Entra admin center, you can also enable the registration campaign policy using Graph Explorer. To enable the registration campaign

policy, you must use the Authentication Methods Policy using Graph APIs. Those assigned at least the [Authentication Policy Administrator](#) role can update the policy.

To configure the policy using Graph Explorer:

1. Sign in to Graph Explorer and ensure you've consented to the **Policy.Read.All** and **Policy.ReadWrite.AuthenticationMethod** permissions.

To open the Permissions panel:

The screenshot shows the Microsoft Graph Explorer interface. At the top, there are dropdown menus for 'GET' and 'beta', and a URL bar containing 'https://graph.microsoft.com/beta/policies/authenticationmethodspolicy'. On the right, a blue 'Run query' button is visible. Below the URL bar, there are four tabs: 'Request body', 'Request headers', 'Modify permissions (Preview)', and 'Access token'. The 'Modify permissions (Preview)' tab is highlighted with a yellow background. A note below the tabs states: 'Permissions for the query are missing on this tab. Open the permissions panel to see the full list of Microsoft Graph permissions and select the permission(s) you want and consent to them from there.' A yellow box highlights the word 'permissions' in this note.

2. Retrieve the Authentication methods policy:

The screenshot shows the Microsoft Graph Explorer interface. On the left, a 'JSON' button is selected. In the main area, a 'GET' method is chosen, and the URL 'https://graph.microsoft.com/v1.0/policies/authenticationmethodspolicy' is displayed.

3. Update the registrationEnforcement and authenticationMethodsRegistrationCampaign section of the policy to enable the nudge on a user or group.

```
"registrationEnforcement": {  
    "authenticationMethodsRegistrationCampaign": {  
        "snoozeDurationInDays": 1,  
        "enforceRegistrationAfterAllowedSnoozes": true,  
        "state": "default",  
        "excludeTargets": [],  
        "includeTargets": [  
            {  
                "id": "all_users",  
                "targetType": "group",  
                "targetedAuthenticationMethod": "microsoftAuthenticator"  
            }  
        ]  
    },  
},
```

To update the policy, perform a PATCH on the Authentication Methods Policy with only the updated registrationEnforcement section:

The screenshot shows the Microsoft Graph Explorer interface. On the left, a 'JSON' button is selected. In the main area, a 'PATCH' method is chosen, and the URL 'https://graph.microsoft.com/v1.0/policies/authenticationmethodspolicy' is displayed.

PATCH

<https://graph.microsoft.com/v1.0/policies/authenticationmethodspolicy>

The following table lists **authenticationMethodsRegistrationCampaign** properties.

[+] [Expand table](#)

Name	Possible values	Description
snoozeDurationInDays	Range: 0 - 14	Defines the number of days before the user is nudged again. If the value is 0, the user is nudged during every MFA attempt. Default: 1 day
enforceRegistrationAfterAllowedSnoozes	"true" "false"	Dictates whether a user is required to perform setup after 3 snoozes. If true, user is required to register. If false, user can snooze indefinitely. Default: true
state	"enabled" "disabled" "default"	Allows you to enable or disable the feature. Default value is used when the configuration hasn't been explicitly set and will use Microsoft Entra ID default value for this setting. The default state is enabled for voice call and text message users in all tenants. Change state to enabled (for all users) or disabled as needed.
excludeTargets	N/A	Allows you to exclude different users and groups that you want omitted from the feature. If a user is in a group that is excluded and a group that is included, the user will be excluded from the feature.
includeTargets	N/A	Allows you to include different users and groups that you want the feature to target.

The following table lists **includeTargets** properties.

[+] [Expand table](#)

Name	Possible values	Description
targetType	"user" "group"	The kind of entity targeted.
ID	A guid identifier	The ID of the user or group targeted.
targetedAuthenticationMethod	"microsoftAuthenticator"	The authentication method user is prompted to register. The only permissible value is "microsoftAuthenticator".

The following table lists **excludeTargets** properties.

[+] Expand table

Name	Possible values	Description
targetType	"user" "group"	The kind of entity targeted.
ID	A string	The ID of the user or group targeted.

Examples

Here are a few sample JSONs you can use to get started!

- Include all users

If you want to include ALL users in your tenant, update the following JSON example with the relevant GUIDs of your users and groups. Then paste it in Graph Explorer and run **PATCH** on the endpoint.

JSON

```
{
  "registrationEnforcement": {
    "authenticationMethodsRegistrationCampaign": {
      "snoozeDurationInDays": 1,
      "enforceRegistrationAfterAllowedSnoozes": true,
      "state": "enabled",
      "excludeTargets": [],
      "includeTargets": [
        {
          "id": "all_users",
          "targetType": "group",
          "targetedAuthenticationMethod": "microsoftAuthenticator"
        }
      ]
    }
  }
}
```

```
        ]
    }
}
```

- Include specific users or groups of users

If you want to include certain users or groups in your tenant, update the following JSON example with the relevant GUIDs of your users and groups. Then paste the JSON in Graph Explorer and run **PATCH** on the endpoint.

JSON

```
{
  "registrationEnforcement": {
    "authenticationMethodsRegistrationCampaign": {
      "snoozeDurationInDays": 1,
      "enforceRegistrationAfterAllowedSnoozes": true,
      "state": "enabled",
      "excludeTargets": [],
      "includeTargets": [
        {
          "id": "*****PLEASE ENTER GUID*****",
          "targetType": "group",
          "targetedAuthenticationMethod": "microsoftAuthenticator"
        },
        {
          "id": "*****PLEASE ENTER GUID*****",
          "targetType": "user",
          "targetedAuthenticationMethod": "microsoftAuthenticator"
        }
      ]
    }
  }
}
```

- Include and exclude specific users or groups

If you want to include AND exclude certain users or groups in your tenant, update the following JSON example with the relevant GUIDs of your users and groups. Then paste it in Graph Explorer and run **PATCH** on the endpoint.

JSON

```
{
  "registrationEnforcement": {
    "authenticationMethodsRegistrationCampaign": {
      "snoozeDurationInDays": 1,
```

```
"enforceRegistrationAfterAllowedSnoozes": true,
"state": "enabled",
"excludeTargets": [
    {
        "id": "*****PLEASE ENTER GUID*****",
        "targetType": "group"
    },
    {
        "id": "*****PLEASE ENTER GUID*****",
        "targetType": "user"
    }
],
"includeTargets": [
    {
        "id": "*****PLEASE ENTER GUID*****",
        "targetType": "group",
        "targetedAuthenticationMethod": "microsoftAuthenticator"
    },
    {
        "id": "*****PLEASE ENTER GUID*****",
        "targetType": "user",
        "targetedAuthenticationMethod": "microsoftAuthenticator"
    }
]
}
```

Identify the GUIDs of users to insert in the JSONs

1. Sign in to the Microsoft Entra admin center [↗](#) as at least an [Authentication Policy Administrator](#).
2. In the **Manage** blade, tap **Users**.
3. In the **Users** page, identify the specific user you want to target.
4. When you tap the specific user, you'll see their **Object ID**, which is the user's GUID.

The screenshot shows a user profile for 'Chris'. It includes the name 'Chris', email 'chris@contoso.com', a green circular icon with 'CH', creation time '5/24/2021, 3:26:26 PM', and an 'Identity' section with fields: Name (Chris), User Principal Name (chris@contoso.com), and Object ID (f7976cb3-9d83-432c-b268-402375720431). A blue arrow points to the Object ID field.

Identify the GUIDs of groups to insert in the JSONs

1. Sign in to the Microsoft Entra admin center [↗](#) as at least an [Authentication Policy Administrator](#).
2. In the Manage blade, tap **Groups**.
3. In the Groups page, identify the specific group you want to target.
4. Tap the group and get the **Object ID**.

The screenshot shows a group named 'Authenticator Nudge Group' (AN). It displays the following details:

- Membership type: Assigned
- Source: Cloud
- Type: Security
- Object Id: 9e55fe73-aaf8-47fc-b15d-9c4ae05203c7
- Creation date: 5/24/2021, 5:40:19 PM

A blue arrow points to the Object Id field.

Limitations

The nudge won't appear on mobile devices that run Android or iOS.

Frequently asked questions

Is registration campaign available for MFA Server?

No, the registration campaign is available only for users using Microsoft Entra multifactor authentication.

Can users be nudged within an application?

Yes, we support embedded browser views in certain applications. We don't nudge users in out of the box experiences or in browser views embedded in Windows settings.

Can users be nudged on a mobile device?

The registration campaign isn't available on mobile devices.

How long does the campaign run for?

You can enable the campaign for as long as you like. Whenever you want to be done running the campaign, use the admin center or APIs to disable the campaign.

Can each group of users have a different snooze duration?

No. The snooze duration for the prompt is a tenant-wide setting and applies to all groups in scope.

Can users be nudged to set up passwordless phone sign-in?

The feature aims to empower admins to get users set up with MFA using the Authenticator app and not passwordless phone sign-in.

Will a user who signs in with a 3rd party authenticator app see the nudge?

Yes. If a user is enabled for the registration campaign and doesn't have Microsoft Authenticator set up for push notifications, the user is nudged to set up Authenticator.

Will a user who has Authenticator set up only for TOTP codes see the nudge?

Yes. If a user is enabled for the registration campaign and Authenticator app isn't set up for push notifications, the user is nudged to set up push notification with Authenticator.

If a user just went through MFA registration, are they nudged in the same sign-in session?

No. To provide a good user experience, users won't be nudged to set up the Authenticator in the same session that they registered other authentication methods.

Can I nudge my users to register another authentication method?

No. The feature, for now, aims to nudge users to set up the Authenticator app only.

Is there a way for me to hide the snooze option and force my users to setup the Authenticator app?

Set the **Limited number of snoozes** to **Enabled** such that users can postpone the app setup up to three times, after which setup is required.

Will I be able to nudge my users if I am not using Microsoft Entra multifactor authentication?

No. The nudge only works for users who are doing MFA using the Microsoft Entra multifactor authentication service.

Will Guest/B2B users in my tenant be nudged?

Yes. If they have been scoped for the nudge using the policy.

What if the user closes the browser?

It's the same as snoozing. If setup is required for a user after they snoozed three times, the user is prompted the next time they sign in.

Why don't some users see a nudge when there is a Conditional Access policy for "Register security information"?

A nudge won't appear if a user is in scope for a Conditional Access policy that blocks access to the **Register security information** page.

Do users see a nudge when there is a terms of use (ToU) screen presented to the user during sign-in?

A nudge won't appear if a user is presented with the [terms of use \(ToU\)](#) screen during sign-in.

Do users see a nudge when Conditional Access custom controls are applicable to the sign-in?

A nudge won't appear if a user is redirected during sign-in due to [Conditional Access custom controls](#) settings.

Are there any plans to discontinue SMS and Voice as methods usable for MFA?

No, there are no such plans.

Next steps

[Enable passwordless sign-in with Microsoft Authenticator](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Customize the user experience for Microsoft Entra self-service password reset

Article • 04/27/2025

Self-service password reset (SSPR) gives users in Microsoft Entra ID the ability to change or reset their password, with no administrator or helpdesk involvement. If a user's account is locked or they forget their password, they can follow prompts to unblock themselves and get back to work. This ability reduces helpdesk calls and loss of productivity when a user can't sign in to their device or an application.

To improve the SSPR experience for users, you can customize the look and feel of the password reset page, email notifications, or sign-in pages. Customization options help to make it clear to users that they're in the right place and give them confidence that they're accessing company resources.

This article shows you how to customize the SSPR e-mail link for users, company branding, and the Active Directory Federation Services (AD FS) sign-in page link. Anyone who is assigned the [Authentication Policy Administrator](#) role can customize most of these options.

Customize the Contact your administrator link

To help users reach out for assistance with SSPR, a **Contact your administrator** link is shown in the password reset portal. If a user selects this link, it does one of two things:

- If this contact link is left in the default state, an email is sent to your administrators and asks them to help in changing the user's password. The following sample e-mail shows this default e-mail message:



ms@.com

To: IL

Cc: LH; +3 others



Thu 2/29/2024 5:45 PM

Request to reset user's password

The following user in your organization has requested a password reset be performed for their account:

- A@.com
- First Name: A
- Last Name: W

Consider contacting this user to validate this request is authentic before continuing.

If you have determined that this is a valid request, use your service's admin portal (Office 365, Windows Intune, Windows Azure, etc.) to reset the password for this user.

Want to let your users reset their own passwords? [Check out how you can enable password reset for users in your organization with just a few clicks.](#)

Sincerely,
Contoso

Microsoft Corporation | One Microsoft Way Redmond, WA 98052-6399

This message was sent from an unmonitored email address. Please do not reply to this message.

[Privacy](#) | [Legal](#)

- If this contact link is customized, it sends the user to a webpage or sends an email to the address specified by the administrator for assistance.
 - If you customize this link, we recommend that you set it to something that users are already familiar with for support.

Warning

If you customize this setting with an email address and account that needs a password reset, the user might be unable to ask for assistance.

Default email behavior

The default contact email is sent to recipients in the following order:

1. If the Helpdesk Administrator role or Password Administrator role is assigned, administrators with these roles are notified.
2. If no Helpdesk Administrator or Password Administrator is assigned, then administrators with the User Administrator role are notified.
3. If none of the previous roles are assigned, then the Global Administrators are notified.

In all cases, a maximum of 100 recipients are notified.

To find out more about the different administrator roles and how to assign them, see [Assign administrator roles in Microsoft Entra ID](#).

Disable Contact your administrator emails

If your organization doesn't want to notify administrators about password reset requests, you can use the following configuration options:

- Customize the helpdesk link to provide a web URL address that users can use to get assistance. This option is under **Password Reset > Customization > Custom helpdesk email or URL**.
- Enable self-service password reset for all users. This option is under **Password Reset > Properties**. If you don't want users to reset their own passwords, you can scope access to an empty group. *We don't recommend this option.*

Customize the sign-in page and access panel

You can customize the sign-in page, for example, to add a logo that appears along with the image that fits your company branding. For more information on how to configure company branding, see [Add company branding to your sign-in page in Microsoft Entra ID](#).

The graphics that you choose appear in the following circumstances:

- After a user enters their username.
- If the user accesses the customized URL:
 - By passing the `whr` parameter to the password reset page, like
`https://login.microsoftonline.com/?whr=contoso.com`.
 - By passing the `username` parameter to the password reset page, like
`https://login.microsoftonline.com/?username=admin@contoso.com`.

SSPR honors browser language settings. When there's a customization for browser language, the page appears in the browser language customization. Otherwise, it falls to the default locale customization.

Directory name

To make things look more personalized, you can change the organization name in the portal and in the automated communications.

To change the directory name attribute in the Microsoft Entra admin center:

Important

Microsoft recommends that you use roles with the fewest permissions. This practice helps improve security for your organization. Global Administrator is a highly privileged role that should be limited to emergency scenarios or when you can't use an existing role.

1. Sign in to the [Microsoft Entra admin center](#) as a **Global Administrator**.
2. Browse to **Entra ID > Overview > Properties**.
3. Update the name.
4. Select **Save**.

This organization name option is the most visible in automated emails, as in the following examples:

- **Email display name:** For example, *Microsoft on behalf of CONTOSO demo*
- **Email subject line:** For example, *CONTOSO demo account email verification code*

Customize the AD FS sign-in page

If you use AD FS for user sign-in events, you can add a link to the sign-in page by using the guidance in the article to [Add a sign-in page description](#).

Provide users with a link to the page for them to enter the SSPR workflow, such as

`https://passwordreset.microsoftonline.com`. To add a link to the AD FS sign-in page, use the following command on your AD FS server:

PowerShell

```
Set-ADFSGlobalWebContent -SigninPageDescriptionText "<p><a href='https://passwordreset.microsoftonline.com' target='_blank'>Can't access your account?</a></p>"
```

Related content

- To understand the use of SSPR in your environment, see [Reporting options for Microsoft Entra password management](#).
- If you or users have problems with SSPR, see [Troubleshoot self-service password reset](#).

Prepopulate user authentication contact information for Microsoft Entra self-service password reset (SSPR)

Article • 04/27/2025

To use Microsoft Entra self-service password reset (SSPR), authentication information for a user must be present. Most organizations have users register their authentication data themselves while collecting information for multifactor authentication.

Some organizations prefer to bootstrap this process through synchronization of authentication data that already exists in Active Directory Domain Services. This synchronized data is made available to Microsoft Entra ID and SSPR without requiring user interaction. When users need to change or reset their password, they can do so even if they haven't previously registered their contact information.

You can prepopulate authentication contact information if you meet the following requirements:

- You formatted the data in your on-premises directory properly.
- You configured [Microsoft Entra Connect](#) for your Microsoft Entra tenant.

Phone numbers must be in the format *+CountryCode PhoneNumber*, such as *+1 4251234567*.

Further restrictions are:

- There must be a space between the country code and the phone number.
- Password reset doesn't support phone extensions. Even in the *+1 4251234567X12345* format, extensions are removed before the call is placed.

Fields populated

If you use the default settings in Microsoft Entra Connect, the following mappings are made to populate authentication contact information for SSPR.

 [Expand table](#)

On-premises Active Directory	Microsoft Entra ID
telephoneNumber	Office phone
mobile	Mobile phone

After a user verifies their mobile phone number, the **Phone** field under **Authentication contact info** in Microsoft Entra ID is also populated with that number.

Authentication contact information

On the **Authentication methods** page for a Microsoft Entra user in the Microsoft Entra admin center, users who are assigned at least the [Privileged Authentication Administrator](#) role can manually set the authentication contact information for anyone. You can review existing methods under the **Usable authentication methods** section or by selecting **+Add authentication method**.

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation sidebar with options like Home, Contoso, Users, Bala Sandhu (selected), Diagnose and solve problems, Manage (selected), Profile, Assigned roles, Administrative units, Groups, Applications, Licenses, Devices, Azure role assignments, and Authentication methods (which is highlighted with a red box). Below this is an Activity section with Sign-ins and Audit logs. The main content area shows a user profile for Bala Sandhu with an 'Add authentication method' button. A tooltip says 'Want to switch back to the old user authentication experience?'. A list titled 'Usable authentication methods' contains 'Authentication method' and 'Phone number', both of which are highlighted with a red box. To the right, an 'Add authentication method' dialog box is open, showing a dropdown for 'Choose method' set to 'Email', a text input for 'Email address *' containing 'balas@contoso.com' with a green checkmark, and a blue 'Add' button.

The following considerations apply for this authentication contact information:

- If the **Phone** field is populated and **Mobile phone** is enabled in the SSPR policy, the user sees that number on the password reset registration page and during the password reset workflow.
- If the **Email** field is populated and **Email** is enabled in the SSPR policy, the user sees that email on the password reset registration page and during the password reset workflow.

Security questions and answers

The security questions and answers are stored securely in your Microsoft Entra tenant and are accessible to users only via the My Security-Info [combined registration experience](#). Administrators can't see, set, or modify the contents of another user's questions and answers.

What happens when a user registers?

When a user registers, the registration page sets the following fields:

- Authentication Phone
- Authentication Email
- Security Questions and Answers

If you provided a value for **Mobile phone** or **Alternate email**, users can immediately use those values to reset their passwords, even if they haven't registered for the service.

Users also see those values when they register for the first time, and they can modify them if they want to. After they successfully register, these values are persisted in the **Authentication Phone** and **Authentication Email** fields, respectively.

Set and read the authentication data through PowerShell

You can set the following fields through PowerShell:

- Alternate email
- Mobile phone
- Office phone
 - Can be set only if you're not synchronizing with an on-premises directory.

You can use [Microsoft Graph PowerShell](#) to interact with Microsoft Entra ID. You can also use the [Microsoft Graph REST API](#) for managing authentication methods.

Use Microsoft Graph PowerShell

To get started, [download and install the Microsoft Graph PowerShell module](#).

To quickly install from recent versions of PowerShell that support `Install-Module`, run the following commands. The first line checks to see if the module is already installed.

```
PowerShell  
  
Get-Module Microsoft.Graph  
Install-Module Microsoft.Graph  
Select-MgProfile -Name "beta"  
Connect-MgGraph -Scopes "User.ReadWrite.All"
```

After the module is installed, use the following steps to configure each field.

Set the authentication data with Microsoft Graph PowerShell

PowerShell

```
Connect-MgGraph -Scopes "User.ReadWrite.All"

Update-MgUser -UserId 'user@domain.com' -otherMails @("emails@domain.com")
Update-MgUser -UserId 'user@domain.com' -mobilePhone "+1 4251234567"
Update-MgUser -UserId 'user@domain.com' -businessPhones "+1 4252345678"

Update-MgUser -UserId 'user@domain.com' -otherMails @("emails@domain.com") -
mobilePhone "+1 4251234567" -businessPhones "+1 4252345678"
```

Read the authentication data with Microsoft Graph PowerShell

PowerShell

```
Connect-MgGraph -Scopes "User.Read.All"

Get-MgUser -UserId 'user@domain.com' | select otherMails
Get-MgUser -UserId 'user@domain.com' | select mobilePhone
Get-MgUser -UserId 'user@domain.com' | select businessPhones

Get-MgUser -UserId 'user@domain.com' | Select businessPhones, mobilePhone,
otherMails | Format-Table
```

Next step

After the authentication contact information is prepopulated for users, complete the following tutorial to enable self-service password reset:

[Enable Microsoft Entra self-service password reset](#)

Enable Microsoft Entra self-service password reset on the Windows sign-in screen

Article • 04/27/2025

By using self-service password reset (SSPR) in Microsoft Entra ID, users can change or reset their password with no administrator or helpdesk involvement. Typically, users open a web browser on another device to access the [SSPR portal](#). To improve the experience on computers that run Windows 7, 8, 8.1, 10, and 11, you can enable users to reset their password on the Windows sign-in screen.



This article shows administrators how to enable SSPR for Windows devices in an enterprise.

If your IT team hasn't enabled the ability to use SSPR from your Windows device or you have problems during sign-in, reach out to your helpdesk for more assistance.

General limitations

The following limitations apply to using SSPR from the Windows sign-in screen:

- Password reset isn't currently supported from a Remote Desktop or from Hyper-V enhanced sessions.
- Some non-Microsoft credential providers are known to cause problems with this feature.
- Disabling user account control via modification of the [EnableLUA registry key](#) is known to cause issues.

- This feature doesn't work for networks with 802.1x network authentication deployed and the option **Perform immediately before user logon**. For networks with 802.1x network authentication deployed, we recommend that you use machine authentication to enable this feature.
- Microsoft Entra hybrid-joined machines must have network connectivity line of sight to a domain controller to use the new password and update cached credentials. The devices must either be on the organization's internal network or on a virtual private network with network access to an on-premises domain controller. If SSPR is the only requirement, the network connection line to the domain controller isn't required.
- If you use an image, before you run `sysprep` ensure that the web cache is cleared for the built-in administrator before you perform the `CopyProfile` step. For more information, see [Performance poor when using custom default user profile ↗](#).
- The following settings are known to interfere with the ability to use and reset passwords on Windows 10 devices:
 - If lock screen notifications are turned off, **Reset password** won't work.
 - `HideFastUserSwitching` is set to **Enabled** or 1.
 - `DontDisplayLastUserName` is set to **Enabled** or 1.
 - `NoLockScreen` is set to **Enabled** or 1.
 - `BlockNonAdminUserInstall` is set to **Enabled** or 1.
 - `EnableLostMode` is set on the device.
 - `Explorer.exe` is replaced with a custom shell.
 - **Interactive logon: Require smart card** is set to **Enabled** or 1.
- The combination of the following specific three settings can cause this feature to not work.
 - **Interactive logon: Do not require CTRL+ALT+DEL** is set to **Disabled** (only for Windows 10 version 1710 and earlier).
 - `DisableLockScreenAppNotifications` is set to **Enabled** or 1.
 - Windows version is the Home edition.

 **Note**

These limitations also apply to Windows Hello for Business PIN reset from the device lock screen.

Windows 11 and Windows 10 password reset

To configure a Windows 11 or Windows 10 device for SSPR on the sign-in screen, review the following prerequisites and configuration steps.

Windows 11 and Windows 10 prerequisites

- Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Policy Administrator** and [enable Microsoft Entra SSPR](#).
- Users must register for SSPR before they use this feature on the [Windows sign-in screen](#).
 - All users must provide authentication contact information before they can reset their password, which isn't unique to using SSPR from the Windows sign-in screen.
- Network proxy requirements:
 - Port 443 to `passwordreset.microsoftonline.com` and `ajax.aspnetcdn.com`.
 - Windows 10 devices require a machine-level proxy configuration or scoped proxy configuration for the temporary `defaultuser1` account that's used to perform SSPR.
For more information, see the [Troubleshooting](#) section.
- Run at least Windows 10, version April 2018 Update (v1803), and the devices must be either:
 - Microsoft Entra joined.
 - Microsoft Entra hybrid joined.

Enable for Windows 11 and Windows 10 by using Intune

Deploying the configuration change to enable SSPR from the Windows sign-in screen by using Intune is the most flexible method. With Intune, you can deploy the configuration change to a specific group of machines that you define. This method requires Intune enrollment of the device.

Create a device configuration policy in Intune

1. Sign in to the [Microsoft Intune admin center](#).
2. Create a new device configuration profile by going to **Device configuration > Profiles** and then selecting **+ Create Profile**:
 - For **Platform**, choose **Windows 10 and later**.
 - For **Profile type**, choose **Templates** and then select the **Custom** template.
3. Select **Create**, and then provide a meaningful name for the profile, such as **Windows 11 sign-in screen SSPR**.
 Optionally, provide a meaningful description of the profile, and then select **Next**.

4. Under **Configuration settings**, select **Add** and provide the following OMA-URI setting to enable the reset password link:

- Enter a meaningful name to explain what the setting is doing, such as **Add SSPR link**.
- Optionally, enter a meaningful description of the setting.
- Set **OMA-URI** to
`./Device/Vendor/MSFT/Policy/Config/Authentication/AllowAadPasswordReset`.
- Set **Data type** to **Integer**.
- Set **Value** to **1**.

Select **Add**, and then select **Next**.

5. You can assign the policy to specific users, devices, or groups. Assign the profile that you want for your environment. Best practice is to assign it to a test group of devices first, and then select **Next**.

For more information, see [Assign user and device profiles in Microsoft Intune](#).

6. Configure the applicability rules that you want for your environment, such as **Assign profile if OS edition is Windows 10 Enterprise**, and then select **Next**.

7. Review your profile, and then select **Create**.

Enable for Windows 11 and Windows 10 by using the registry

To enable SSPR on the Windows sign-in screen by using a registry key, follow these steps:

1. Sign in to the Windows PC by using administrative credentials.
2. Select **Windows + R** to open the **Run** dialog, and then run **regedit** as an administrator.
3. Set the following registry key:

Windows Command Prompt

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\AzureADAccount
"AllowPasswordReset"=dword:00000001
```

Troubleshoot Windows 11 and Windows 10 password reset

If you have problems using SSPR from the Windows sign-in screen, the Microsoft Entra audit log includes information about the IP address and `ClientType`, where the password reset occurred, as shown in the following example output.

Activity Details: Audit log

Activity

Date : 9/10/2018 3:56:33 PM
Name : Reset password (self-service)
CorrelationId : eab2dxxx-xxxx-xxxx-xxxx-xxx7549ff5fc
Category : Self-service Password Management

Activity Status

Status : Success
Reason : Successfully completed reset.

Initiated By (Actor)

Type : User
ObjectId : 00000000-0000-0000-0000-000000000000
Upn : [Alain@contoso.com](#)
IpAddress : 192.168.10.10

Target(s)

Target
Type : User
ObjectId : eab2dxxx-xxxx-xxxx-xxxx-xxx7549fxxxx
Upn : [Alain@contoso.com](#)

Additional Details

ClientType : LogonClient_Windows7

When users reset their password from the sign-in screen of a Windows 11 or 10 device, a low-privilege temporary account called `defaultuser1` is created. This account is used to keep the password reset process secure.

The account itself has a randomly generated password, which is validated against an organization's password policy. The password doesn't show up for device sign-in and is automatically removed after the user resets their password. Multiple `defaultuser` profiles might exist, but you can safely ignore them.

Proxy configurations for Windows password reset

During the password reset, SSPR creates a temporary local user account to connect to <https://passwordreset.microsoftonline.com/n/passwordreset>. When a proxy is configured for user authentication, it might fail with the error "Something went wrong. Please, try again later." This error occurs because the local user account isn't authorized to use the authenticated proxy.

In this case, use one of the following workarounds:

- Configure a machine-wide proxy setting that doesn't depend on the type of user signed in to the machine. For example, you can enable the Group Policy **Make proxy settings per-machine (rather than per-user)** for the workstations.
- You can also use per-user proxy configuration for SSPR if you modify the registry template for the default account. The commands are:

Windows Command Prompt

```
reg load "hku\Default" "C:\Users\Default\NTUSER.DAT"
reg add "hku\Default\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings" /v ProxyEnable /t REG_DWORD /d "1" /f
reg add "hku\Default\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings" /v ProxyServer /t REG_SZ /d "<your proxy:port>" /f
reg unload "hku\Default"
```

- The error "Something went wrong" can also occur when anything interrupts connectivity to the URL <https://passwordreset.microsoftonline.com/n/passwordreset>. For example, this error can occur when antivirus software runs on the workstation without exclusions for the URLs passwordreset.microsoftonline.com, ajax.aspnetcdn.com, and ocsp.digicert.com. Disable this software temporarily to test if the issue is resolved or not.

Windows 7, 8, and 8.1 password reset

To configure a Windows 7, 8, or 8.1 device for SSPR on the Windows sign-in screen, review the following prerequisites and configuration steps.

Windows 7, 8, and 8.1 prerequisites

- Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Policy Administrator** and [enable Microsoft Entra SSPR](#).
- Users must register for SSPR on the [Windows sign-in screen](#) before they use this feature.
 - All users must provide authentication contact information before they can reset their password, which isn't unique to using SSPR from the Windows sign-in screen.
- Network proxy requirements:
 - Port 443 to passwordreset.microsoftonline.com.
- Patched Windows 7 or Windows 8.1 operating system.
- TLS 1.2 enabled by following the guidance in [Transport Layer Security \(TLS\) registry settings](#).
- If more than one non-Microsoft credential provider is enabled on your machine, users see more than one user profile on the Windows sign-in screen.

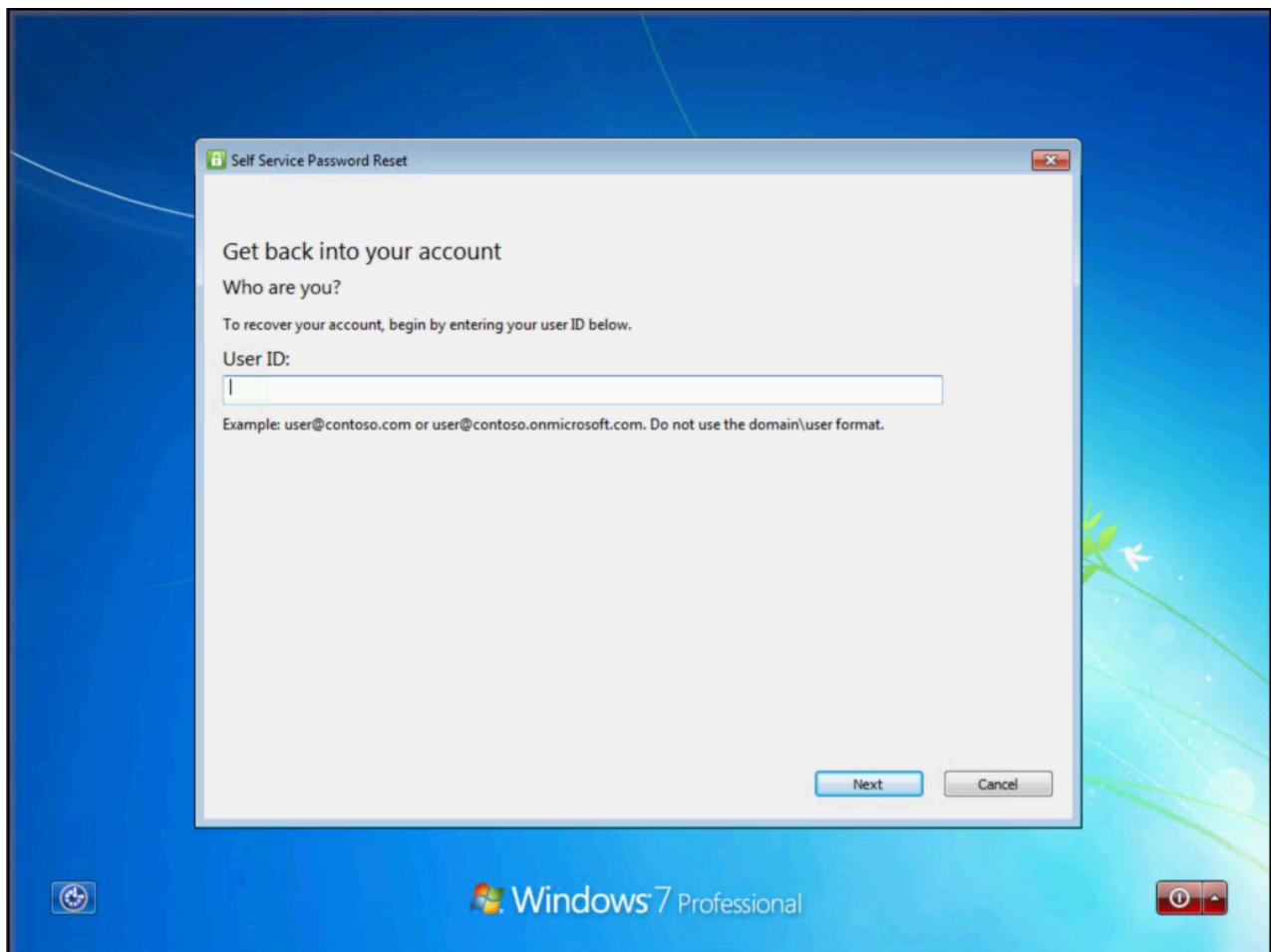
Warning

TLS 1.2 must be enabled, not just set to autonegotiate.

Install the SSPR component

For Windows 7, 8, and 8.1, a small component must be installed on the machine to enable SSPR on the Windows sign-in screen. To install this SSPR component, follow these steps:

1. Download the appropriate installer for the Windows version that you want to enable.
The software installer is available on the [Microsoft Download Center](#).
2. Sign in to the machine where you want to install, and run the installer.
3. After installation, we recommend that you perform a reboot.
4. After the reboot, on the Windows sign-in screen, choose a user and select **Forgot password?** to initiate the password reset workflow.
5. Follow the steps to reset your password.



Silent installation

To install or uninstall the SSPR component without prompts, use the following commands:

- **Silent install:** Use `msiexec /i SsprWindowsLogon.PROD.msi /qn`.
- **Silent uninstall:** Use `msiexec /x SsprWindowsLogon.PROD.msi /qn`.

Troubleshoot Windows 7, 8, and 8.1 password reset

If you have problems when you use SSPR from the Windows sign-in screen, events are logged on the machine and in Microsoft Entra ID. Microsoft Entra events include information about the IP address and the `ClientType` parameter where the password reset occurred.

The screenshot shows a window titled "Activity Details: Audit log". It contains the following sections and their details:

- Activity**
 - Date : 9/10/2018 3:56:33 PM
 - Name : Reset password (self-service)
 - CorrelationId : eab2dxxx-xxxx-xxxx-xxxx-xxx7549ff5fc
 - Category : Self-service Password Management
- Activity Status**
 - Status : Success
 - Reason : Successfully completed reset.
- Initiated By (Actor)**
 - Type : User
 - ObjectId : 00000000-0000-0000-0000-000000000000
 - Upn : [Alain@contoso.com](#)
 - IpAddress : 192.168.10.10
- Target(s)**
 - Target**
 - Type : User
 - ObjectId : eab2dxxx-xxxx-xxxx-xxxx-xxx7549xxxx
 - Upn : [Alain@contoso.com](#)
- Additional Details**
 - ClientType : LogonClient_Windows7

If more logging is required, change a registry key on the machine to enable verbose logging. Enable verbose logging for troubleshooting purposes only by using the following registry key value:

Windows Command Prompt

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential  
Providers\{86D2F0AC-2171-46CF-9998-4E33B3D7FD4F}
```

- To enable verbose logging, create `REG_DWORD: "EnableLogging"` and set it to **1**.
- To disable verbose logging, change `REG_DWORD: "EnableLogging"` to **0**.
- Review the debug logging in the Application event log under the source `AADPasswordResetCredentialProvider`.

What do users see?

With SSPR configured for your Windows devices, what are the changes for the user? How do they know that they can reset their password on the sign-in screen? The following example screenshots show other options for a user to reset their password by using SSPR.



When users attempt to sign in, they see a **Reset password** or **Forgot password** link that opens the SSPR experience on the sign-in screen. Now users can reset their password without having to use another device to access a web browser.

For more information on how to use this feature, see [Reset your work or school password ↗](#).

Related content

To simplify the user registration experience, you can [prepopulate user authentication contact information for SSPR](#).

Plan a Microsoft Entra multifactor authentication deployment

Article • 02/14/2025

Microsoft Entra multifactor authentication helps safeguard access to data and applications, providing another layer of security by using a second form of authentication. Organizations can enable multifactor authentication with [Conditional Access](#) to make the solution fit their specific needs.

This deployment guide shows you how to plan and implement a [Microsoft Entra multifactor authentication](#) roll-out.

Prerequisites for deploying Microsoft Entra multifactor authentication

Before you begin your deployment, ensure you meet the following prerequisites for your relevant scenarios.

[+] [Expand table](#)

Scenario	Prerequisite
Cloud-only identity environment with modern authentication	No prerequisite tasks
Hybrid identity scenarios	Deploy Microsoft Entra Connect and synchronize user identities between the on-premises Active Directory Domain Services (AD DS) and Microsoft Entra ID.
On-premises legacy applications published for cloud access	Deploy Microsoft Entra application proxy

Choose authentication methods for MFA

There are many methods that can be used for a second-factor authentication. You can choose from the list of available authentication methods, evaluating each in terms of security, usability, and availability.

ⓘ Important

Enable more than one MFA method so that users have a backup method available in case their primary method is unavailable. Methods include:

- Windows Hello for Business
- Microsoft Authenticator app
- FIDO2 security key
- Hardware OATH tokens (preview)
- Software OATH tokens
- SMS verification
- Voice call verification

When choosing authenticating methods that will be used in your tenant consider the security and usability of these methods:

Bad: Password	Good: Password and...	Better: Password and...	Best: Passwordless
123456 qwerty password iloveyou Password1	 SMS  Voice	 Authenticator (Push Notifications)  Software Tokens OTP  Hardware Tokens OTP (Preview)	 Authenticator (Phone Sign-in)  Window Hello  FIDO2 security key  Certificates

To learn more about the strength and security of these methods and how they work, see the following resources:

- [What authentication and verification methods are available in Microsoft Entra ID?](#)
- [Video: Choose the right authentication methods to keep your organization safe ↗](#)

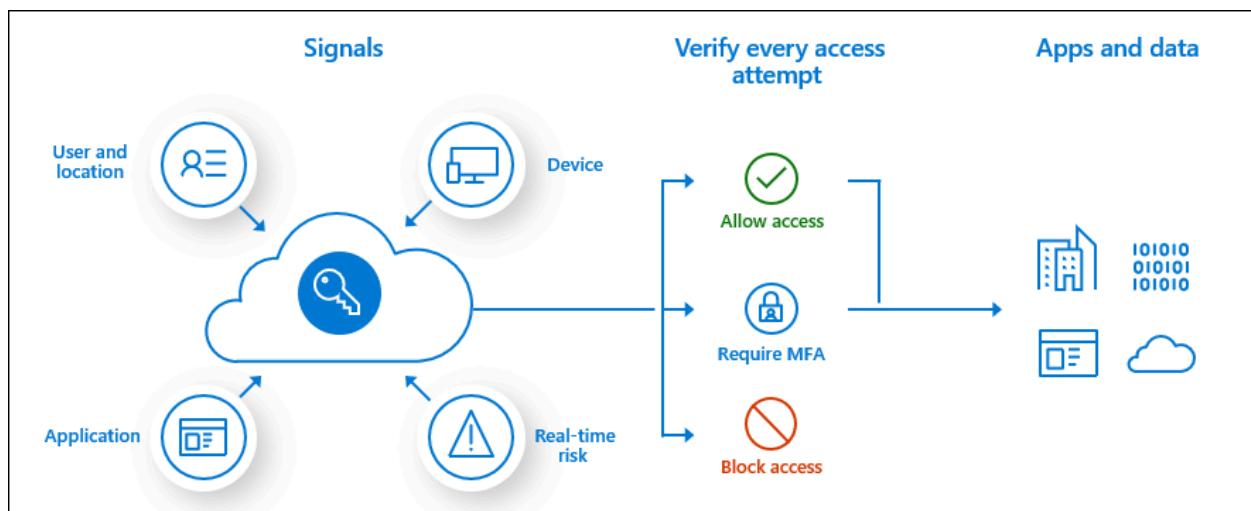
For the best flexibility and usability, use the Microsoft Authenticator app. This authentication method provides the best user experience and multiple modes, such as passwordless, MFA push notifications, and OATH codes. The Microsoft Authenticator app also meets the National Institute of Standards and Technology (NIST) [Authenticator Assurance Level 2 requirements](#).

You can control the authentication methods available in your tenant. For example, you may want to block some of the least secure methods, such as SMS.

Authentication method	Manage from	Scoping
Microsoft Authenticator (Push notification and passwordless phone sign-in)	MFA settings or Authentication methods policy	Authenticator passwordless phone sign-in can be scoped to users and groups
FIDO2 security key	Authentication methods policy	Can be scoped to users and groups
Software or Hardware OATH tokens	MFA settings	
SMS verification	MFA settings Manage SMS sign-in for primary authentication in authentication policy	SMS sign-in can be scoped to users and groups.
Voice calls	Authentication methods policy	

Plan Conditional Access policies

Microsoft Entra multifactor authentication is enforced with Conditional Access policies. These policies allow you to prompt users for MFA when needed for security and stay out of users' way when not needed.



In the Microsoft Entra admin center, you configure Conditional Access policies under **Protection > Conditional Access**.

To learn more about creating Conditional Access policies, see [Conditional Access policy to prompt for Microsoft Entra multifactor authentication when a user signs in](#). This helps you to:

- Become familiar with the user interface
- Get a first impression of how Conditional Access works

For end-to-end guidance on Microsoft Entra Conditional Access deployment, see the [Conditional Access deployment plan](#).

Common policies for Microsoft Entra multifactor authentication

Common use cases to require Microsoft Entra multifactor authentication include:

- For [administrators](#)
- To [specific applications](#)
- For [all users](#)
- For [Azure management](#)
- From [network locations you don't trust](#)

Named locations

To manage your Conditional Access policies, the location condition of a Conditional Access policy enables you to tie access controls settings to the network locations of your users. We recommend using [Named Locations](#) so that you can create logical groupings of IP address ranges or countries and regions. This creates a policy for all apps that blocks sign-in from that named location. Be sure to exempt your administrators from this policy.

Risk-based policies

If your organization uses [Microsoft Entra ID Protection](#) to detect risk signals, consider using [risk-based policies](#) instead of named locations. Policies can be created to force password changes when there's a threat of compromised identity or require MFA when a sign-in is deemed [at risk](#) such as leaked credentials, sign-ins from anonymous IP addresses, and more.

Risk policies include:

- [Require all users to register for Microsoft Entra multifactor authentication](#)
- [Require a password change for users that are high-risk](#)
- [Require MFA for users with medium or high sign in risk](#)

Convert users from per-user MFA to Conditional Access based MFA

If your users were enabled using per-user MFA enabled and enforced Microsoft Entra multifactor authentication, we recommend that you enable Conditional Access for all users and then manually disable per-user multifactor authentication. For more information, see [Create a Conditional Access policy](#).

Plan user session lifetime

When planning your multifactor authentication deployment, it's important to think about how frequently you would like to prompt your users. Asking users for credentials often seems like a sensible thing to do, but it can backfire. If users are trained to enter their credentials without thinking, they can unintentionally supply them to a malicious credential prompt. Microsoft Entra ID has multiple settings that determine how often you need to reauthenticate. Understand the needs of your business and users and configure settings that provide the best balance for your environment.

We recommend using devices with Primary Refresh Tokens (PRT) for improved end user experience and reduce the session lifetime with sign-in frequency policy only on specific business use cases.

For more information, see [Optimize reauthentication prompts and understand session lifetime for Microsoft Entra multifactor authentication](#).

Plan user registration

A major step in every multifactor authentication deployment is getting users registered to use Microsoft Entra multifactor authentication. Authentication methods such as Voice and SMS allow preregistration, while others like the Authenticator App require user interaction. Administrators must determine how users will register their methods.

Combined registration for SSPR and Microsoft Entra multifactor authentication

[The combined registration experience for Microsoft Entra multifactor authentication and self-service password reset \(SSPR\)](#) enables users to register for both MFA and SSPR in a unified experience. SSPR allows users to reset their password in a secure way using the same methods they use for Microsoft Entra multifactor authentication. To make sure you

understand the functionality and end-user experience, see the [Combined security information registration concepts](#).

It's critical to inform users about upcoming changes, registration requirements, and any necessary user actions. We provide [communication templates](#) and [user documentation](#) to prepare your users for the new experience and help to ensure a successful rollout. Send users to <https://myprofile.microsoft.com> to register by selecting the **Security Info** link on that page.

Registration with Microsoft Entra ID Protection

Microsoft Entra ID Protection contributes both a registration policy for and automated risk detection and remediation policies to the Microsoft Entra multifactor authentication story. Policies can be created to force password changes when there's a threat of compromised identity or require MFA when a sign-in is deemed risky. If you use Microsoft Entra ID Protection, [configure the Microsoft Entra multifactor authentication registration policy](#) to prompt your users to register the next time they sign in interactively.

Registration without Microsoft Entra ID Protection

If you don't have licenses that enable Microsoft Entra ID Protection, users are prompted to register the next time that MFA is required at sign-in. To require users to use MFA, you can use Conditional Access policies and target frequently used applications like HR systems. If a user's password is compromised, it could be used to register for MFA, taking control of their account. We therefore recommend [securing the security registration process with Conditional Access policies](#) requiring trusted devices and locations. You can further secure the process by also requiring a [Temporary Access Pass](#). A time-limited passcode issued by an admin that satisfies strong authentication requirements and can be used to onboard other authentication methods, including Passwordless ones.

Increase the security of registered users

If you have users registered for MFA using SMS or voice calls, you may want to move them to more secure methods such as the Microsoft Authenticator app. Microsoft now offers a public preview of functionality that allows you to prompt users to set up the Microsoft Authenticator app during sign-in. You can set these prompts by group, controlling who is prompted, enabling targeted campaigns to move users to the more secure method.

Plan recovery scenarios

As mentioned before, ensure users are registered for more than one MFA method, so that if one is unavailable, they have a backup. If the user doesn't have a backup method available, you can:

- Provide them a Temporary Access Pass so that they can manage their own authentication methods. You can also provide a Temporary Access Pass to enable temporary access to resources.
- Update their methods as an administrator. To do so, select the user in the Microsoft Entra admin center, then select **Protection > Authentication methods** and update their methods.

Plan integration with on-premises systems

Applications that authenticate directly with Microsoft Entra ID and have modern authentication (WS-Fed, SAML, OAuth, OpenID Connect) can make use of Conditional Access policies. Some legacy and on-premises applications don't authenticate directly against Microsoft Entra ID and require additional steps to use Microsoft Entra multifactor authentication. You can integrate them by using Microsoft Entra application proxy or [Network policy services](#).

Integrate with AD FS resources

We recommend migrating applications secured with Active Directory Federation Services (AD FS) to Microsoft Entra ID. However, if you aren't ready to migrate these to Microsoft Entra ID, you can use the Azure multifactor authentication adapter with AD FS 2016 or newer.

If your organization is federated with Microsoft Entra ID, you can [configure Microsoft Entra multifactor authentication as an authentication provider with AD FS resources](#) both on-premises and in the cloud.

RADIUS clients and Microsoft Entra multifactor authentication

For applications that are using RADIUS authentication, we recommend moving client applications to modern protocols such as SAML, OpenID Connect, or OAuth on Microsoft Entra ID. If the application can't be updated, then you can deploy [Network Policy Server \(NPS\) extension](#). The network policy server (NPS) extension acts as an

adapter between RADIUS-based applications and Microsoft Entra multifactor authentication to provide a second factor of authentication.

Common integrations

Many vendors now support SAML authentication for their applications. When possible, we recommend federating these applications with Microsoft Entra ID and enforcing MFA through Conditional Access. If your vendor doesn't support modern authentication – you can use the NPS extension. Common RADIUS client integrations include applications such as [Remote Desktop Gateways](#) and [VPN servers](#).

Others might include:

- Citrix Gateway
 - [Citrix Gateway](#) supports both RADIUS and NPS extension integration, and a SAML integration.
- Cisco VPN
 - The Cisco VPN supports both RADIUS and [SAML authentication for SSO](#).
 - By moving from RADIUS authentication to SAML, you can integrate the Cisco VPN without deploying the NPS extension.
- All VPNs

Deploy Microsoft Entra multifactor authentication

Your Microsoft Entra multifactor authentication rollout plan should include a pilot deployment followed by deployment waves that are within your support capacity. Begin your rollout by applying your Conditional Access policies to a small group of pilot users. After evaluating the effect on the pilot users, process used, and registration behaviors, you can either add more groups to the policy or add more users to the existing groups.

Follow these steps:

1. Meet the necessary prerequisites
2. Configure chosen authentication methods
3. Configure your Conditional Access policies
4. Configure session lifetime settings
5. Configure Microsoft Entra multifactor authentication registration policies

Manage Microsoft Entra multifactor authentication

This section provides reporting and troubleshooting information for Microsoft Entra multifactor authentication.

Reporting and Monitoring

Microsoft Entra ID has reports that provide technical and business insights, follow the progress of your deployment and check if your users are successful at sign-in with MFA. Have your business and technical application owners assume ownership of and consume these reports based on your organization's requirements.

You can monitor authentication method registration and usage across your organization using the [Authentication Methods Activity dashboard](#). This helps you understand what methods are being registered and how they're being used.

Use sign-in logs to review MFA events

The Microsoft Entra sign-in logs include authentication details for events when a user is prompted for MFA, and if any Conditional Access policies were in use.

NPS extension and AD FS logs for cloud MFA activity are now included in the [sign-in logs](#), and no longer published to the [Activity report](#).

For more information, and additional Microsoft Entra multifactor authentication reports, see [Review Microsoft Entra multifactor authentication events](#).

Troubleshoot Microsoft Entra multifactor authentication

See [Troubleshooting Microsoft Entra multifactor authentication](#) for common issues.

Guided walkthrough

For a guided walkthrough of many of the recommendations in this article, see the [Microsoft 365 Configure multifactor authentication guided walkthrough](#).

Next steps

[Deploy other identity features](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Configure Microsoft Entra multifactor authentication settings

Article • 03/24/2025

To customize the end-user experience for Microsoft Entra multifactor authentication (MFA), you can configure options for reporting suspicious activities. The following table describes Microsoft Entra MFA settings, and subsections cover each setting in more detail.

[+] Expand table

Feature	Description
Account lockout (MFA Server only)	Temporarily lock accounts from using Microsoft Entra MFA if there are too many denied authentication attempts in a row. This feature applies only to users who use MFA Server to enter a PIN to authenticate.
Report suspicious activity	Configure settings that allow users to report fraudulent verification requests.
OATH tokens	Used in cloud-based Microsoft Entra MFA environments to manage OATH tokens for users.
Phone call settings	Configure settings related to phone calls and greetings for cloud and on-premises environments.
Providers	This will show any existing authentication providers that you've associated with your account. Adding new providers is disabled as of September 1, 2018.

Account lockout (MFA Server only)

(i) Note

Account lockout only affects users who sign in by using MFA Server on-premises.

To prevent repeated MFA attempts as part of an attack, the account lockout settings let you specify how many failed attempts to allow before the account becomes locked out for a period of time. The account lockout settings are applied only when a PIN code is entered for the MFA prompt by using MFA Server on-premises.

The following settings are available:

- Number of MFA denials that trigger account lockout
- Minutes until account lockout counter is reset
- Minutes until account is automatically unblocked

To configure account lockout settings, complete these steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Multifactor authentication > Account lockout**. You might need to click **Show more** to see **Multifactor authentication**.
3. Enter the values for your environment, and then select **Save**.

The screenshot shows the Microsoft Entra admin center interface for configuring account lockout settings. The URL in the address bar is: Home > Multifactor authentication | Getting started > Security | Multifactor authentication > Multifactor authentication. The main title is "Multifactor authentication | Account lockout". On the left, there's a sidebar with links like "Getting started", "Diagnose and solve problems", "Settings" (which is expanded), and "Providers". Under "Settings", the "Account lockout" option is selected and highlighted with a grey background. The main content area contains three input fields with validation checkmarks: "Number of multifactor authentication denials to trigger account lockout" (set to 1), "Minutes until account lockout counter is reset" (set to 30), and "Minutes until account is automatically unblocked" (set to 90).

Report suspicious activity

When an unknown and suspicious MFA prompt is received, users can report the activity by using Microsoft Authenticator or through their phone. **Report suspicious activity** is integrated with [Microsoft Entra ID Protection](#) for risk-driven remediation, reporting, and least-privileged administration.

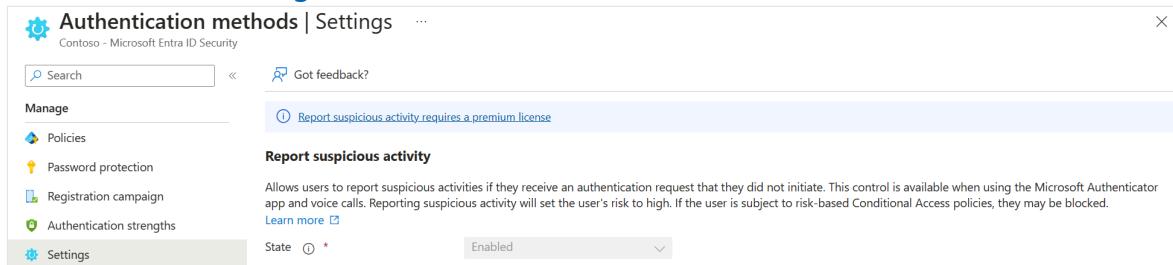
Users who report an MFA prompt as suspicious are set to **High User Risk**. Administrators can use risk-based policies to limit access for these users, or enable self-service password reset (SSPR) for users to remediate problems on their own.

If you don't have a Microsoft Entra ID P2 license for risk-based policies, you can use risk detection events to either manually identify and disable impacted users, or set up automation by using custom workflows with Microsoft Graph. For more information about investigating and remediating user risk, see:

- [How to: Investigate risk](#)
- [How to: Remediate risks and unblock users](#)

To enable **Report suspicious activity** from the Authentication methods policy **Settings**:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Settings**.
3. Set **Report suspicious activity** to **Enabled**. The feature remains disabled if you choose **Microsoft managed**. For more information about Microsoft managed values, see [Protecting authentication methods in Microsoft Entra ID](#).



4. Select **All users** or a specific group.
5. If you also upload custom greetings for your tenant, select a **Reporting code**. The reporting code is the number that users enter into their phone to report suspicious activity. The reporting code is only applicable if custom greetings are also uploaded by an [Authentication Policy Administrator](#). Otherwise, the default code is 0, regardless of any value specified in the policy.
6. Click **Save**.

Remediating risk for tenants with Microsoft Entra ID P1 license

When a user reports an MFA prompt as suspicious, the event shows up in the sign-in logs (as a sign-in that was rejected by the user), in the Audit logs, and in the Risk detections report.

[Expand table](#)

Report	Admin center	Details
Risk detections report	Protection > Identity Protection > Risk detection	Detection type: User Reported Suspicious Activity Risk level: High Source End user reported
Sign-in logs	Identity > Monitoring & health > Sign-in logs > Authentication details	Result detail will show as MFA denied

Report	Admin center	Details
Audit logs	Identity > Monitoring & health > Audit logs	The suspicious activity appears under Activity type

 **Note**

A user isn't reported as High Risk if they perform passwordless authentication.

You can also query for risk detections and users flagged as risky by using Microsoft Graph.

 Expand table

API	Detail
riskDetection resource type	riskEventType: <code>userReportedSuspiciousActivity</code>
List riskyUsers	riskLevel = <code>high</code>

For manual remediation, administrators or helpdesk can ask the users to reset their password by using self-service password reset (SSPR), or do so on their behalf. For automated remediation, use the Microsoft Graph APIs, or use PowerShell to create a script that changes the user's password, forces SSPR, revokes sign-in sessions, or temporarily disables the user account.

Remediating risk for tenants with Microsoft Entra ID P2 license

Tenants with a Microsoft Entra ID P2 license can use risk-based Conditional Access policies to automatically remediate user risk, in addition to the options Microsoft Entra ID P2 license.

Configure a policy that looks at user risk under **Conditions > User risk**. Look for users where risk = high to either block them from sign in or require them to reset their password.

Home > Conditional Access | Overview >

New ...

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies. [Learn more ↗](#)

Name *

Example: 'Device compliance app policy'

Assignments

Users ⓘ 0 users and groups selected

Target resources ⓘ No target resources selected

Network **NEW** ⓘ Not configured

Conditions ⓘ 1 condition selected

Access controls

Grant ⓘ 1 control selected

Session ⓘ 0 controls selected

User risk

Configure ⓘ Yes No

Configure user risk levels needed for policy to be enforced

High

Medium

Low

Sign-in risk ⓘ Sign-in risk level is the likelihood that the sign-in session is compromised.

Not configured

Insider risk (Preview) ⓘ Insider risk assesses the user's risky data-related activity in Microsoft Purview Insider Risk Management.

Not configured

Device platforms ⓘ Not configured

Locations ⓘ Not configured

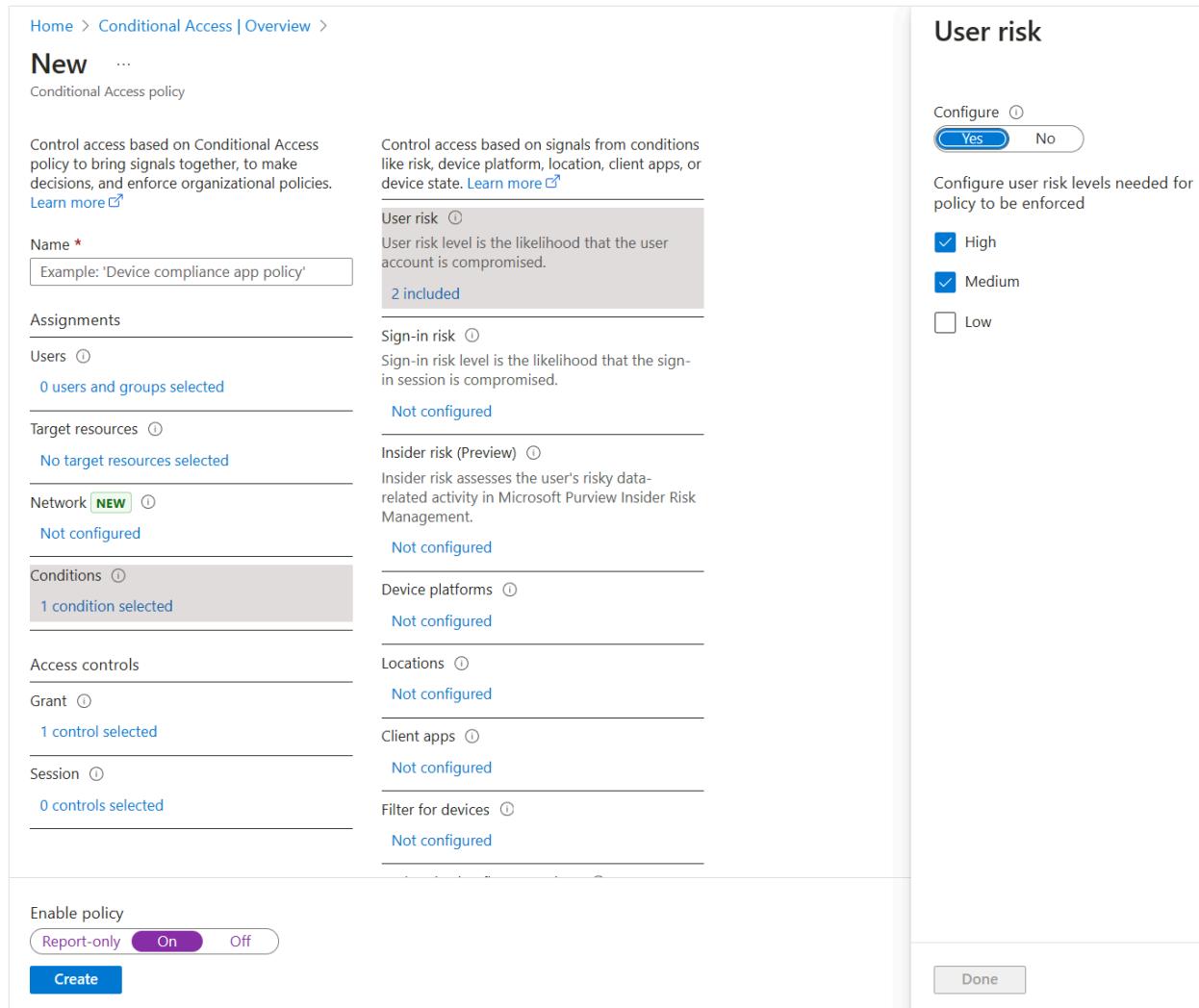
Client apps ⓘ Not configured

Filter for devices ⓘ Not configured

Enable policy

Report-only On Off

Create Done



For more information, see [Sign-in risk-based Conditional Access policy](#).

OATH tokens

Microsoft Entra ID supports the use of OATH TOTP SHA-1 tokens that refresh codes every 30 or 60 seconds. You can purchase these tokens from the vendor of your choice.

OATH TOTP hardware tokens typically come with a secret key, or seed, pre-programmed in the token. You need to input these keys into Microsoft Entra ID as described in the following steps. Secret keys are limited to 128 characters, which might not be compatible with all tokens. The secret key can contain only the characters *a-z* or *A-Z* and digits 1-7. It must be encoded in Base32.

Programmable OATH TOTP hardware tokens that can be reseeded can also be set up with Microsoft Entra ID in the software token setup flow.

OATH hardware tokens are supported as part of a public preview. For more information about previews, see [Supplemental Terms of Use for Microsoft Azure Previews ↗](#).

To get started, select the Upload button above and choose a .csv file. This file should contain the secret keys for the OATH tokens you wish to use. The columns in the file should be: "upn, serial number, secret key, time interval, manufacturer, model". For more information on available authentication and verification methods, view the public documentation.

Name	Username	Serial Number	Model	Manufacturer	Activated
<input type="checkbox"/> le	le	SafeID	DeepNet Security	Activate	

After you acquire tokens, you need to upload them in a comma-separated values (CSV) file format. Include the UPN, serial number, secret key, time interval, manufacturer, and model, as shown in this example:

```
CSV

upn,serial number,secret key,time interval,manufacturer,model
Helga@contoso.com,1234567,1234567abcdef1234567abcdef,60,Contoso,HardwareKey
```

ⓘ Note

Be sure to include the header row in your CSV file.

1. Sign in to the [Microsoft Entra admin center](#) as a [Global Administrator](#).
2. Go to **Protection > Multifactor authentication > OATH tokens**, and upload the CSV file.

Depending on the size of the CSV file, it might take a few minutes to process. Select **Refresh** to get the status. If there are any errors in the file, you can download a CSV file that lists them. The field names in the downloaded CSV file are different from those in the uploaded version.

After any errors are addressed, the administrator can activate each key by selecting **Activate** for the token and entering the OTP displayed in the token.

Users can have a combination of up to five OATH hardware tokens or authenticator applications, such as the Microsoft Authenticator app, configured for use at any time.

ⓘ Important

Make sure to only assign each token to a single user. In the future, support for the assignment of a single token to multiple users will stop to prevent a security risk.

Phone call settings

If users receive phone calls for MFA prompts, you can configure their experience, such as caller ID or the voice greeting they hear.

In the United States, if you haven't configured MFA caller ID, voice calls from Microsoft come from the following numbers. Users with spam filters should exclude these numbers.

Default number: +1 (855) 330-8653

The following table lists more numbers for different countries/regions.

[+] [Expand table](#)

Country/Region	Number(s)
Austria	+43 6703062076
Bangladesh	+880 9604606026
China	+44 1235619418, +44 1235619536, +44 1235619537, +44 1235619538, +44 1235619539, +44 1235619535, +44 7897087681, +44 7897087690, +44 7897087692, +66 977832930
Croatia	+385 15507766
Ecuador	+593 964256042
Estonia	+372 6712726
France	+33 744081468
Ghana	+233 308250245
Greece	+30 2119902739
Guatemala	+502 23055056
Hong Kong SAR	+852 25716964
India	+91 3371568300, +91 1205089400, +91 4471566601, +91 2271897557, +91 1203524400, +91 3335105700, +91 2235544120, +91 4435279600
Jordan	+962 797639442
Kenya	+254 709605276
Netherlands	+31 202490048

Country/Region	Number(s)
Nigeria	+234 7080627886
Pakistan	+92 4232618686, +44 7897087681, +44 7897087690, +44 7897087692, +66 977832930
Poland	+48 699740036
Saudi Arabia	+966 115122726
South Africa	+27 872405062
Spain	+34 913305144
Sri Lanka	+94 117750440
Sweden	+46 701924176
Taiwan	+886 277515260, +886 255686508
Türkiye	+90 8505404893
Ukraine	+380 443332393
United Arab Emirates	+971 44015046
Vietnam	+84 2039990161

ⓘ Note

When Microsoft Entra multifactor authentication calls are placed through the public telephone network, sometimes the calls are routed through a carrier that doesn't support caller ID. Because of this, caller ID isn't guaranteed, even though Microsoft Entra multifactor authentication always sends it. This applies both to phone calls and text messages provided by Microsoft Entra multifactor authentication. If you need to validate that a text message is from Microsoft Entra multifactor authentication, see [What short codes are used for sending messages?](#).

To configure your own caller ID number, complete the following steps:

1. Go to **Protection > Multifactor authentication > Phone call settings**.
2. Set the **MFA caller ID number** to the number you want users to see on their phones. Only US-based numbers are allowed.
3. Select **Save**.

Note

When Microsoft Entra multifactor authentication calls are placed through the public telephone network, sometimes the calls are routed through a carrier that doesn't support caller ID. Because of this, caller ID isn't guaranteed, even though Microsoft Entra multifactor authentication always sends it. This applies both to phone calls and text messages provided by Microsoft Entra multifactor authentication. If you need to validate that a text message is from Microsoft Entra multifactor authentication, see [What short codes are used for sending messages?](#).

Custom voice messages

You can use your own recordings or greetings for Microsoft Entra multifactor authentication. These messages can be used in addition to the default Microsoft recordings or to replace them.

Before you begin, be aware of the following restrictions:

- The supported file formats are .wav and .mp3.
- The file size limit is 1 MB.
- Authentication messages should be shorter than 20 seconds. Messages that are longer than 20 seconds can cause the verification to fail. If the user doesn't respond before the message finishes, the verification times out.

Custom message language behavior

When a custom voice message is played to the user, the language of the message depends on the following factors:

- The language of the user.
 - The language detected by the user's browser.
 - Other authentication scenarios might behave differently.
- The language of any available custom messages.
 - This language is chosen by the administrator when a custom message is added.

For example, if there's only one custom message, and it's in German:

- A user who authenticates in the German language will hear the custom German message.
- A user who authenticates in English will hear the standard English message.

Custom voice message defaults

You can use the following sample scripts to create your own custom messages. These phrases are the defaults if you don't configure your own custom messages.

[Expand table](#)

Message name	Script
Authentication successful	Your sign-in was successfully verified. Goodbye.
Extension prompt	Thank you for using Microsoft's sign-in verification system. Please press the pound key to continue.
Activation	Thank you for using the Microsoft sign-in verification system. Please press the pound key to finish your verification.
Authentication denied retry	Verification denied.
Retry (standard)	Thank you for using the Microsoft sign-in verification system. Please press the pound key to finish your verification.
Greeting (standard)	Thank you for using the Microsoft sign-in verification system. Please press the pound key to finish your verification.
Greeting (PIN)	Thank you for using Microsoft's sign-in verification system. Please enter your PIN followed by the pound key to finish your verification.
Retry (PIN)	Thank you for using Microsoft's sign-in verification system. Please enter your PIN followed by the pound key to finish your verification.
Extension prompt after digits	If already at this extension, press the pound key to continue.
Authentication denied	I'm sorry, we cannot sign you in at this time. Please try again later.
Activation greeting (standard)	Thank you for using the Microsoft sign-in verification system. Please press the pound key to finish your verification.
Activation retry (standard)	Thank you for using the Microsoft sign-in verification system. Please press the pound key to finish your verification.
Activation greeting (PIN)	Thank you for using Microsoft's sign-in verification system. Please enter your PIN followed by the pound key to finish your verification.
Extension prompt before digits	Thank you for using Microsoft's sign-in verification system. Please transfer this call to extension <extension>.

Set up a custom message

To use your own custom messages, complete the following steps:

1. Go to **Protection > Multifactor authentication > Phone call settings**.
2. Select **Add greeting**.
3. Choose the **Type** of greeting, such as **Greeting (standard)** or **Authentication successful**.
4. Select the **Language**. See the previous section on [custom message language behavior](#).
5. Browse for and select an .mp3 or .wav sound file to upload.
6. Select **Add** and then **Save**.

MFA service settings

Settings for app passwords, trusted IPs, verification options, and remembering multifactor authentication on trusted devices are available in the service settings. This is a legacy portal.

You can access service settings from the [Microsoft Entra admin center](#) by going to **Protection > Multifactor authentication > Getting started > Configure > Additional cloud-based MFA settings**. A window or tab opens with additional service settings options.

Trusted IPs

Location conditions are the recommended way to configure MFA with Conditional Access because of IPv6 support and other improvements. For more information about location conditions, see [Using the location condition in a Conditional Access policy](#). For steps to define locations and create a Conditional Access policy, see [Conditional Access: Block access by location](#).

The trusted IPs feature of Microsoft Entra multifactor authentication also bypasses MFA prompts for users who sign in from a defined IP address range. You can set trusted IP ranges for your on-premises environments. When users are in one of these locations, there's no Microsoft Entra multifactor authentication prompt. The trusted IPs feature requires Microsoft Entra ID P1 edition.

 **Note**

The trusted IPs can include private IP ranges only when you use MFA Server. For cloud-based Microsoft Entra multifactor authentication, you can use only public IP address ranges.

IPv6 ranges are supported in [named locations](#).

If your organization uses the NPS extension to provide MFA to on-premises applications, the source IP address will always appear to be the NPS server that the authentication attempt flows through.

[] [Expand table](#)

Microsoft Entra tenant type	Trusted IP feature options
Managed	Specific range of IP addresses: Administrators specify a range of IP addresses that can bypass multifactor authentications for users who sign in from the company intranet. A maximum of 50 trusted IP ranges can be configured.
Federated	All Federated Users: All federated users who sign in from inside the organization can bypass multifactor authentications. Users bypass verifications by using a claim that's issued by Active Directory Federation Services (AD FS). Specific range of IP addresses: Administrators specify a range of IP addresses that can bypass multifactor authentication for users who sign in from the company intranet.

Trusted IP bypass works only from inside the company intranet. If you select the **All Federated Users** option and a user signs in from outside the company intranet, the user has to authenticate by using multifactor authentication. The process is the same even if the user presents an AD FS claim.

(!) **Note**

If both per-user MFA and Conditional Access policies are configured in the tenant, you need to add trusted IPs to the Conditional Access policy and update the MFA service settings.

User experience inside the corporate network

When the trusted IPs feature is disabled, multifactor authentication is required for browser flows. App passwords are required for older rich-client applications.

When trusted IPs are used, multifactor authentication isn't required for browser flows. App passwords aren't required for older rich-client applications if the user hasn't created an app password. After an app password is in use, the password is required.

User experience outside the corporate network

Regardless of whether trusted IPs are defined, multifactor authentication is required for browser flows. App passwords are required for older rich-client applications.

Enable named locations by using Conditional Access

You can use Conditional Access rules to define named locations by using the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Conditional Access Administrator](#).
2. Browse to **Protection > Conditional Access > Named locations**.
3. Select **New location**.
4. Enter a name for the location.
5. Select **Mark as trusted location**.
6. Enter the IP range for your environment in CIDR notation. For example, **40.77.182.32/27**.
7. Select **Create**.

Enable the trusted IPs feature by using Conditional Access

To enable trusted IPs by using Conditional Access policies, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Conditional Access Administrator](#).
2. Browse to **Protection > Conditional Access > Named locations**.
3. Select **Configure multifactor authentication trusted IPs**.
4. On the **Service Settings** page, under **Trusted IPs**, choose one of these options:
 - **For requests from federated users originating from my intranet**: To choose this option, select the checkbox. All federated users who sign in from the corporate network bypass multifactor authentications by using a claim that's issued by AD FS. Ensure that AD FS has a rule to add the intranet claim to the appropriate traffic. If the rule doesn't exist, create the following rule in AD FS:

```
c:[Type==  
"https://schemas.microsoft.com/ws/2012/01/insidecorporatenetwork"] =>  
issue(claim = c);
```

ⓘ Note

The **Skip multifactor authentication for requests from federated users on my intranet** option will affect the Conditional Access evaluation for locations. Any request with the **insidecorporatenetwork** claim would be treated as coming from a Trusted location if that option is selected.

- **For requests from a specific range of public IPs:** To choose this option, enter the IP addresses in the text box, in CIDR notation.
 - For IP addresses that are in the range xxx.xxx.xxx.1 through xxx.xxx.xxx.254, use notation like **xxx.xxx.xxx.0/24**.
 - For a single IP address, use notation like **xxx.xxx.xxx.xxx/32**.
 - Enter up to 50 IP address ranges. Users who sign in from these IP addresses bypass multifactor authentications.

5. Select **Save**.

Enable the trusted IPs feature by using service settings

If you don't want to use Conditional Access policies to enable trusted IPs, you can configure the service settings for Microsoft Entra multifactor authentication by using the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Policy Administrator**.
2. Browse to **Protection > Multifactor authentication > Additional cloud-based MFA settings**.
3. On the **Service settings** page, under **Trusted IPs**, choose one or both of the following options:
 - **For requests from federated users on my intranet:** To choose this option, select the checkbox. All federated users who sign in from the corporate network bypass multifactor authentication by using a claim that's issued by AD FS. Ensure that AD FS has a rule to add the intranet claim to the appropriate traffic. If the rule doesn't exist, create the following rule in AD FS:

```
c:[Type==  
"https://schemas.microsoft.com/ws/2012/01/insidecorporatenetwork"] =>  
issue(claim = c);
```

- **For requests from a specified range of IP address subnets:** To choose this option, enter the IP addresses in the text box, in CIDR notation.
 - For IP addresses that are in the range xxx.xxx.xxx.1 through xxx.xxx.xxx.254, use notation like **xxx.xxx.xxx.0/24**.
 - For a single IP address, use notation like **xxx.xxx.xxx.xxx/32**.
 - Enter up to 50 IP address ranges. Users who sign in from these IP addresses bypass multifactor authentications.

4. Select Save.

Verification methods

You can choose the verification methods that are available for your users in the service settings portal. When your users enroll their accounts for Microsoft Entra multifactor authentication, they choose their preferred verification method from the options that you've enabled. Guidance for the user enrollment process is provided in [Set up my account for multifactor authentication ↗](#).

ⓘ Important

In March 2023, we announced the deprecation of managing authentication methods in the legacy multifactor authentication and self-service password reset (SSPR) policies. Beginning September 30, 2025, authentication methods can't be managed in these legacy MFA and SSPR policies. We recommend customers use the manual migration control to migrate to the Authentication methods policy by the deprecation date. For help with the migration control, see [How to migrate MFA and SSPR policy settings to the Authentication methods policy for Microsoft Entra ID](#).

The following verification methods are available:

[+] Expand table

Method	Description
Call to phone	Places an automated voice call. The user answers the call and presses # on the phone to authenticate. The phone number isn't synchronized to on-premises Active Directory.

Method	Description
Text message to phone	Sends a text message that contains a verification code. The user is prompted to enter the verification code into the sign-in interface. This process is called one-way SMS. Two-way SMS means that the user must text back a particular code. Two-way SMS is deprecated and not supported after November 14, 2018. Administrators should enable another method for users who previously used two-way SMS.
Notification through mobile app	Sends a push notification to the user's phone or registered device. The user views the notification and selects Verify to complete verification. The Microsoft Authenticator app is available for Windows Phone , Android , and iOS .
Verification code from mobile app or hardware token	The Microsoft Authenticator app generates a new OATH verification code every 30 seconds. The user enters the verification code into the sign-in interface. The Microsoft Authenticator app is available for Windows Phone , Android , and iOS .

For more information, see [What authentication and verification methods are available in Microsoft Entra ID?](#)

Enable and disable verification methods

To enable or disable verification methods, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Identity > Users > All users**.
3. Select **Per-user MFA**.
4. Under **Multifactor authentication** at the top of the page, select **Service settings**.
5. On the **Service settings** page, under **Verification options**, select or clear the appropriate checkboxes.
6. Select **Save**.

Remember multifactor authentication

The **remember multifactor authentication** feature lets users bypass subsequent verifications for a specified number of days, after they've successfully signed in to a device by using MFA. To enhance usability and minimize the number of times a user has to perform MFA on a given device, select a duration of 90 days or less.

 **Important**

If an account or device is compromised, remembering MFA for trusted devices can affect security. If a corporate account becomes compromised or a trusted device is lost or stolen, you should [Revoke MFA Sessions](#).

The revoke action revokes the trusted status from all devices, and the user is required to perform multifactor authentication again. You can also instruct your users to restore the original MFA status on their own devices as noted in [Manage your settings for multifactor authentication](#).

How the feature works

The **remember multifactor authentication** feature sets a persistent cookie on the browser when a user selects the **Don't ask again for X days** option at sign-in. The user isn't prompted again for MFA from that browser until the cookie expires. If the user opens a different browser on the same device or clears the cookies, they're prompted again to verify.

The **Don't ask again for X days** option isn't shown on non-browser applications, regardless of whether the app supports modern authentication. These apps use *refresh tokens* that provide new access tokens every hour. When a refresh token is validated, Microsoft Entra ID checks that the last multifactor authentication occurred within the specified number of days.

The feature reduces the number of authentications on web apps, which normally prompt every time. The feature can increase the number of authentications for modern authentication clients that normally prompt every 180 days, if a lower duration is configured. It might also increase the number of authentications when combined with Conditional Access policies.

Important

The **remember multifactor authentication** feature isn't compatible with the **keep me signed in** feature of AD FS, when users perform multifactor authentication for AD FS through MFA Server or a third-party multifactor authentication solution.

If your users select **keep me signed in** on AD FS and also mark their device as trusted for MFA, the user isn't automatically verified after the **remember multifactor authentication** number of days expires. Microsoft Entra ID requests a fresh multifactor authentication, but AD FS returns a token with the original MFA claim and date, rather than performing multifactor authentication again. *This reaction sets off a verification loop between Microsoft Entra ID and AD FS.*

The **remember multifactor authentication** feature isn't compatible with B2B users and won't be visible for B2B users when they sign in to the invited tenants.

The **remember multifactor authentication** feature isn't compatible with the Sign-in frequency Conditional Access control. For more information, see [Configure authentication session management with Conditional Access](#).

Enable remember multifactor authentication

To enable and configure the option to allow users to remember their MFA status and bypass prompts, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Identity > Users > All users**.
3. Select **Per-user MFA**.
4. Under **Multifactor authentication** at the top of the page, select **service settings**.
5. On the **service settings** page, under **remember multifactor authentication**, select **Allow users to remember multifactor authentication on devices they trust**.
6. Set the number of days to allow trusted devices to bypass multifactor authentications. For the optimal user experience, extend the duration to 90 or more days.
7. Select **Save**.

Mark a device as trusted

After you enable the **remember multifactor authentication** feature, users can mark a device as trusted when they sign in by selecting **Don't ask again**.

Next steps

To learn more, see [What authentication and verification methods are available in Microsoft Entra ID?](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

How to verify that users are set up for mandatory MFA

Article • 04/25/2025

This topic covers steps to verify that users in your organization are set up to meet Azure's mandatory MFA requirements. For more information about which applications and accounts are affected and how the rollout works, see [Planning for mandatory multifactor authentication for Azure and other admin portals](#).

Verify MFA for a personal account

A user might use their personal account to create a Microsoft Entra tenant for only a few users. If you used your personal account to subscribe to Azure, complete the following steps to confirm that your account is set up for MFA.

1. Sign in to your Microsoft account [Advanced security options](#).
2. Under **Additional security** and **Two-step verification** choose **Turn on**.
3. Follow the instructions shown on the screen.

For more information, see [How to use two-step verification with your Microsoft account](#).

Find users who sign in with and without MFA

Use the following resources to find users who sign in with and without MFA:

- To export a list of users and their authentication methods, use [PowerShell](#).
- If you run queries to analyze user sign-ins, use the application IDs of the [applications that require MFA](#).

Verify MFA enablement

All users who access [applications that require MFA](#) must be set up to use MFA. Mandatory MFA isn't restricted to privileged roles. As a best practice, all users who access *any* administration portal should use MFA.

Use the following steps to verify that MFA is set up for your users, or to enable it if needed.

1. Sign in to Azure portal as a Global Reader.
2. Browse to **Entra ID > Overview**.

3. Check the license type for the tenant subscription.
4. Follow the steps for your license type to verify MFA is enabled, or enable it if needed. To complete these steps, you need to sign out as a Global Reader, and sign back in with a more privileged role.
 - Microsoft Entra ID P1 or Microsoft Entra ID P2
 - Microsoft 365 or Microsoft Entra ID Free

Verify MFA is enabled for Microsoft Entra ID P1 or Microsoft Entra ID P2 license

If you have a Microsoft Entra ID P1 or Microsoft Entra ID P2 license, you can create a Conditional Access policy to require MFA for users who access [applications that require MFA](#):

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Conditional Access Administrator](#).
2. Browse to **Entra ID > Conditional Access > Policies**.
3. Select **New policy**.
4. Give your policy a name. We recommend that organizations create a meaningful standard for the names of their policies.
5. Under **Assignments**, select **Users or workload identities**.
6. Under **Include**, select **All users**, or a group of users who sign in to the [applications that require MFA](#).
7. Under **Target resources > Cloud apps > Include**, Select **apps**, select **Microsoft Admin Portals and Windows Azure Service Management API**.
8. Under **Access controls > Grant**, select **Grant access, Require authentication strength**, select **Multifactor authentication**, and select **Select**.
9. Confirm your settings and set **Enable policy to Report-only**.
10. Select **Create** to create to enable your policy.

 **Important**

Create the CA Policy in Report-Only mode to understand impact for your tenant so you don't get locked out.

For more information, see [Common Conditional Access policy: Require multifactor authentication for admins accessing Microsoft admin portals](#).

You can use the Conditional Access insights and reporting workbook that contains sign-in logs to understand impact for your tenant. As a prerequisite, you need to:

- [Create a workspace](#).
- [Integrate activity logs with Azure Monitor](#).

Choose the following parameters to see how mandatory MFA affects your tenant:

- Select the previously created MFA Conditional Access policy
- Select a time range to evaluate the data
- Select **Data View** to see results in terms of number of users or number of sign-ins

You can query the sign-in details or download the sign-in logs to dive deeper into the data. For more information, see [Conditional Access insights and reporting](#).

Verify MFA is enabled for Microsoft 365 or Microsoft Entra ID Free

If you have a Microsoft 365 or Microsoft Entra ID Free license, you can enable MFA by using security defaults. Users are prompted for MFA as needed, but you can't define your own rules to control the behavior.

To enable security defaults:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Security Administrator**.
2. Browse to **Entra ID > Overview > Properties**.
3. Select **Manage security defaults**.
4. Set **Security defaults** to **Enabled**.
5. Select **Save**.

For more information about security defaults, see [Security defaults in Microsoft Entra ID](#).

If you don't want to use security defaults, you can enable per-user MFA. When you enable users individually, they perform MFA each time they sign in. An Authentication Administrator can enable some exceptions. To enable per-user MFA:

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Administrator**.
2. Browse to **Entra ID > Users**.
3. Select a user account, and click **Enable MFA**.
4. Confirm your selection in the pop-up window that opens.

After you enable users, notify them by email. Tell the users that a prompt is displayed to ask them to register the next time they sign in. For more information, see [Enable per-user Microsoft Entra multifactor authentication to secure sign-in events](#).

Related content

Review the following topics to learn more about MFA:

- [How to postpone enforcement for a tenant where users are unable to sign in after rollout of mandatory multifactor authentication \(MFA\) requirement for the Azure portal, Microsoft Entra admin center, or Microsoft Intune admin center](#)
- [Planning for mandatory multifactor authentication for Azure and other admin portals](#)
- [Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication](#)
- [Secure sign-in events with Microsoft Entra multifactor](#)
- [Plan a Microsoft Entra multifactor authentication deployment](#)
- [Phishing-resistant MFA methods](#)
- [Microsoft Entra multifactor authentication](#)
- [Authentication methods](#)

How to postpone enforcement for a tenant where users are unable to sign in after rollout of mandatory MFA requirement

Article • 04/20/2025

Users might not be able to sign into the Azure portal, Microsoft Entra admin center, or Microsoft Intune admin center if they have trouble using their MFA method after the mandatory requirement to use MFA is rolled out to their tenant.

If users are unable to sign in, you can run the following script as a Global Administrator to temporarily postpone the MFA enforcement for your tenant.

For more information about Azure's mandatory MFA requirements, see [Planning for mandatory multifactor authentication for Azure and other admin portals](#). The following script applies only to applications in Phase 1.

Script actions

The script takes the following actions:

- Picks the user's tenant if they have one, or presents a list of tenants for them to choose from. Optionally, the script asks for the date of enforcement. The default date is September 30, 2025.
- Logs the user into that tenant.
- Gets the relevant authentication tokens.
- Checks if user has elevated access. If not, the script does the elevation.
- Checks if the appropriate role is assigned for the user on the settings resource provider (RP). If not, the script assigns the appropriate role.
- Updates the enforcement date in Entra ID.
- Tries to remove elevated access if the script added it.

Prerequisites

- [Az PowerShell module](#)
- Global Administrator role

Script

PowerShell

```

param (
    [Parameter(Mandatory=$false)]
    [string]$TenantId,
    [Parameter(Mandatory=$false)]
    [string]$PostponementDateInUTC
)

# Make sure the Az.Accounts module is imported
Import-Module Az.Accounts

function Set-TenantSettingsMFAPostponement {
    Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope Process

    $cutoffDate = [datetime]::Parse("2025-10-01T00:00:00Z")
    $isDefaultDate = $false
    if ($PostponementDateInUTC) {
        # ISO 8601 check (basic): YYYY-MM-DDTHH:mm:ssZ
        if ($PostponementDateInUTC -notmatch '^\\d{4}-\\d{2}-\\d{2}T\\d{2}:\\d{2}:\\d{2}Z$') {
            Write-Host "Invalid PostponementDateInUTC format. Must be in ISO 8601 format like '2025-09-30T23:59:59Z'." -ForegroundColor Red
            return
        }
        $valid = $false
        $today = [DateTime]::UtcNow.Date
        $minDate = $today.AddDays(1)
        $maxDate = [DateTime]::ParseExact("2025-09-30T23:59:59Z", "yyyy-MM-ddTHH:mm:ssZ", $null).ToUniversalTime()
        $valid = Check-Date-Is-Valid -maxDate $maxDate -minDate $minDate -dateToCheck $PostponementDateInUTC
        if (-not $valid) {
            return
        }
    } else {
        $PostponementDateInUTC = "2025-09-30T23:59:59Z"
        $isDefaultDate = $true
    }

    # If user didn't specify a tenant in params, let them select.
    if (-not $TenantId) {
        try {
            # Have user log into relevant account
            $connected = Connect-AzAccount -ErrorAction Stop
            # Get all tenants the user has access to
            $tenants = Get-AzTenant -ErrorAction Stop
        } catch {
            Write-Host "Failed to connect and/or fetch list of user's tenants. Error: $($_.Exception.Message)" -ForegroundColor Red
            Write-Host
            return
        }
    }
}

```

```

if (-not $tenants) {
    Write-Host "No tenants found for this user." -ForegroundColor Red
    return
}

# Display them as a numbered list
Write-Host "Please select a tenant from the list below"
Write-Host " "
for ($i = 0; $i -lt $tenants.Count; $i++) {
    Write-Host "$($i + 1)) $($tenants[$i].TenantId) - $($tenants[$i].Name)
($($tenants[$i].DefaultDomain))"
}
Write-Host

# Ask user to select one
$selection = Read-Host "Enter the number for the tenant you want to use"

# Validate and extract selected tenant
if ($selection -match '^\\d+$') {
    $selection = [int]$selection
    if ($selection -ge 1 -and $selection -le $tenants.Count) {
        $chosenTenant = $tenants[$selection - 1]
        Write-Host "You selected tenant: $($chosenTenant.TenantId) -
 $($chosenTenant.Name) ($($chosenTenant.DefaultDomain))" -ForegroundColor Green
        Write-Host
        # Use $chosenTenant.TenantId later in the script
        $TenantId = $chosenTenant.TenantId
    } else {
        Write-Host "Number is out of range. Exiting..." -ForegroundColor Red
        return
    }
} else {
    Write-Host "Invalid selection. Exiting..." -ForegroundColor Red
    return
}

if ($isDefaultDate) {
    $newDate = Select-Postponement-Date
    if (-not $newDate) {
        return
    } else {
        $PostponementDateInUTC = $newDate.ToString("yyyy-MM-ddTHH:mm:ssZ")
        $isDefaultDate = $false
    }
}

if ($isDefaultDate) {
    Write-Host "This will update the MFA enforcement date for TenantId:
 '$($TenantId)' to the DEFAULT date of '$($PostponementDateInUTC)'"
} else {
    Write-Host "This will update the MFA enforcement date for TenantId:
 '$($TenantId)' to the date of '$($PostponementDateInUTC)'"
}

```

```

Write-Host
$confirmation = Read-Host "Do you want to continue (Y/N)?"
if ($confirmation -match '^([Yy])') {
    Write-Host "Proceeding..." -ForegroundColor Green
    Write-Host
} else {
    Write-Host "Operation canceled by user." -ForegroundColor Red
    return
}

try {
    $connected = Connect-AzAccount -TenantId $TenantId
} catch {
    Write-Host "Failed to log the user in to specified tenant. Error: $($_.Exception.Message)" -ForegroundColor Red
    Write-Host
    return
}
Start-Sleep -Seconds 3

# Constants
$ELEVATED_TENANT_ADMIN_ROLE_ID =
"/providers/Microsoft.Authorization/roleDefinitions/18d7d88d-d35e-4fb5-a5c3-
7773c20a72d9"
$OWNER_ROLE_ID = "/providers/Microsoft.Authorization/roleDefinitions/8e3af657-
a8ff-443c-a75c-2fe8c4bcb635"

# Get tokens
Write-Host "Fetching necessary authorization tokens..."
try {
    $armToken = (Get-AzAccessToken -ResourceUrl
"https://management.azure.com/").Token
    if ($null -eq $armToken) {
        Write-Host "Failed to fetch an authorization token for Azure Resource Manager. Make sure you run: Connect-AzAccount -TenantId '<your tenant id>'" -ForegroundColor Red
        return
    }

    Start-Sleep -Seconds 3
} catch {
    Write-Host "Failed to fetch Azure Resource Manager token. Error: $($_.Exception.Message)" -ForegroundColor Red
    Write-Host
    return
}

try {
    $coreToken = (Get-AzAccessToken -ResourceUrl
"https://management.core.windows.net/").Token
    if ($null -eq $coreToken) {
        Write-Host "Failed to fetch an authorization token for Azure Resource Manager core. Make sure you run: Connect-AzAccount -TenantId '<your tenant id>'" -ForegroundColor Red
        return
    }
}

```

```

        return
    }

    Start-Sleep -Seconds 3
} catch {
    Write-Host "Failed to fetch Azure Resource Manager token. Error:
$($_.Exception.Message)" -ForegroundColor Red
    Write-Host
    return
}

$armClaims = Decode-JwtPayload -Jwt $armToken
objectId = $armClaims.oid
if ($null -eq objectId) {
    Write-Host "Failed to parse objectId from oid claim in Azure Resource
Manager token. Make sure you are an admin of this tenant." -ForegroundColor Red
    return
}

Write-Host "Successfully fetched authorization tokens." -ForegroundColor Green
Write-Host

# Check elevated access
try {
    $roleCheckUri =
"https://management.azure.com/providers/Microsoft.PortalServices/providers/Microso
ft.Authorization/roleAssignments?api-version=2022-04-01&$filter=principalId eq
'$objectId'"
    $roleAssignments = Invoke-RestMethod -Headers @{Authorization = "Bearer
$armToken"} -Uri $roleCheckUri -Method GET

    Start-Sleep -Seconds 3
} catch {
    Write-Host "Failed to check user's elevated access. Error:
$($_.Exception.Message)" -ForegroundColor Red
    Write-Host
    return
}

Write-Host "Checking for elevated access..."
$hasElevatedAccess = $false
foreach ($item in $roleAssignments.value) {
    if ($item.properties.roleDefinitionId -eq $ELEVATED_TENANT_ADMIN_ROLE_ID)
{
        $hasElevatedAccess = $true
    }
}

# Used to determine whether or not to delete elevated access at end
$alreadyHadElevatedStatus = $hasElevatedAccess

if (-not $hasElevatedAccess) {
    Write-Host "User does NOT have elevated access. Elevating access..."
    $elevateUri =
"https://management.azure.com/providers/Microsoft.Authorization/elevateAccess?api-

```

```

version=2017-05-01"

try {
    # Attempt the API call and capture the response
    $response = Invoke-RestMethod -Headers @{Authorization = "Bearer
$coreToken"} -Uri $elevateUri -Method POST -ErrorAction Stop

        # Even if there's no content in the response, the request could still
have succeeded
        Write-Host "Successfully elevated access." -ForegroundColor Green
        Write-Host

        Start-Sleep -Seconds 3
} catch {
    Write-Host "Failed to elevate access. Error: $($_.Exception.Message)"
    Write-Host "Make sure you are already a tenant admin"
    return
}
} else {
    Write-Host "User already has elevated access." -ForegroundColor Green
    Write-Host
}

try {
    # Re-check role assignments after possible elevation
    $roleAssignments = Invoke-RestMethod -Headers @{Authorization = "Bearer
$armToken"} -Uri $roleCheckUri -Method GET

    Start-Sleep -Seconds 3
} catch {
    Write-Host "Failed to re-check user's elevated access. Error:
$($_.Exception.Message)" -ForegroundColor Red
    Write-Host

    # Clean up elevated access if we added it
    if (-not $alreadyHadElevatedStatus) {
        Remove-ElevatedAccess -objectId $objectId -TenantId $TenantId
    }
    return
}

Write-Host "Checking if owner role exists..."
$hasOwnerRole = $false
foreach ($item in $roleAssignments.value) {
    if ($item.properties.roleDefinitionId -eq $OWNER_ROLE_ID) {
        $hasOwnerRole = $true
    }
}

try {
    if (-not $hasOwnerRole) {
        Write-Host "Owner role does NOT exist. Assigning Owner Role..."

        # register provider
        $regProviderUri =

```

```

"https://management.azure.com/providers/Microsoft.PortalServices/register?api-
version=2024-03-01"
    try {
        $providerRegistered = Invoke-RestMethod -Headers @{Authorization =
"Bearer $armToken"} -Uri $regProviderUri -Method POST

        Start-Sleep -Seconds 3
    } catch {
        Write-Host "Failed to register PortalServices provider. Error:
$($_.Exception.Message)" -ForegroundColor Red
        Write-Host
        throw "Provider registration failed"
    }

    # assign owner role
    $assignmentId = [guid]::NewGuid()
    $assignUri =
"https://management.azure.com/providers/Microsoft.PortalServices/providers/Microso
ft.Authorization/roleAssignments/$($assignmentId)?api-version=2020-04-01-preview"
    $assignBody = @{
        properties = @{
            roleDefinitionId = $OWNER_ROLE_ID
            principalId = $objectId
            principalType = "User"
            scope = "/providers/Microsoft.PortalServices"
        }
    } | ConvertTo-Json -Depth 5
    try {
        Invoke-RestMethod -Headers @{Authorization = "Bearer $armToken";
"Content-Type" = "application/json"} `

        -Uri $assignUri -Method PUT -Body $assignBody
        Start-Sleep -Seconds 3
        Write-Host "Successfully assigned owner role." -ForegroundColor
Green
        } catch {
            Write-Host "Failed to assign owner role. Error:
$($_.Exception.Message)" -ForegroundColor Red
            Write-Host
            throw "Owner role assignment failed"
        }
    } else {
        Write-Host "Owner role already exists."
        Write-Host
    }

    # Update Tenant Settings
    Write-Host "Trying to postpone MFA enforcement..."
    $settingsUri =
"https://management.azure.com/providers/Microsoft.PortalServices/settings/default?
api-version=2024-09-01-preview"
    $settingsBody = @{
        properties = @{
            multiFactorAuthentication = @{
                portalEnforcement = "OptOut"
                portalJustification = "Postponed MFA by user with Powershell

```

```

script"
    portalEnforcementDate = $PostponementDateInUTC
}
}
} | ConvertTo-Json -Depth 5

$successfulUpdate = $false
try {
    $updateResults = Invoke-WebRequest -Headers @{Authorization = "Bearer
$armToken"; "Content-Type" = "application/json"} `

-Uri $settingsUri -Method PUT -Body $settingsBody

Start-Sleep -Seconds 3
} catch {
    Write-Host "Failed to postpone MFA. Error: $($_.Exception.Message)" -
ForegroundColor Red
    Write-Host
    throw "MFA postponement failed"
}

if ($updateResults.StatusCode -ge 200 -and $updateResults.StatusCode -lt
300) {
    # Convert content to JSON
    $jsonResponse = $updateResults.Content | ConvertFrom-Json

    # Check if provisioningState is 'Succeeded'
    if ($jsonResponse.properties.provisioningState -eq "Succeeded") {
        Write-Host "Successfully postponed MFA to
 $($PostponementDateInUTC)." -ForegroundColor Green
        Write-Host
        $successfulUpdate = $true
    } else {
        Write-Host "Provisioning state is not Succeeded. It is
 $($jsonResponse.properties.provisioningState)." -ForegroundColor Red
        Write-Host
        throw "MFA postponement failed - incorrect provisioning state"
    }
} else {
    Write-Host "Request failed with status: $($updateResults.StatusCode)" -
ForegroundColor Red
    Write-Host
    throw "MFA postponement failed - incorrect status code"
}

# Optional verification
if ($successfulUpdate) {
    Write-Host "Verifying that postponement date was properly stored..."
    try {
        $verify = Invoke-RestMethod -Headers @{Authorization = "Bearer
$armToken"} `

-Uri $settingsUri -Method GET

Start-Sleep -Seconds 3
    Write-Host "The postponement date of

```

```

'$($verify.properties.multiFactorAuthentication.portalEnforcementDate)' is set for
tenant '$($TenantId)'" -ForegroundColor Green
    Write-Host
} catch {
    Write-Host "Failed to fetch the stored postponement date. Error:
$($_.Exception.Message)" -ForegroundColor Red
    Write-Host
    # Continue despite verification failure as update was successful
}
}
catch {
    Write-Host "An error occurred during the operation:
$($_.Exception.Message)" -ForegroundColor Red
    Write-Host
}
finally {
    # Remove elevated access only if we were the ones that added it in the
script.
    if (-not $alreadyHadElevatedStatus) {
        Remove-ElevatedAccess -objectId $objectId -TenantId $TenantId
    }
}
}

function Remove-ElevatedAccess {
param (
    [string]$objectId,
    [string]$TenantId
)

    Write-Host "Removing temporary elevated access..."
    $roleCheckUri =
"https://management.azure.com/providers/Microsoft.Authorization/roleAssignments?
api-version=2022-04-01&$filter=principalId+eq+'$($objectId)''"
    try {
        $roleAssignments = Invoke-RestMethod -Headers @{Authorization = "Bearer
$armToken"} -Uri $roleCheckUri -Method GET
    } catch {
        Write-Host "Failed to fetch elevated access status. Error:
$($_.Exception.Message)" -ForegroundColor Red
        Write-Host
        return
    }

    $newAssignmentId = $false
    foreach ($item in $roleAssignments.value) {
        if ($item.properties.roleDefinitionId -eq $ELEVATED_TENANT_ADMIN_ROLE_ID)
{
            $newAssignmentId = $($item.name)
        }
    }

    if ($newAssignmentId -eq $false) {
        Write-Host "Could not find the elevated role assignment id. You will need

```

```

to manually delete your elevated status.2" -ForegroundColor Red
    return
}

try {
    $connected = Connect-AzAccount -TenantId $TenantId
} catch {
    Write-Host "Failed re-connect user. You will need to manually delete your
elevated status. Error: $($_.Exception.Message)" -ForegroundColor Red
    Write-Host
    return
}

Write-Host "Refreshing authorization tokens..."
Start-Sleep -Seconds 3
try {
    $coreToken = (Get-AzAccessToken -ResourceUrl
"https://management.core.windows.net/").Token
    if ($null -eq $coreToken) {
        Write-Host "Failed to fetch an authorization token for Azure Resource
Manager core. You will need to manually delete your elevated status." -
ForegroundColor Red
        return
    }
} catch {
    Write-Host "Failed to refresh authorization tokens. You will need to
manually delete your elevated status. Error: $($_.Exception.Message)" -
ForegroundColor Red
    Write-Host
    return
}

$retryCount = 0
$maxRetries = 3

do {
    $result = Delete-Elevated-Access -roleAssignmentId $newAssignmentId -
coreToken $coreToken -retryCount $retryCount
    if ($result -eq $false) {
        $retryCount = $retryCount + 1
    } else {
        return
    }
} while ($retryCount -lt $maxRetries)

if ($retryCount -ge $maxRetries) {
    Write-Host "Failed to remove elevated access. You will need to manually
delete your elevated status. " -ForegroundColor Red
    Write-Host
    return
}
}

function Delete-Elevated-Access {
param (

```

```

        [string]$roleAssignmentId,
        [string]$coreToken,
        [int]$retryCount
    )

    try {
        $deleteUri =
"https://management.azure.com/providers/Microsoft.Authorization/roleAssignments/"
+ $roleAssignmentId + "?api-version=2018-07-01"
        # Attempt the API call and capture the response
        $response = Invoke-RestMethod -Headers @{Authorization = "Bearer
$coreToken"} -Uri $deleteUri -Method DELETE -ErrorAction Stop

        # Even if there's no content in the response, the request could still have
succeeded
        Write-Host "Successfully removed elevated access." -ForegroundColor Green
        Write-Host

        Start-Sleep -Seconds 3
        return $true
    } catch {
        Write-Host "(Attempt # $($retryCount)): Failed to remove elevated access.
Error: $($_.Exception.Message)" -ForegroundColor Yellow
        Start-Sleep -Seconds 3
        return $false
    }
}

function Decode-JwtPayload {
    param (
        [string]$Jwt
    )

    $parts = $Jwt -split '\.'
    if ($parts.Count -lt 2) {
        throw "Invalid JWT format"
    }

    $payload = $parts[1]

    # Replace URL-safe base64 chars
    $payload = $payload.Replace('-', '+').Replace('_', '/')

    # Add padding if needed
    switch ($payload.Length % 4) {
        2 { $payload += '==' }
        3 { $payload += '=' }
        1 { throw "Invalid base64url string" }
    }

    $json =
[System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String($payload))
    return $json | ConvertFrom-Json
}

```

```

function Check-Date-Is-Valid {
    param (
        [DateTime]$maxDate,
        [DateTime]$minDate,
        [string]$dateToCheck
    )

    $inputDate = $maxDate
    if ([string]::IsNullOrEmpty($dateToCheck)) {
        Write-Host "No input provided. Please enter a date in the required
format.`n" -ForegroundColor Red
        return $valid
    }

    $parsed = [DateTime]::TryParse($dateToCheck, [ref]$inputDate)
    if (-not $parsed) {
        Write-Host "Invalid date format. Please try again using format like 2025-
09-15T00:00:00Z.`n" -ForegroundColor Red
        return $valid
    }

    $inputDate = $inputDate.ToUniversalTime()
    if ($inputDate -ge $minDate -and $inputDate -le $maxDate) {
        return $inputDate
    } else {
        Write-Host "Date must be between $($minDate.ToString("u")) and
 $($maxDate.ToString("u")) (UTC). Try again.`n" -ForegroundColor Red
    }

    return $valid
}

function Select-Postponement-Date {
    $valid = $false
    $today = [DateTime]::UtcNow.Date
    $minDate = $today.AddDays(1)
    $maxDate = [DateTime]::ParseExact("2025-09-30T23:59:59Z", "yyyy-MM-
ddTHH:mm:ssz", $null).ToUniversalTime()

    $inputDate = $maxDate
    while (-not $valid) {
        $inputDateStr = Read-Host "Enter a UTC date up to 2025-09-30T23:59:59Z
(e.g., 2025-09-15T00:00:00Z) or Enter to use the default"
        $defaultChosen = $false
        if ([string]::IsNullOrEmpty($inputDateStr)) {
            $inputDateStr = "2025-09-30T23:59:59Z"
            $defaultChosen = $true
        }

        $inputDate = Check-Date-Is-Valid -maxDate $maxDate -minDate $minDate -
dateToCheck $inputDateStr

        if (-not $inputDate) {
            $valid = $false
        } else {

```

```
$valid = $true
if ($defaultChosen) {
    Write-Host "You chose the enforcement date: $($inputDateStr)" -
ForegroundColor Green
} else {
    Write-Host "You entered the enforcement date: $($inputDateStr)" -
ForegroundColor Green
}
Write-Host
}

return $inputDate
}

# Call the function
Set-TenantSettingsMFAPostponement -TenantId $TenantId -PostponementDateInUTC
$PostponementDateInUTC
```

Related content

- [Planning for mandatory MFA for Azure and other admin portals](#)
- [How to verify that users are set up for mandatory MFA](#)

Satisfy Microsoft Entra ID multifactor authentication (MFA) controls with MFA claims from a federated IdP

Article • 03/04/2025

This document outlines the assertions Microsoft Entra ID requires from a [federated identity provider \(IdP\)](#) to honor configured [federatedIdpMfaBehaviour](#) values of `acceptIfMfaDoneByFederatedIdp` and `enforceMfaByFederatedIdp` for Security Assertions Markup Language (SAML) and WS-Fed federation.

💡 Tip

Configuring Microsoft Entra ID with a federated IdP is [optional](#). Microsoft Entra recommends [authentication methods](#) available in Microsoft Entra ID.

- Microsoft Entra ID includes support for authentication methods previously only available via a federated IdP such as certificate/smartcards with [Entra Certificate Based Authentication](#)
- Microsoft Entra ID includes support for integrating 3rd party MFA providers with [External Authentication Methods](#)
- Applications integrated with a federated IdP can be [integrated directly with Microsoft Entra ID](#)

Using WS-Fed or SAML 1.1 federated IdP

When an admin optionally configures their Microsoft Entra tenant to use a [federated IdP](#) using WS-Fed federation, Microsoft Entra redirects to IdP for authentication and expect a response in the form of a Request Security Token Response (RSTR) containing a SAML 1.1 assertion. If configured to do so, Microsoft Entra honors MFA done by the IdP if one of the following two claims is present:

- `http://schemas.microsoft.com/claims/multipleauthn`
- `http://schemas.microsoft.com/claims/wiaormultiauthn`

They can be included in the assertion as part of the `AuthenticationStatement` element. For example:

XML

```
<saml:AuthenticationStatement  
    AuthenticationMethod="http://schemas.microsoft.com/claims/multipleauthn"  
...>  
    <saml:Subject> ... </saml:Subject>  
</saml:AuthenticationStatement>
```

Or they can be included in the assertion as part of the `AttributeStatement` elements. For example:

XML

```
<saml:AttributeStatement>  
    <saml:Attribute AttributeName="authenticationmethod"  
AttributeNamespace="http://schemas.microsoft.com/ws/2008/06/identity/claims"  
>  
        <saml:AttributeValue>...</saml:AttributeValue>  
  
<saml:AttributeValue>http://schemas.microsoft.com/claims/multipleauthn</saml  
:AttributeValue>  
    </saml:Attribute>  
</saml:AttributeStatement>
```

Using sign-in frequency and session control Conditional Access policies with WS-Fed or SAML 1.1

[Sign-in frequency](#) uses `UserAuthenticationInstant` (SAML assertion <http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant>), which is `AuthInstant` of first factor authentication using password for SAML1.1/WS-Fed.

Using SAML 2.0 federated IdP

When an admin optionally configures their Microsoft Entra ID tenant to use a [federated IdP](#) using [SAML/SAML 2.0](#) federation, Microsoft Entra will redirect to the IdP for authentication, and expect a response that contains a SAML 2.0 assertion. The inbound MFA assertions must be present in the `AuthnContext` element of the `AuthnStatement`.

XML

```
<AuthnStatement AuthnInstant="2024-11-22T18:48:07.547Z">  
    <AuthnContext>  
  
<AuthnContextClassRef>http://schemas.microsoft.com/claims/multipleauthn</Aut  
hnContextClassRef>
```

```
</AuthnContext>  
</AuthnStatement>
```

As a result, for inbound MFA assertions to be processed by Microsoft Entra, they **must** be present in the `AuthnContext` element of the `AuthnStatement`. Only one method can be presented in this manner.

Using sign-in frequency and session control Conditional Access policies with SAML 2.0

[Sign-in frequency](#) uses `AuthInstant` of either MFA or First Factor auth provided in the `AuthnStatement`. Any assertions shared in the `AttributeReference` section of the payload are ignored, including

<http://schemas.microsoft.com/ws/2017/04/identity/claims/multifactorauthenticationinstant>.

Related content

[federatedIdpMfaBehaviour](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Find and address gaps in strong authentication coverage for your administrators

Article • 03/04/2025

Requiring multifactor authentication (MFA) for the administrators in your tenant is one of the first steps you can take to increase the security of your tenant. In this article, we'll cover how to ensure all of your administrators are covered by multifactor authentication.

Detect current usage for Microsoft Entra Built-in administrator roles

The [Microsoft Entra ID Secure Score](#) provides a score for **Require MFA for administrative roles** in your tenant. This improvement action tracks the MFA usage of those with [administrator roles](#).

There are different ways to check if your admins are covered by an MFA policy.

- To troubleshoot sign-in for a specific administrator, you can use the sign-in logs. The sign-in logs let you filter **Authentication requirement** for specific users. Any sign-in where **Authentication requirement** is **Single-factor authentication** means there was no multifactor authentication policy that was required for the sign-in.

The screenshot shows the Microsoft Entra ID Sign-in activity log interface. At the top, there are filters: 'Date: Last 24 hours', 'Show dates as: Local', 'Authentication requirement: None Selected', and a 'Add filters' button. Below these, there are two tabs: 'User sign-ins (interactive)' and 'User sign-ins (non-interactive)'. The 'User sign-ins (interactive)' tab is selected. It displays a table with columns: Date, Request ID, User, and a dropdown menu for 'Authentication requirement' which is set to 'Multifactor authentication'. There are four rows of data corresponding to sign-ins on 2/26/2024 at 8:08:13 PM, 8:08:08 PM, 7:50:32 PM, and 7:50:29 PM. Each row shows the user as 'MOD Administrator' and the location as 'Azure Portal'. The status for each sign-in is listed in a separate column: Success, Interrupted, Success, and Success respectively. The IP address, location, conditional access status, and authentication requirement are also listed for each sign-in.

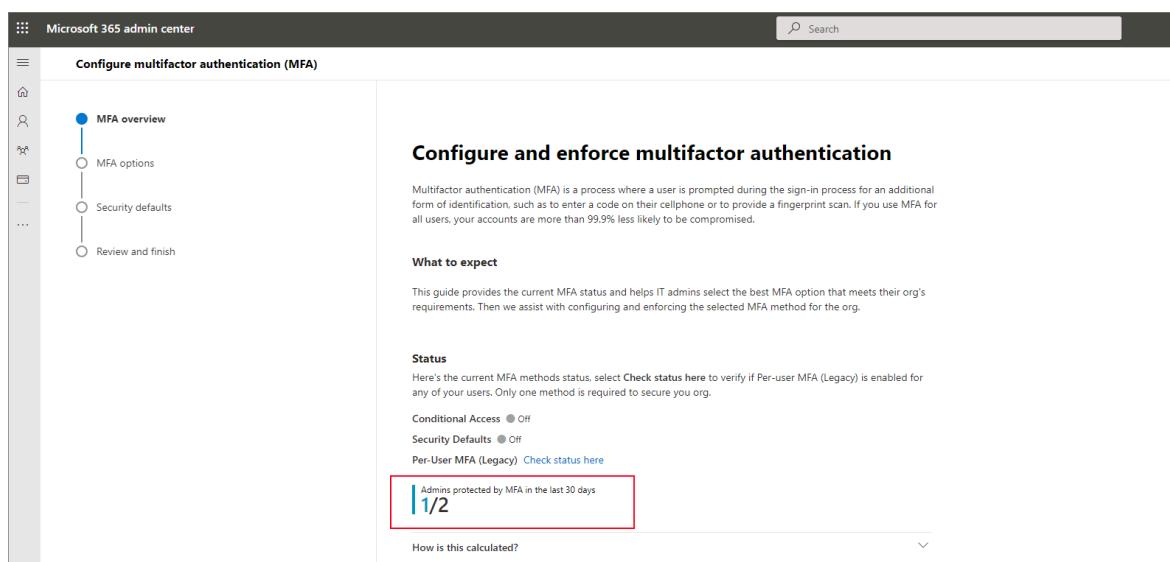
Date	Request ID	User	Authentication requirement	Status	IP address	Location	Conditional Access	Authentication re...
2/26/2024, 8:08:13 PM	MOD	MOD Administrator	Single-factor authentication	Success		Phoenix, Arizona, US	Not Applied	Single-factor authenti...
2/26/2024, 8:08:08 PM	MOD	MOD Administrator	Multifactor authentication	Interrupted		Phoenix, Arizona, US	Not Applied	Single-factor authenti...
2/26/2024, 7:50:32 PM	MOD	MOD Administrator	Multifactor authentication	Success		Phoenix, Arizona, US	Not Applied	Single-factor authenti...
2/26/2024, 7:50:29 PM	MOD	MOD Administrator	Multifactor authentication	Success		Phoenix, Arizona, US	Not Applied	Single-factor authenti...

When viewing the details of a specific sign-in, select the **Authentication details** tab for details about the MFA requirements. For more information, see [Sign-in log activity details](#).

Activity Details: Sign-ins

Basic info	Location	Device info	Authentication Details	Conditional Access	Report-only	Additional Details
Authentication Policies Applied						
Conditional Access						
Date	Authentication met...	Authentication met...	Succeeded	Result detail	Requirements	
10/21/2021, 2:15:29 PM	FIDO2 security key	My FIDO2 Key	true			
10/21/2021, 2:15:29 PM	Previously satisfied		true	MFA requirement satis...		

- To choose which policy to enable based on your user licenses, we have a new MFA enablement wizard to help you [compare MFA policies](#) and see which steps are right for your organization. The wizard shows administrators who were protected by MFA in the last 30 days.



The screenshot shows the Microsoft 365 admin center interface. On the left, there's a navigation sidebar with icons for users, groups, and more. The main content area has a title 'Configure multifactor authentication (MFA)'. On the left side of the main area, there's a tree view with 'MFA overview' selected, followed by 'MFA options', 'Security defaults', and 'Review and finish'. To the right of this tree view is a large section titled 'Configure and enforce multifactor authentication'. It contains a brief description of what MFA is, a 'What to expect' section with a note about the current MFA status, and a 'Status' section. In the 'Status' section, there's a summary box that says 'Admins protected by MFA in the last 30 days | 1/2'. This summary box is highlighted with a red rectangle.

- You can run [this script](#) to programmatically generate a report of all users with directory role assignments who have signed in with or without MFA in the last 30 days. This script will enumerate all active built-in and custom role assignments, all eligible built-in and custom role assignments, and groups with roles assigned.

Enforce multifactor authentication on your administrators

If you find administrators who aren't protected by multifactor authentication, you can protect them in one of the following ways:

- If your administrators are licensed for Microsoft Entra ID P1 or P2, you can [create a Conditional Access policy](#) to enforce MFA for administrators. You can also update this policy to require MFA from users who are in custom roles.
- Run the [MFA enablement wizard](#) to choose your MFA policy.

- If you assign custom or built-in admin roles in [Privileged Identity Management](#), require multifactor authentication upon role activation.

Use Passwordless and phishing resistant authentication methods for your administrators

After your admins are enforced for multifactor authentication and have been using it for a while, it's time to increase authentication strength and use Passwordless and phishing resistant authentication methods:

- [Phone Sign-in \(with Microsoft Authenticator\)](#)
- [FIDO2](#)
- [Windows Hello for Business](#)

You can read more about these authentication methods and their security considerations in [Microsoft Entra authentication methods](#).

Next steps

[Enable passwordless sign-in with Microsoft Authenticator](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) ↗

Enable per-user Microsoft Entra multifactor authentication to secure sign-in events

Article • 03/04/2025

To secure user sign-in events in Microsoft Entra ID, you can require Microsoft Entra multifactor authentication (MFA). The best way to protect users with Microsoft Entra MFA is to create a Conditional Access policy. Conditional Access is a Microsoft Entra ID P1 or P2 feature that lets you apply rules to require MFA as needed in certain scenarios. To get started using Conditional Access, see [Tutorial: Secure user sign-in events with Microsoft Entra multifactor authentication](#).

For Microsoft Entra ID Free tenants without Conditional Access, you can [use security defaults to protect users](#). Users are prompted for MFA as needed, but you can't define your own rules to control the behavior.

If needed, you can instead enable each account for per-user Microsoft Entra MFA. When you enable users individually, they perform MFA each time they sign in. You can enable exceptions, such as when they sign in from trusted IP addresses, or when the **remember MFA on trusted devices** feature is turned on.

Changing [user states](#) isn't recommended unless your Microsoft Entra ID licenses don't include Conditional Access and you don't want to use security defaults. For more information on the different ways to enable MFA, see [Features and licenses for Microsoft Entra multifactor authentication](#).

Important

This article details how to view and change the status for per-user Microsoft Entra multifactor authentication. If you use Conditional Access or security defaults, you don't review or enable user accounts using these steps.

Enabling Microsoft Entra multifactor authentication through a Conditional Access policy doesn't change the state of the user. Don't be alarmed if users appear disabled. Conditional Access doesn't change the state.

Don't enable or enforce per-user Microsoft Entra multifactor authentication if you use Conditional Access policies.

Microsoft Entra multifactor authentication user states

A user's state reflects whether an Authentication Administrator enrolled them in per-user Microsoft Entra multifactor authentication. User accounts in Microsoft Entra multifactor authentication have the following three distinct states:

[\[+\] Expand table](#)

State	Description	Legacy authentication affected	Browser apps affected	Modern authentication affected
Disabled	The default state for a user not enrolled in per-user Microsoft Entra multifactor authentication.	No	No	No
Enabled	The user is enrolled in per-user Microsoft Entra multifactor authentication, but can still use their password for legacy authentication. If the user has no registered MFA authentication methods, they receive a prompt to register the next time they sign in using modern authentication (such as when they sign in on a web browser).	No. Legacy authentication continues to work until the registration process is completed.	Yes. After the session expires, Microsoft Entra multifactor authentication registration is required.	Yes. After the access token expires, Microsoft Entra multifactor authentication registration is required.
Enforced	The user is enrolled per-user in Microsoft Entra multifactor authentication. If the user has no registered authentication methods, they receive a prompt to register the next time they sign in using modern authentication (such as when they sign in on a web browser). Users who complete	Yes. Apps require app passwords.	Yes. Microsoft Entra multifactor authentication is required at sign-in.	Yes. Microsoft Entra multifactor authentication is required at sign-in.

State	Description	Legacy authentication affected	Browser apps affected	Modern authentication affected
	registration while they're <i>Enabled</i> are automatically moved to the <i>Enforced</i> state.			

All users start out *Disabled*. When you enroll users in per-user Microsoft Entra multifactor authentication, their state changes to *Enabled*. When enabled users sign in and complete the registration process, their state changes to *Enforced*. Administrators may move users between states, including from *Enforced* to *Enabled* or *Disabled*.

ⓘ Note

If per-user MFA is re-enabled on a user and the user doesn't re-register, their MFA state doesn't transition from *Enabled* to *Enforced* in MFA management UI. The administrator must move the user directly to *Enforced*.

View the status for a user

The per-user MFA administration experience in the Microsoft Entra admin center is recently improved. To view and manage user states, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Identity > Users > All users**.
3. Select a user account, and select **User MFA settings**.
4. After you make any changes, select **Save**.

Microsoft Azure

Home > Per-user multifactor authentication ...

Bulk update Got feedback?

This is the new per-user MFA management experience. For the legacy experience please click [here](#).

Users Service settings

Use multifactor authentication (MFA) to protect your users and data. Our recommended approach to enforce MFA is to use adaptive Conditional Access policies. [Learn more](#)

Before you begin, take a look at the [multifactor authentication deployment guide](#).

Enable MFA Disable MFA Enforce MFA User MFA settings

Search Status : All View : Sign-in allowed users Reset filters

Name	UPN	Status
Alain		Disabled
Amy		Disabled
AzureAdmin		Disabled
Bala Sandhu		Disabled
Bala Sandhu		Disabled
Barclay		Disabled
Bill		Disabled
Carla Costa		Disabled
Celeste		Disabled
Dickson		Disabled
Eunice		Disabled
Faith		Disabled
Faith		Disabled

Save Discard

User MFA settings

Require selected users to provide contact methods again
 Delete all existing app passwords generated by the selected users
 Restore multifactor authentication on all remembered devices

If you try to sort thousands of users, the result might gracefully return **There are no users to display**. Try to enter more specific search criteria to narrow the search, or apply specific **Status** or **View** filters.

test11 X Status : All View : Sign-in allowed users

Name ↑ UPN Status

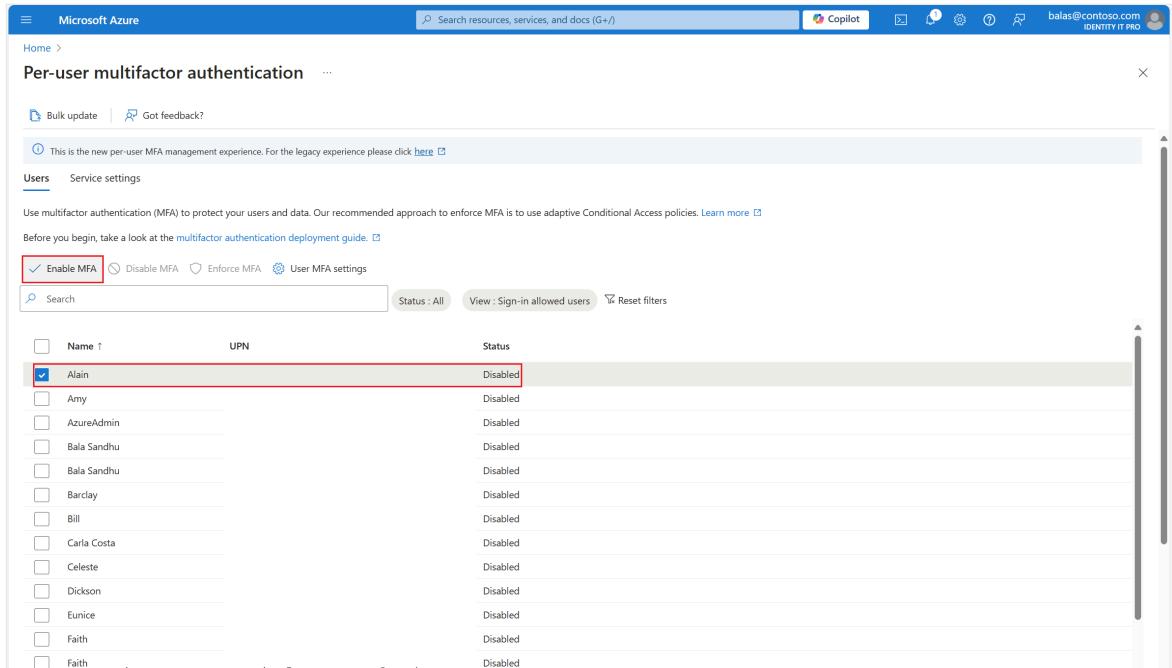
Name	UPN	Status
test11		Disabled

Change the status for a user

To change the per-user Microsoft Entra multifactor authentication state for a user, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Identity > Users > All users**.

3. Select a user account, and select **Enable MFA**.



The screenshot shows the Microsoft Azure portal interface for managing per-user multifactor authentication. At the top, there's a navigation bar with 'Microsoft Azure', a search bar, and various icons. Below it, a header says 'Per-user multifactor authentication'. A note indicates this is the new experience, with a link to the legacy experience. There are tabs for 'Users' (which is selected) and 'Service settings'. Below the tabs, a message encourages using adaptive Conditional Access policies. A 'Before you begin' section links to a deployment guide. A table lists users with columns for Name, UPN, and Status. The 'Enable MFA' checkbox is checked for 'Alain', and his row is highlighted with a red border.

Name ↑	UPN	Status
<input checked="" type="checkbox"/> Alain		Disabled
<input type="checkbox"/> Amy		Disabled
<input type="checkbox"/> AzureAdmin		Disabled
<input type="checkbox"/> Bala Sandhu		Disabled
<input type="checkbox"/> Bala Sandhu		Disabled
<input type="checkbox"/> Barclay		Disabled
<input type="checkbox"/> Bill		Disabled
<input type="checkbox"/> Carla Costa		Disabled
<input type="checkbox"/> Celeste		Disabled
<input type="checkbox"/> Dickson		Disabled
<input type="checkbox"/> Eunice		Disabled
<input type="checkbox"/> Faith		Disabled
<input type="checkbox"/> Faith		Disabled

💡 Tip

Enabled users are automatically switched to *Enforced* when they register for Microsoft Entra multifactor authentication. Don't manually change the user state to *Enforced* unless the user is already registered or if it is acceptable for the user to experience interruption in connections to legacy authentication protocols.

4. Confirm your selection in the pop-up window that opens.

After you enable users, notify them by email. Tell the users that a prompt is displayed to ask them to register the next time they sign in. If your organization uses applications that don't run in a browser or support modern authentication, you can create application passwords. For more information, see [Enforce Microsoft Entra multifactor authentication with legacy applications using app passwords](#).

Use Microsoft Graph to manage per-user MFA

You can manage per-user MFA settings by using the Microsoft Graph REST API Beta. You can use the [authentication resource type](#) to expose authentication method states for users.

To manage per-user MFA, use the `perUserMfaState` property within `users/id/authentication/requirements`. For more information, see [strongAuthenticationRequirements resource type](#).

View per-user MFA state

To retrieve the per-user multifactor authentication state for a user:

HTTP

```
GET /users/{id | userPrincipalName}/authentication/requirements
```

For example:

HTTP

```
GET https://graph.microsoft.com/beta/users/071cc716-8147-4397-a5ba-b2105951cc0b/authentication/requirements
```

If the user is enabled for per-user MFA, the response is:

HTTP

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "perUserMfaState": "enforced"
}
```

For more information, see [Get authentication method states](#).

Change MFA state for a user

To change multifactor authentication state for a user, use the user's strongAuthenticationRequirements. For example:

HTTP

```
PATCH https://graph.microsoft.com/beta/users/071cc716-8147-4397-a5ba-b2105951cc0b/authentication/requirements
Content-Type: application/json

{
  "perUserMfaState": "disabled"
}
```

If successful, the response is:

HTTP

HTTP/1.1 204 No Content

For more information, see [Update authentication method states](#).

Next steps

To configure Microsoft Entra multifactor authentication settings, see [Configure Microsoft Entra multifactor authentication settings](#).

To manage user settings for Microsoft Entra multifactor authentication, see [Manage user settings with Microsoft Entra multifactor authentication](#).

To understand why a user was prompted or not prompted to perform MFA, see [Microsoft Entra multifactor authentication reports](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) ↗

How to manage hardware OATH tokens in Microsoft Entra ID (Preview)

Article • 03/26/2025

This topic covers how to manage hardware oath tokens in Microsoft Entra ID, including Microsoft Graph APIs that you can use to upload, activate, and assign hardware OATH tokens.

Enable hardware OATH tokens in the Authentication methods policy

You can view and enable hardware OATH tokens in the Authentication methods policy by using Microsoft Graph APIs or the Microsoft Entra admin center.

- To view the hardware OATH tokens policy status by using the APIs:

```
https  
  
GET  
https://graph.microsoft.com/beta/policies/authenticationMethodsPolicy/a  
uthenticationMethodConfigurations/hardwareOath
```

- To enable hardware OATH tokens policy by using the APIs.

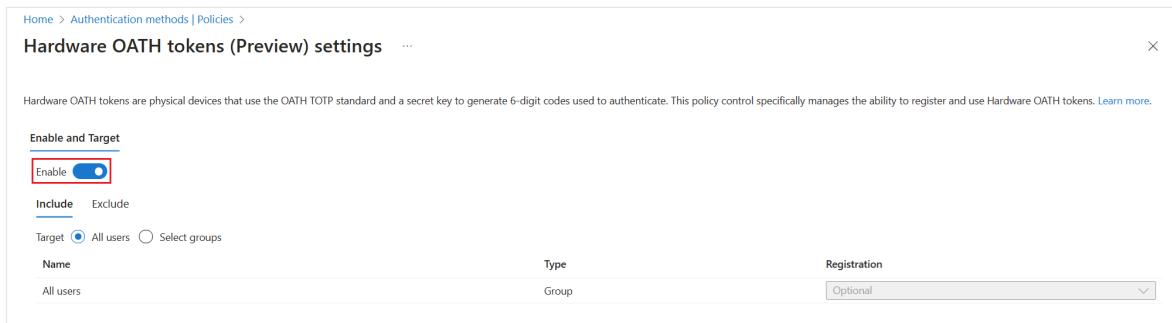
```
https  
  
PATCH  
https://graph.microsoft.com/beta/policies/authenticationMethodsPolicy/a  
uthenticationMethodConfigurations/hardwareOath
```

In the request body, add:

```
https  
  
{  
  "state": "enabled"  
}
```

To enable hardware OATH tokens in the Microsoft Entra admin center:

1. Sign in to the Microsoft Entra admin center [↗](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Protection > Authentication methods > Hardware OATH tokens (Preview)**.
3. Select **Enable**, choose which groups of users to include in the policy, and select **Save**.



We recommend that you [migrate to the Authentication methods policy](#) to manage hardware OATH tokens. If you enable OATH tokens in the legacy MFA policy, browse to the policy in the Microsoft Entra admin center as an Authentication Policy Administrator: **Protection > Multifactor authentication > Additional cloud-based multifactor authentication settings**. Clear the checkbox for **Verification code from mobile app or hardware token**.

Scenario: Admin creates, assigns, and activates a hardware OATH token

This scenario covers how to create, assign, and activate a hardware OATH token as an admin, including the necessary API calls and verification steps. For more information about the permissions required to invoke these APIs and to inspect the request-response samples, see [Create hardwareOathTokenAuthenticationMethodDevice](#).

Note

There might be up to a 20-minute delay for the policy propagation. Allow an hour for the policy to update before users can sign in with their hardware OATH token and see it in their [Security info](#) [↗](#).

Let's look at an example where an Authentication Policy Administrator creates a token and assigns it to a user. You can allow assignment without activation.

For the body of the POST in this example, you can find the **serialNumber** from your device and the **secretKey** is delivered to you.

```
https  
  
POST  
https://graph.microsoft.com/beta/directory/authenticationMethodDevices/hardw  
areOathDevices  
{  
    "serialNumber": "GALT11420104",  
    "manufacturer": "Thales",  
    "model": "OTP 110 Token",  
    "secretKey": "C2dE3fH4iJ5kL6mN7oP1qR2sT3uV4w",  
    "timeIntervalInSeconds": 30,  
    "assignTo": {"id": "00aa00aa-bb11-cc22-dd33-44ee44ee44ee"}  
}
```

The response includes the token **id**, and the user **id** that the token is assigned to:

```
HTTP  
  
{  
    "@odata.context":  
    "https://graph.microsoft.com/beta/$metadata#directory/authenticationMethodDe  
vices/hardwareOathDevices/$entity",  
    "id": "3dee0e53-f50f-43ef-85c0-b44689f2d66d",  
    "displayName": null,  
    "serialNumber": "GALT11420104",  
    "manufacturer": "Thales",  
    "model": "OTP 110 Token",  
    "secretKey": null,  
    "timeIntervalInSeconds": 30,  
    "status": "available",  
    "lastUsedDateTime": null,  
    "hashFunction": "hmacsha1",  
    "assignedTo": {  
        "id": "00aa00aa-bb11-cc22-dd33-44ee44ee44ee",  
        "displayName": "Test User"  
    }  
}
```

Here's how the Authentication Policy Administrator can activate the token. Replace the verification code in the Request body with the code from your hardware OATH token.

```
https  
  
POST https://graph.microsoft.com/beta/users/00aa00aa-bb11-cc22-dd33-  
44ee44ee44ee/authentication/hardwareOathMethods/3dee0e53-f50f-43ef-85c0-  
b44689f2d66d/activate
```

```
{  
    "verificationCode" : "903809"  
}
```

To validate the token is activated, sign in to [Security info](#) as the test user. If you're prompted to approve a sign-in request from Microsoft Authenticator, select Use a verification code.

You can GET to list tokens:

```
https  
  
GET  
https://graph.microsoft.com/beta/directory/authenticationMethodDevices/hardwareOathDevices
```

This example creates a single token:

```
https  
  
POST  
https://graph.microsoft.com/beta/directory/authenticationMethodDevices/hardwareOathDevices
```

In the request body, add:

```
https  
  
{  
    "serialNumber": "GALT11420104",  
    "manufacturer": "Thales",  
    "model": "OTP 110 Token",  
    "secretKey": "abcdef2234567abcdef2234567",  
    "timeIntervalInSeconds": 30,  
    "hashFunction": "hmacsha1"  
}
```

The response includes the token ID.

```
HTTP  
  
##### Response  
{  
    "@odata.context":  
        "https://graph.microsoft.com/beta/$metadata#directory/authenticationMethodDevices/hardwareOathDevices/$entity",  
    "id": "3dee0e53-f50f-43ef-85c0-b44689f2d66d",
```

```
        "displayName": null,  
        "serialNumber": "GALT11420104",  
        "manufacturer": "Thales",  
        "model": "OTP 110 Token",  
        "secretKey": null,  
        "timeIntervalInSeconds": 30,  
        "status": "available",  
        "lastUsedDateTime": null,  
        "hashFunction": "hmacsha1",  
        "assignedTo": null  
    }  
}
```

Authentication Policy Administrators or an end user can unassign a token:

```
https
```

```
DELETE https://graph.microsoft.com/beta/users/66aa66aa-bb77-cc88-dd99-  
00ee00ee00ee/authentication/hardwareoathmethods/6c0272a7-8a5e-490c-bc45-  
9fe7a42fc4e0
```

This example shows how to delete a token with token ID 3dee0e53-f50f-43ef-85c0-b44689f2d66d:

```
https
```

```
DELETE  
https://graph.microsoft.com/beta/directory/authenticationMethodDevices/hardw  
areOathDevices/3dee0e53-f50f-43ef-85c0-b44689f2d66d
```

Scenario: Admin creates and assigns a hardware OATH token that a user activates

In this scenario, an Authentication Policy Administrator creates and assigns a token, and then a user can activate it on their Security info page, or by using Microsoft Graph Explorer. When you assign a token, you can share steps for the user to sign in to [Security info](#) to activate their token. They can choose **Add sign-in method > Hardware token**. They need to provide the hardware token serial number, which is typically on the back of the device.

```
https
```

```
POST  
https://graph.microsoft.com/beta/directory/authenticationMethodDevices/hardw  
areOathDevices  
{
```

```
"serialNumber": "GALT11420104",
"manufacturer": "Thales",
"model": "OTP 110 Token",
"secretKey": "C2dE3fH4iJ5kL6mN7oP1qR2sT3uV4w",
"timeIntervalInSeconds": 30,
"assignTo": {"id": "00aa00aa-bb11-cc22-dd33-44ee44ee44ee"}
}
```

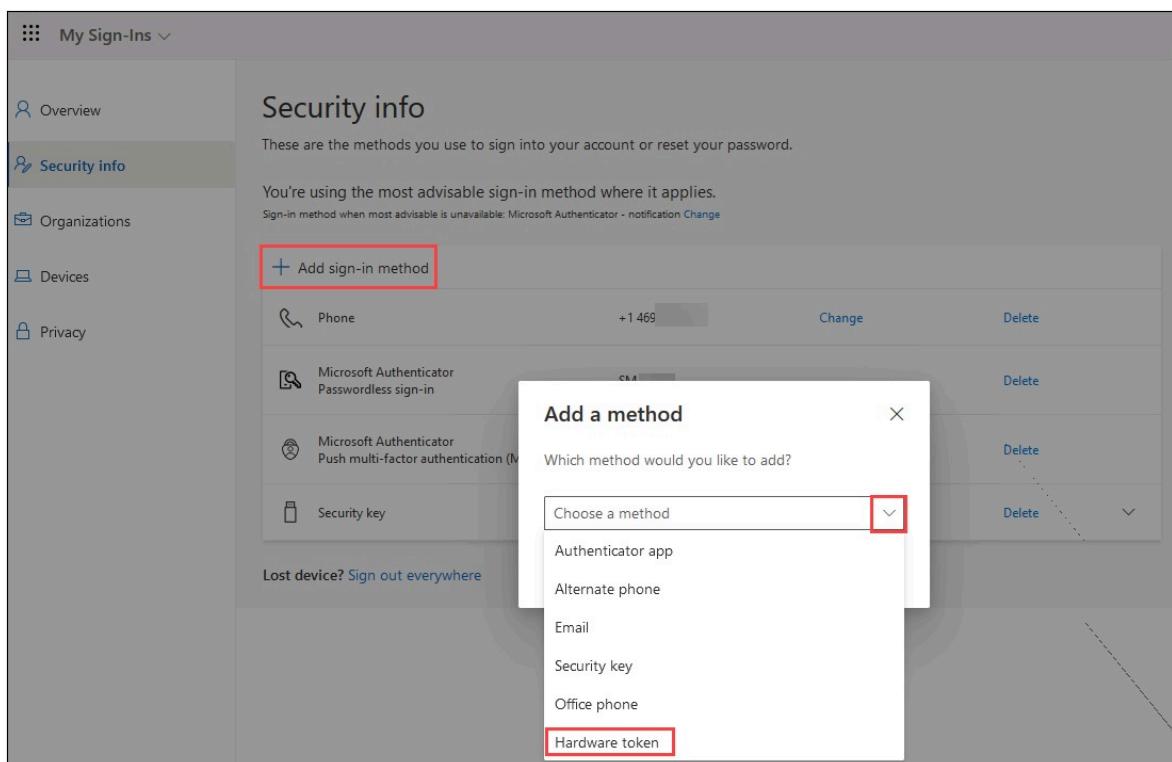
The response includes an **id** value for each token. An Authentication Administrator can assign the token to a user:

```
https
```

```
POST https://graph.microsoft.com/beta/users/00aa00aa-bb11-cc22-dd33-
44ee44ee44ee/authentication/hardwareOathMethods
{
  "device":
  {
    "id": "6c0272a7-8a5e-490c-bc45-9fe7a42fc4e0"
  }
}
```

Here are steps a user can follow to self-activate their hardware OATH token in Security info:

1. Sign in to [Security info](#).
2. Select **Add sign-in method** and choose **Hardware token**.



3. After you select **Hardware token**, select **Add**.

Add a method

X

Which method would you like to add?

Hardware token

Cancel

Add

4. Check the back of the device for the serial number, enter it, and select **Next**.

Hardware token

X

To register the token provided by your organization, start by entering the serial number on your token.

201***

Cancel

Next

5. Create a friendly name to help you choose this method to complete multifactor authentication, and select **Next**.

Hardware token

X

Name your token. This will help to differentiate it from other similar methods.

My Hardware OATH Token

Back

Next

6. Supply the random verification code that appears when you tap the button on the device. For a token that refreshes its code every 30 seconds, you need to enter the code and select **Next** within one minute. For a token that refreshes every 60 seconds, you have two minutes.

Hardware token

Tap the button that's on the token, and enter the 6-digit verification code that appears.

436949

Back

Next

- When you see the hardware OATH token is successfully added, select **Done**.

Hardware token



Your hardware token has been added.

Done

- The hardware OATH token appears in the list of your available authentication methods.

The screenshot shows the 'My Sign-Ins' page in Microsoft Graph Explorer. The left sidebar has a 'Security info' section selected. The main area displays 'Security info' with a list of sign-in methods:

Method	Details	Action
Phone	+1 465	Change Delete
Microsoft Authenticator Passwordless sign-in	SM	Delete
Microsoft Authenticator Push multi-factor authentication (MFA)	SM	Delete
Hardware token	My Hardware OATH Token	Delete
Security key	My Security Key	Delete

A red box highlights the 'Hardware token' row.

Here are steps users can follow to self-activate their hardware OATH token by using Graph Explorer.

- Open Microsoft Graph Explorer, sign in, and consent to the required permissions.

2. Make sure you have the required permissions. For a user to be able to do the self-service API operations, admin consent is required for `Directory.Read.All`, `User.Read.All`, and `User.ReadWrite.All`.
3. Get a list of hardware OATH tokens that are assigned to your account, but not yet activated.

```
https
```

```
GET
```

```
https://graph.microsoft.com/beta/me/authentication/hardwareOathMethods
```

4. Copy the **id** of the token device, and add it to the end of the URL followed by `/activate`. You need to enter the verification code in the request body and submit the POST call before the code changes.

```
https
```

```
POST
```

```
https://graph.microsoft.com/beta/me/authentication/hardwareOathMethods/b65fd538-b75e-4c88-bd08-682c9ce98eca/activate
```

Request body:

```
https
```

```
{
```

```
    "verificationCode": "988659"
```

```
}
```

Scenario: Admin creates multiple hardware OATH tokens in bulk that users self-assign and activate

In this scenario, an Authentication Administrator creates tokens without assignment, and users self-assign and activate the tokens. You can upload new tokens to the tenant in bulk. Users can sign in to [Security info](#) to activate their token. They can choose **Add sign-in method > Hardware token**. They need to provide the hardware token serial number, which is typically on the back of the device.

For greater assurance that the token is only activated by a specific user, you can assign the token to the user, and send the device to them for self-activation.

```
https
```

```
PATCH  
https://graph.microsoft.com/beta/directory/authenticationMethodDevices/hardw  
areOathDevices  
{  
"@context": "#$delta",  
"value": [  
 {  
 "@contentId": "1",  
 "serialNumber": "GALT11420108",  
 "manufacturer": "Thales",  
 "model": "OTP 110 Token",  
 "secretKey": "abcdef2234567abcdef2234567",  
 "timeIntervalInSeconds": 30,  
 "hashFunction": "hmacsha1"  
 },  
 {  
 "@contentId": "2",  
 "serialNumber": "GALT11420112",  
 "manufacturer": "Thales",  
 "model": "OTP 110 Token",  
 "secretKey": "2234567abcdef2234567abcdef",  
 "timeIntervalInSeconds": 30,  
 "hashFunction": "hmacsha1"  
 }  
 ]  
}
```

Troubleshooting hardware OATH token issues

This section covers common

User has two tokens with the same serial number

A user might have two instances of the same hardware OATH token registered as authentication methods. This happens if the legacy token isn't removed from **OATH tokens (Preview)** in the Microsoft Entra admin center after it's uploaded by using Microsoft Graph.

When this happens, both instances of the token are listed as registered for the user:

```
https
```

```
GET https://graph.microsoft.com/beta/users/{user-upn-or-  
objectid}/authentication/hardwareOathMethods
```

Both instances of the token are also listed in **OATH tokens (Preview)** in the Microsoft Entra admin center:

Home > Resource | Security > Security | Multifactor authentication > Multifactor authentication

Multifactor authentication | OATH tokens (Preview) ...

Getting started

Diagnose and solve problems

Settings

Account lockout

Block/unblock users

Fraud alert

Notifications

OATH tokens (Preview)

Phone call settings

Providers

Upload Download Delete Refresh Documentation Columns Got feedback?

Hardware token files uploaded with no errors. View details. →

To get started, select the Upload button above and choose a .csv file. This file should contain the secret keys for the OATH tokens you wish to upload. The file must contain the following columns: Name, Username, Serial Number, Model, Manufacturer, and Activated.

For more information on available authentication and verification methods, view the public documentation.

Username

Enter a user name

Show

All

Name	Username	Serial Number	Model	Manufacturer	Activated
User Name 2	user-name-2@	20033752	SafeID	DeepNet Security	✓
User Name 3	user-name-3@	20033752	SafeID	DeepNet Security	✓

Hardware token status

Delete status Download

2 files uploaded

Hardware token assignment: MyToken.csv
10/24/2024, 12:50:11 PM
Completed
Failures: 0
Successes: 1

To identify and remove the legacy token.

1. List all hardware OATH tokens on the user.

https

```
GET https://graph.microsoft.com/beta/users/{user-upn-or-objectid}/authentication/hardwareOathMethods
```

Find the **id** of both tokens and copy the **serialNumber** of the duplicate token.

2. Identify the legacy token. Only one token is returned in the response of the following command. That token was created by using Microsoft Graph.

https

```
GET  
https://graph.microsoft.com/beta/directory/authenticationMethodDevices/hardwareOathDevices?$filter=serialNumber eq '20033752'
```

3. Remove the legacy token assignment from the user. Now that you know the **id** of the new token, you can identify the **id** of the legacy token from the list returned in step 1. Craft the URL using the legacy token **id**.

https

```
DELETE https://graph.microsoft.com/beta/users/{user-upn-or-objectid}/authentication/hardwareOathMethods/{legacyHardwareOathMethodId}
```

4. Delete the legacy token by using the legacy token **id** in this call.

https

DELETE

<https://graph.microsoft.com/beta/directory/authenticationMethodDevices/hardwareOathDevices/{legacyHardwareOathMethodId}>

Related content

Learn more about [OATH tokens](#). Learn how to [create one or more hardwareOathTokenAuthenticationMethodDevices](#).

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Upload hardware OATH tokens in CSV format

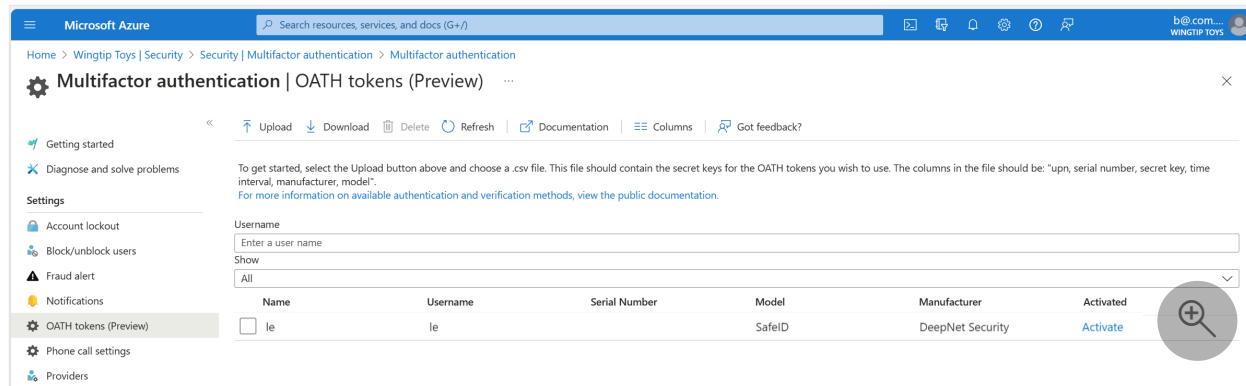
Article • 03/24/2025

Hardware OATH tokens typically come with a secret key, or seed, preprogrammed in the token. Before a user can sign in to their work or school account in Microsoft Entra ID by using a hardware OATH token, an administrator needs to add the token to the tenant.

The recommended way to add the token is by using Microsoft Graph with a least privileged administrator role. For more information, see [Hardware OATH tokens \(preview\)](#).

As an alternative to using Microsoft Graph APIs, tenants with a Microsoft Entra ID Premium license can have a Global Administrator input these keys into Microsoft Entra ID. Secret keys are limited to 128 characters, which isn't compatible with some tokens. The secret key can only contain the characters *a-z* or *A-Z* and digits *2-7*, and must be encoded in *Base32*.

Programmable OATH time-based one-time passcode (TOTP) hardware tokens that can be reseeded can also be set up with Microsoft Entra ID in the software token setup flow.



Name	Username	Serial Number	Model	Manufacturer	Activated
<input type="checkbox"/> le	le	SafeID	DeepNet Security	Activate	

Once tokens are acquired, a Global Administrator must upload them in a comma-separated values (CSV) file format. The file should include the UPN, serial number, secret key, time interval, manufacturer, and model, as shown in the following example:

CSV
upn,serial number,secret key,time interval,manufacturer,model Helga@contoso.com,1234567,2234567abcdef2234567abcdef,60,Contoso,HardwareKey

ⓘ Note

Make sure you include the header row in your CSV file.

Once properly formatted as a CSV file, the Global Administrator can then sign in to the Microsoft Entra admin center, navigate to **Protection > Multifactor authentication > OATH tokens**, and upload the resulting CSV file.

Depending on the size of the CSV file, it can take a few minutes to process. Select the **Refresh** button to get the current status. If there are any errors in the file, you can download a CSV file that lists any errors for you to resolve. The field names in the downloaded CSV file are different than the uploaded version.

Once any errors are addressed, a Privileged Authentication Administrator or an end user can activate a key. Select **Activate** for the token and enter the OTP displayed on the token. You can activate a maximum of 200 OATH tokens every 5 minutes.

Users can have a combination of up to five OATH hardware tokens or authenticator applications, such as the Microsoft Authenticator app, configured for use at any time. Hardware OATH tokens can't be assigned to guest users in the resource tenant.

 **Important**

Make sure to only assign each token to a single user. A single token can't be assigned to multiple users.

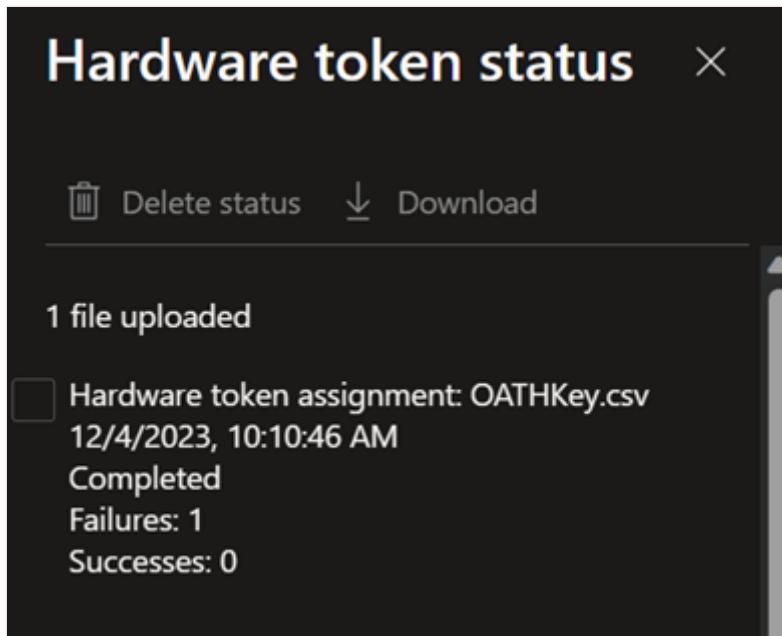
Troubleshooting a failure during upload processing

At times, there may be conflicts or issues that occur with the processing of an upload of the CSV file. If any conflict or issue occurs, you'll receive a notification similar to the following:

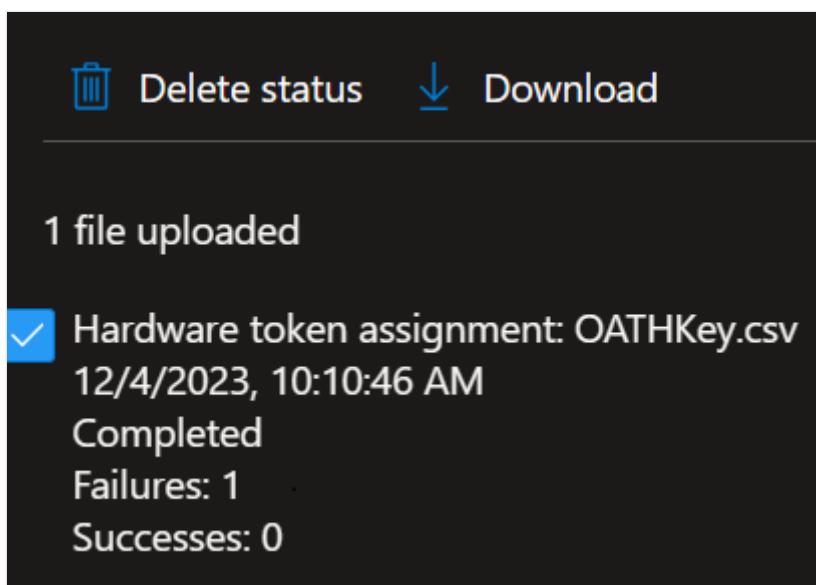


Hardware token files uploaded with errors. You have up to 30 days to fix the errors. [View details.](#) →

To determine the error message, be sure and select **View Details**. The **Hardware token status** blade opens and provides the summary of the status of the upload. It shows that there's been a failure, or multiple failures, as in the following example:

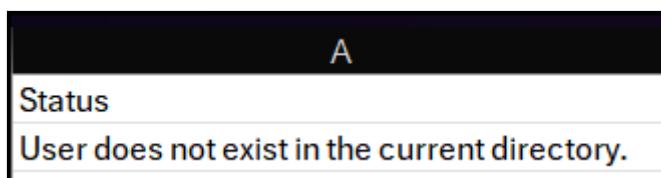


To determine the cause of the failure listed, make sure to click the checkbox next to the status you want to view, which activates the **Download** option. This downloads a CSV file that contains the error identified.



The downloaded file is named **Failures_filename.csv** where *filename* is the name of the file uploaded. It's saved to your default downloads directory for your browser.

This example shows the error identified as a user who doesn't currently exist in the tenant directory:



Once you've addressed the errors listed, upload the CSV again until it processes successfully. The status information for each attempt remains for 30 days. The CSV can

be manually removed by clicking the checkbox next to the status, then selecting **Delete status** if so desired.

Related content

Learn more about [how to manage OATH tokens](#). Learn about [FIDO2 security key providers](#) that are compatible with passwordless authentication.

Feedback

Was this page helpful?

 Yes

 No

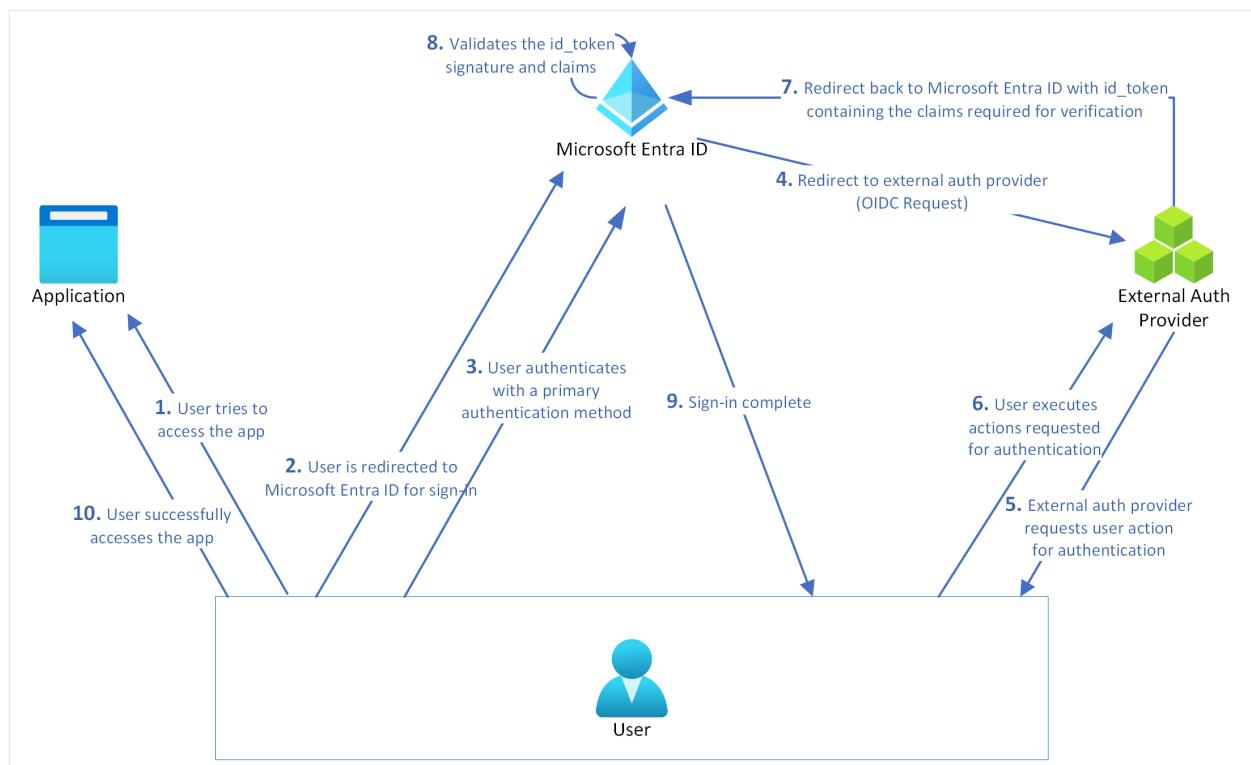
[Provide product feedback ↗](#)

Manage an external authentication method in Microsoft Entra ID (Preview)

Article • 03/04/2025

An external authentication method (EAM) lets users choose an external provider to meet multifactor authentication (MFA) requirements when they sign in to Microsoft Entra ID. An EAM can satisfy MFA requirements from Conditional Access policies, Microsoft Entra ID Protection risk-based Conditional Access policies, Privileged Identity Management (PIM) activation, and when the application itself requires MFA.

EAMs differ from federation in that the user identity is originated and managed in Microsoft Entra ID. With federation, the identity is managed in the external identity provider. EAMs require at least a Microsoft Entra ID P1 license.



Required metadata to configure an EAM

To create an EAM, you need the following information from your external authentication provider:

- An **Application ID** is generally a multitenant application from your provider, which is used as part of the integration. You need to provide admin consent for this application in your tenant.

- A **Client ID** is an identifier from your provider used as part of the authentication integration to identify Microsoft Entra ID requesting authentication.
- A **Discovery URL** is the OpenID Connect (OIDC) discovery endpoint for the external authentication provider.

ⓘ Note

See [Configure a new external authentication provider with Microsoft Entra ID](#) to set up the App registration.

ⓘ Important

Ensure that the kid (Key ID) property is base64-encoded in both the JWT header of the id_token and in the JSON Web Key Set (JWKS) retrieved from the provider's jwks_uri. This encoding alignment is essential for the seamless validation of token signatures during authentication processes. Misalignment can result in issues with key matching or signature validation.

Manage an EAM in the Microsoft Entra admin center

EAMs are managed with the Microsoft Entra ID Authentication methods policy, just like built-in methods.

Create an EAM in the admin center

Before you create an EAM in the admin center, make sure you have the [metadata to configure an EAM](#).

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Privileged Role Administrator](#).
2. Browse to **Protection > Authentication methods > Add external method (Preview)**.

The screenshot shows the Microsoft Entra admin center interface. The left sidebar has sections for Home, Manage (with Policies selected), Monitoring (Activity, User registration details, Registration and reset events, Bulk operation results), and more. The main content area is titled 'Authentication methods | Policies' under 'Contoso - Microsoft Entra ID Security'. It includes a search bar, a 'Manage migration' section with a note about deprecation, and a table of authentication methods. The table has columns for Method, Target, and Enabled. It lists built-in methods like FIDO2 security key, Microsoft Authenticator, SMS, Temporary Access Pass, Hardware OATH tokens (Preview), Third-party software OATH tokens, Voice call, Email OTP, and Certificate-based authentication. It also lists an external method named Adatum.

Method	Target	Enabled
FIDO2 security key	All users, excluding 1 group	Yes
Microsoft Authenticator	All users, excluding 1 group	Yes
SMS		No
Temporary Access Pass	All users, excluding 1 group	Yes
Hardware OATH tokens (Preview)	All users	Yes
Third-party software OATH tokens	1 group	Yes
Voice call		No
Email OTP		No
Certificate-based authentication	2 groups	Yes
Adatum	All users	Yes

Add method properties based on configuration information from your provider.
For example:

- Name: Adatum
- Client ID: 00001111-aaaa-2222-bbbb-3333cccc4444
- Discovery Endpoint: <https://adatum.com/.well-known/openid-configuration>
- App ID: 11112222-bbbb-3333-cccc-4444ddd5555

ⓘ Important

The display name is the name that's shown to the user in the method picker. It can't be changed after the method is created. Display names must be unique.

The screenshot shows the 'Add external method (Preview)' page in the Microsoft Entra admin center. The 'Method Properties' section contains fields for 'Name', 'Client ID', 'Discovery Endpoint', and 'App ID'. Below these are buttons for 'Request admin consent' and 'Request permission'. The 'Enable and target' section has a toggle switch set to 'Off' and a note stating 'Admin consent must be granted in order to enable external authentication methods.' with a 'Learn more' link. The 'Include' tab is selected under 'Enable and target'. A table lists a target named 'All Users' with type 'Group' and registration status 'Optional'. At the bottom are 'Save' and 'Discard' buttons.

You need at least the [Privileged Role Administrator](#) role to grant admin consent for the provider's application. If you don't have the role required to grant consent, you can still save your authentication method, but you can't enable it until consent is granted.

After you enter the values from your provider, press the button to request for admin consent to be granted to the application so that it can read the required info from the user to authenticate correctly. You're prompted to sign in with an account with admin permissions and grant the provider's application with the required permissions.

After you sign in, click **Accept** to grant admin consent:



contoso@example.com

Permissions requested

Review for your organization

Adatum
unverified

This application is not published by Microsoft.

This app would like to:

- ✓ View users' basic profile
- ✓ Maintain access to data that users have given it access to

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at <https://myapps.microsoft.com>. [Show details](#)

Does this app look suspicious? [Report it here](#)

[Cancel](#)

[Accept](#)

You can see the permissions that the provider application requests before you grant consent. After you grant admin consent and the change replicates, the page refreshes to show that admin consent was granted.

The screenshot shows the 'Add external method (Preview)' configuration page in the Microsoft Entra admin center. The 'Method Properties' section contains fields for 'Name' (with a note that it cannot be changed once saved), 'Client ID', 'Discovery Endpoint', and 'App ID' (set to '11112222-bbbb-3333-cccc-4444ddd5555'). The 'Request admin consent' field is checked and shows 'Admin consent granted'. The 'Enable and target' section has 'Enable' set to 'Off', 'Include' selected, and 'Target' set to 'All Users'. A table below shows a single entry: 'Name' (All Users), 'Type' (Group), and 'Registration' (Optional). A note at the bottom states: 'If the application has permissions, then you can also enable the method before saving. Otherwise, you need to save the method in a disabled state, and enable after the application is granted consent.'

If the application has permissions, then you can also enable the method before saving. Otherwise, you need to save the method in a disabled state, and enable after the application is granted consent.

Once the method is enabled, all users in scope can choose the method for any MFA prompts. If the application from the provider doesn't have consent approved, then any sign-in with the method fails.

If the application is deleted or no longer has permission, users see an error and sign-in fails. The method can't be used.

Configure an EAM in the admin center

To manage your EAMs in the Microsoft Entra admin center, open the Authentication methods policy. Select the method name to open the configuration options. You can choose which users are included and excluded from using this method.

The screenshot shows the Microsoft Entra admin center interface. At the top, there's a search bar and a navigation bar with icons for Home, Authentication methods, Policies, and other administrative functions. The main area is titled "Adatum" under "Authentication Methods". A sidebar on the left contains various icons for different administrative tasks. The "Enable and target" tab is selected. Under "Enable", a toggle switch is set to "On". Under "Include", the "All Users" radio button is selected. There's a "Target" section with a "Target *" label and two options: "All Users" (selected) and "Select Targets". Below this is a "Add Target" button. A table lists a single target entry: "All Users" (Type: Group, Registration: Optional). At the bottom are "Save" and "Discard" buttons.

Delete an EAM in the admin center

If you no longer want your users to be able to use the EAM, you can either:

- Set **Enable** to **Off** to save the method configuration
- Click **Delete** to remove the method

The screenshot shows the Microsoft Entra admin center interface, similar to the previous one but with a different focus. It's titled "Adatum" under "Authentication Methods". The "Configure" tab is selected. A section titled "Method Properties" displays the provider information: Name (Adatum), Client ID (00001111-aaaa-2222-bbbb-3333cccc4444), Discovery Endpoint (https://contoso), and App ID (11112222-bbbb-3333-cccc-4444ddd5555). Below this, a "Request admin consent" section shows a green checkmark and the text "Admin consent granted". At the bottom are "Save" and "Discard" buttons.

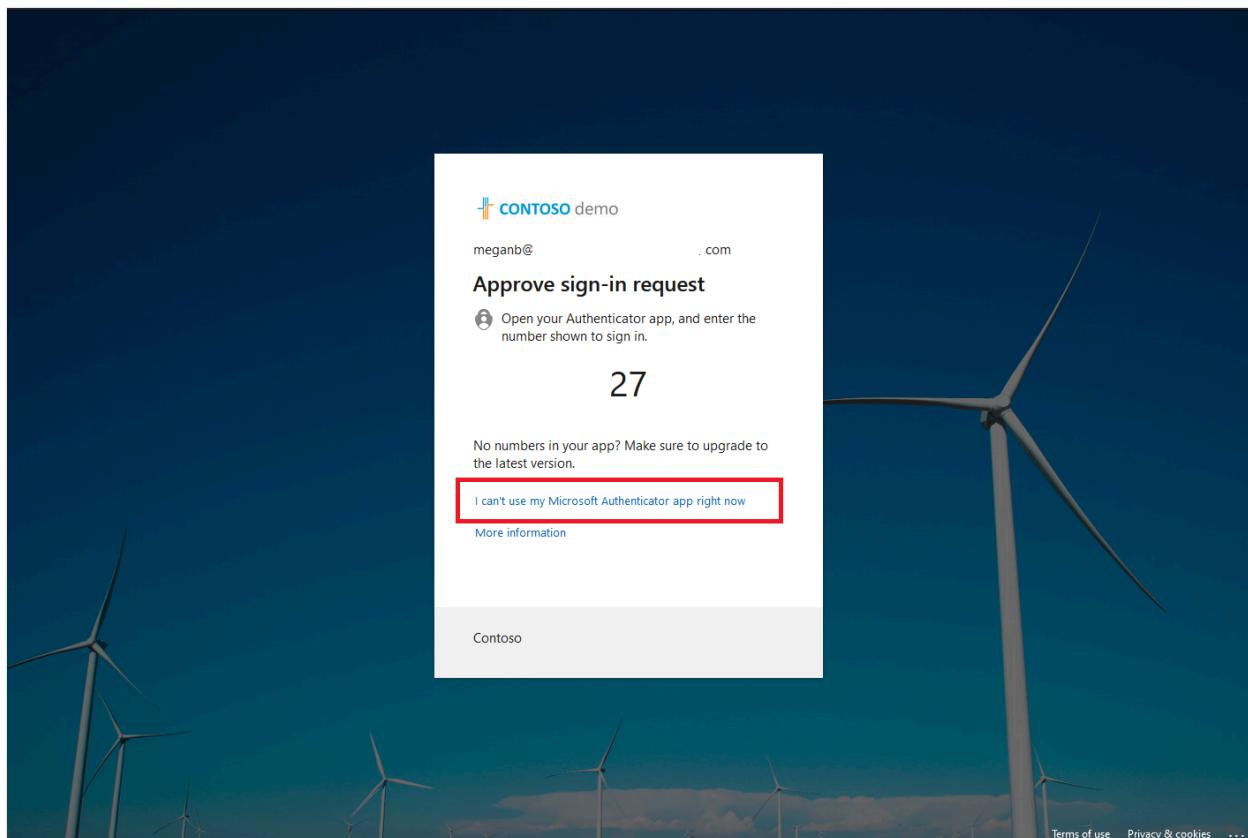
Manage an EAM using Microsoft Graph

To manage the Authentication methods policy by using Microsoft Graph, you need the `Policy.ReadWrite.AuthenticationMethod` permission. For more information, see [Update authenticationMethodsPolicy](#).

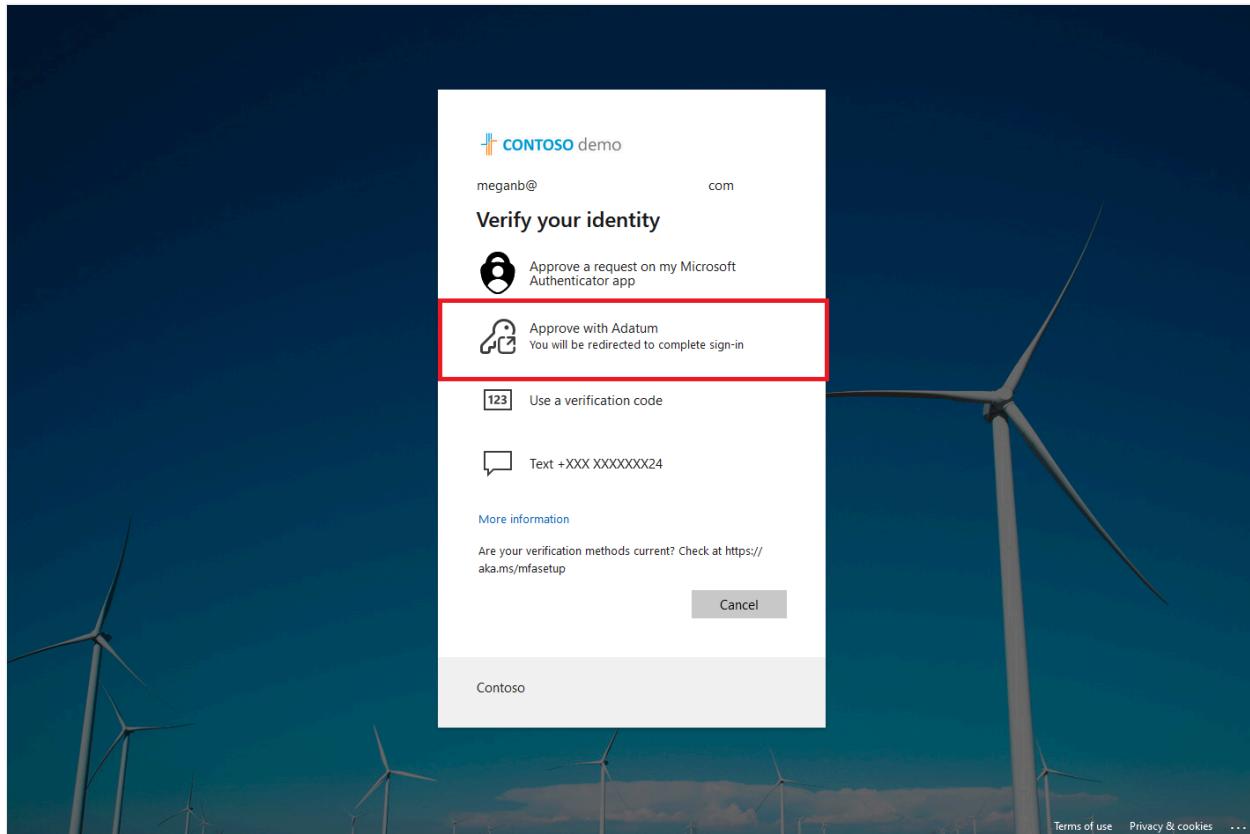
User experience

Users who are enabled for the EAM can use it when they sign-in and multifactor authentication is required.

If the user has other ways to sign in and [system-preferred MFA](#) is enabled, those other methods appear by default order. The user can choose to use a different method, and then select the EAM. For example, if the user has Authenticator enabled as another method, they get prompted for [number matching](#).



If the user has no other methods enabled, they can just choose the EAM. They're redirected to the external authentication provider to complete authentication.



Authentication method registration for EAMs

In the preview, all users in an include group for the EAM are considered MFA capable and can use the external authentication method for satisfying MFA. Users that are MFA-capable due to being an include target for an EAM are not included in reports on authentication method registration.

ⓘ Note

We're actively working on adding registration capability for EAMs. Once registration is added, users that were previously using an EAM will need to have the EAM registered with Entra ID before they will be prompted to use it to satisfy MFA.

Using EAM and Conditional Access custom controls in parallel

EAMs and custom controls can operate in parallel. Microsoft recommends that admins configure two Conditional Access policies:

- One policy to enforce the custom control
- Another policy with the MFA grant required

Include a test group of users for each policy, but not both. If a user is included in both policies, or any policy with both conditions, the user has to satisfy MFA during sign-in. They also have to satisfy the custom control, which makes them redirected to the external provider a second time.

Next steps

For more information about how to manage authentication methods, see [Manage authentication methods for Microsoft Entra ID](#).

For EAM provider reference, see [Microsoft Entra multifactor authentication external method provider reference \(Preview\)](#).

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

When to use a Microsoft Entra multifactor authentication provider

Article • 03/04/2025

ⓘ Important

Effective September 1, 2018 new auth providers may no longer be created. Existing auth providers may continue to be used and updated, but migration is no longer possible. Multifactor authentication continues to be available as a feature in Microsoft Entra ID P1 or P2 licenses.

Two-step verification is available by default for administrators in Microsoft Entra ID, and Microsoft 365 users. However, if you wish to take advantage of [advanced features](#) then you should enable Microsoft Entra multifactor authentication by using Conditional Access. For more information, see [Common Conditional Access policy: Require MFA for all users](#).

A Microsoft Entra multifactor authentication provider is used to take advantage of features provided by Microsoft Entra multifactor authentication for users who **don't have licenses**.

Caveats related to the Microsoft Entra multifactor authentication SDK

Note the SDK is deprecated and calls to the SDK fail after November 14, 2018

What is an MFA provider?

There are two types of Auth providers, and the distinction is around how your Azure subscription is charged. The per-authentication option calculates the number of authentications performed against your tenant in a month. This option is best if some accounts authenticate only occasionally. The per-user option calculates the number of accounts that are eligible to perform MFA, which is all accounts in Microsoft Entra ID, and all enabled accounts in MFA Server. This option is best if some users have licenses but you need to extend MFA to more users beyond your licensing limits.

Manage your MFA provider

You can't change the usage model (per enabled user or per authentication) after an MFA provider is created.

If you purchased enough licenses to cover all users that are enabled for MFA, you can delete the MFA provider altogether.

If your MFA provider isn't linked to a Microsoft Entra tenant, or you link the new MFA provider to a different Microsoft Entra tenant, user settings and configuration options aren't transferred. Also, existing Microsoft Entra multifactor authentication Servers need to be reactivated using activation credentials generated through the MFA Provider.

Removing an authentication provider

Caution

There's no confirmation when deleting an authentication provider. Selecting **Delete** is a permanent process.

Authentication providers can be found in the [Microsoft Entra admin center](#). Sign in as at least an **Authentication Policy Administrator**. Browse to **Protection > Multifactor authentication > Providers**. Click the listed providers to see details and configurations associated with that provider.

Before removing an authentication provider, take note of any customized settings configured in your provider. Decide what settings need to be migrated to general MFA settings from your provider and complete the migration of those settings.

Microsoft Entra multifactor authentication Servers linked to providers need to be reactivated using credentials generated under **Server settings**. Before reactivating, the following files must be deleted from the `\Program Files\Multi-Factor Authentication Server\Data\` directory on Microsoft Entra multifactor authentication Servers in your environment:

- caCert
- cert
- groupCACert
- groupKey
- groupName
- licenseKey
- pkey

Home > Users > Christa Geller > Multifactor authentication

Multifactor authentication | Providers

Overview Diagnose and solve problems

Adding new providers is disabled as of September 1, 2018. See the licensing documentation for details.

Don't see your provider listed? Please check the following:

- Providers are associated with specific subscription, each of which is associated with a tenant. Switch to the tenant associated with your provider's subscription to view that provider.
- Azure Private links are not supported with multifactor authentication providers, disable public link configuration to view providers.

Name	Usage Model	Subscription	Directory
Auth Provider	Per Authentication	8ae807b2-xxxx-xxxx-xxxx-26d277315d...	ee2653bf-xxxx-xxxx-c4c44d37d9...

After you confirm that all settings are migrated, browse to **Providers** and select the ellipses ... and select **Delete**.

⚠ Warning

Deleting an authentication provider deletes any reporting information associated with that provider. You may want to save activity reports before deleting your provider.

ⓘ Note

Users with older versions of the Microsoft Authenticator app and Microsoft Entra multifactor authentication Server may need to re-register their app.

Next steps

[Configure multifactor authentication settings](#)

[Common Conditional Access policy: Require MFA for all users](#)

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Enforce Microsoft Entra multifactor authentication with legacy applications using app passwords

Article • 03/04/2025

Some older, non-browser apps like Office 2010 or earlier and Apple Mail before iOS 11 don't understand pauses or breaks in the authentication process. A Microsoft Entra multifactor authentication (Microsoft Entra multifactor authentication) user who attempts to sign in to one of these older, non-browser apps, can't successfully authenticate. To use these applications in a secure way with Microsoft Entra multifactor authentication enforced for user accounts, you can use app passwords. These app passwords replaced your traditional password to allow an app to bypass multifactor authentication and work correctly.

Modern authentication is supported for the Microsoft Office 2013 clients and later. Office 2013 clients, including Outlook, support modern authentication protocols and can work with two-step verification. After Microsoft Entra multifactor authentication is enforced, app passwords aren't required for the client.

This article shows you how to use app passwords for legacy applications that don't support multifactor authentication prompts.

ⓘ Note

App passwords don't work for accounts that are required to use modern authentication.

Overview and considerations

When a user account is enforced for Microsoft Entra multifactor authentication, the regular sign-in prompt is interrupted by a request for additional verification. Some older applications don't understand this break in the sign-in process, so authentication fails. To maintain user account security and leave Microsoft Entra multifactor authentication enforced, app passwords can be used instead of the user's regular username and password. When an app password is used during sign-in, there's no additional verification prompt, so authentication is successful.

App passwords are automatically generated, not specified by the user. This automatically generated password makes it harder for an attacker to guess, so is more secure. Users don't have to keep track of the passwords or enter them every time as app passwords are only entered once per application.

When you use app passwords, the following considerations apply:

- There's a limit of 40 app passwords per user.
- Applications that cache passwords and use them in on-premises scenarios can fail because the app password isn't known outside the work or school account. An example of this scenario is Exchange emails that are on-premises, but the archived mail is in the cloud. In this scenario, the same password doesn't work.
- After Microsoft Entra multifactor authentication is enforced on a user's account, app passwords can be used with most non-browser clients like Outlook and Microsoft Skype for Business. However, administrative actions can't be performed by using app passwords through non-browser applications, such as Windows PowerShell. The actions can't be performed even when the user has an administrative account.
 - To run PowerShell scripts, create a service account with a strong password and don't enforce the account for two-step verification.
- If you suspect that a user account is compromised and revoke / reset the account password, app passwords should also be updated. App passwords aren't automatically revoked when a user account password is revoked / reset. The user should delete existing app passwords and create new ones.
 - For more information, see [Create and delete app passwords from the Additional security verification page](#).

Warning

App passwords don't work in hybrid environments where clients communicate with both on-premises and cloud auto-discover endpoints. Domain passwords are required to authenticate on-premises. App passwords are required to authenticate with the cloud.

App password names

App password names should reflect the device on which they're used. If you have a laptop that has non-browser applications like Outlook, Word, and Excel, create one app password named **Laptop** for these apps. Create another app password named **Desktop** for the same applications that run on your desktop computer.

It's recommended to create one app password per device, rather than one app password per application.

Federated or single sign-on app passwords

Microsoft Entra ID supports federation, or single sign-on (SSO), with on-premises Active Directory Domain Services (AD DS). If your organization is federated with Microsoft Entra ID and you're using Microsoft Entra multifactor authentication, the following app password considerations apply:

 **Note**

The following points apply only to federated (SSO) customers.

- App passwords are verified by Microsoft Entra ID, and therefore, bypass federation. Federation is actively used only when setting up app passwords.
- The Identity Provider (IdP) is not contacted for federated (SSO) users, unlike the passive flow. The app passwords are stored in the work or school account. If a user leaves the company, the user's information flows to the work or school account by using DirSync in real time. The disable / deletion of the account can take up to three hours to synchronize, which can delay the disable / deletion of the app password in Microsoft Entra ID.
- On-premises client Access Control settings aren't honored by the app passwords feature.
- No on-premises authentication logging or auditing capability is available with the app passwords feature.

Some advanced architectures require a combination of credentials for multifactor authentication with clients. These credentials can include a work or school account username and password, and app passwords. The requirements depend on how the authentication is performed. For clients that authenticate against an on-premises infrastructure, a work or school account username and password are required. For clients that authenticate against Microsoft Entra ID, an app password is required.

For example, suppose you have the following architecture:

- Your on-premises instance of Active Directory is federated with Microsoft Entra ID.
- You use Exchange online.
- You use Skype for Business on-premises.
- You use Microsoft Entra multifactor authentication.

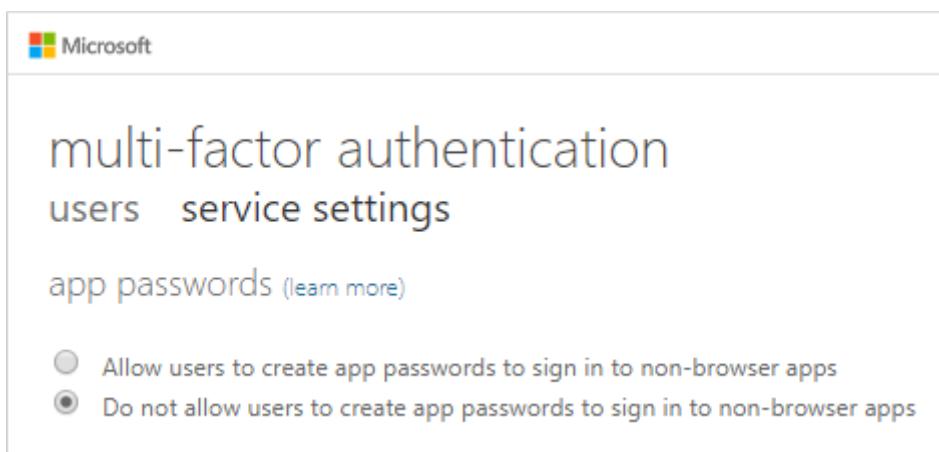
In this scenario, you use the following credentials:

- To sign in to Skype for Business, use your work or school account username and password.
- To access the address book from an Outlook client that connects to Exchange online, use an app password.

Allow users to create app passwords

By default, users can't create app passwords. The app passwords feature must be enabled before users can use them. To give users the ability to create app passwords, **admin needs to complete the following steps:**

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Conditional Access > Named locations**.
3. Select "Configure MFA trusted IPs" in the bar across the top of the *Conditional Access | Named Locations* window.
4. On the **Multifactor authentication** page, select the **Allow users to create app passwords to sign in to non-browser apps** option.



ⓘ Note

When you disable the ability for users to create app passwords, existing app passwords continue to work. However, users can't manage or delete those existing app passwords once you disable this ability.

When you disable the ability to create app passwords, it's also recommended to [create a Conditional Access policy to disable the use of legacy authentication](#).

This approach prevents existing app passwords from working, and forces the use of modern authentication methods.

Create an app password

When users complete their initial registration for Microsoft Entra multifactor authentication, there's an option to create app passwords at the end of the registration process.

Users can also create app passwords after registration. For more information and detailed steps for your users, see the following resource:

- [Create app passwords from the Security info page ↗](#)

Next steps

- For more information on how to allow users to quickly register for Microsoft Entra multifactor authentication, see [Combined security information registration overview](#).
- For more information about enabled and enforced user states for Microsoft Entra multifactor authentication, see [Enable per-user Microsoft Entra multifactor authentication to secure sign-in events](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Configure Microsoft Entra multifactor authentication as authentication provider using AD FS

Article • 03/13/2024 •

Applies Windows Server 2025, Windows Server 2022, Windows Server 2019,
to: Windows Server 2016

The information in this article applies to Windows 2016 and later.

If your organization is federated with Microsoft Entra ID, you can use Microsoft Entra multifactor authentication to secure Active Directory Federation Services (AD FS) resources, both on-premises and in the cloud. Microsoft Entra multifactor authentication enables you to eliminate passwords and provide a more secure way to authenticate. With AD FS, you can configure Microsoft Entra multifactor authentication for primary authentication or use it as an extra authentication provider.

Unlike with AD FS in Windows Server 2012 R2, the AD FS 2016 Microsoft Entra multifactor authentication adapter integrates directly with Microsoft Entra ID and doesn't require an on-premises Azure Multifactor Authentication Server. The Microsoft Entra multifactor authentication adapter is built into Windows Server 2016. No other installation is required.

Register users for Microsoft Entra multifactor authentication by using AD FS

AD FS doesn't support inline "proofup" registration of Microsoft Entra multifactor authentication security verification information, such as on a phone number or mobile app. Without support for inline proof, users must get proofed up by visiting <https://account.activedirectory.windowsazure.com/Proofup.aspx> before they use Microsoft Entra multifactor authentication to authenticate to AD FS applications. When a user that hasn't yet proofed up in Microsoft Entra ID tries to authenticate with Microsoft Entra multifactor authentication at AD FS, they get an AD FS error. As an AD FS administrator, you can customize this error experience to guide the user to the proofup page instead. You can create this message by using onload.js customization to detect the error message string within the AD FS page. Then you can show a new message to direct the user to <https://aka.ms/mfasetup> so that they can reattempt authentication. For more information, see [Customize the AD FS web page to guide users to register MFA verification methods](#).

Note

Prior to this update, users had to authenticate by using Microsoft Entra multifactor authentication for registration by visiting

<https://account.activedirectory.windowsazure.com/Proofup.aspx>. With this update, an AD FS user who hasn't yet registered Microsoft Entra multifactor authentication verification information can access the Azure proofup page by using the shortcut <https://aka.ms/mfasetup> with only primary authentication, such as Windows Integrated Authentication or username and password at the AD FS web pages. If the user has no verification methods configured, Microsoft Entra ID performs inline registration. The user sees the message, "Your admin has required that you set up this account for additional security verification." Then the user selects **Set it up now**. Users who already have at least one verification method configured will still be prompted to provide multifactor authentication (MFA) when visiting the proofup page.

Recommended deployment topologies

This section covers using Microsoft Entra multifactor authentication as the primary authentication method with AD FS and Microsoft Entra multifactor authentication for Office 365.

Microsoft Entra multifactor authentication as primary authentication

There are a couple of great reasons to use Microsoft Entra multifactor authentication as Primary Authentication with AD FS:

- It avoids passwords for sign-in to Microsoft Entra ID, Office 365, and other AD FS apps.
- It protects password based sign-in by requiring another factor, such as verification code prior to the password.

You also might want to use Microsoft Entra multifactor authentication as the primary authentication method and Microsoft Entra Conditional Access, including true MFA by prompting for extra factors. To use Microsoft Entra multifactor authentication on premises, you can configure the Microsoft Entra domain setting by setting `SupportsMfa` to `$true`. In this configuration, Microsoft Entra ID can prompt AD FS to perform extra authentication or "true MFA" for conditional access scenarios that require it.

Any AD FS user who isn't registered (hasn't yet configured MFA verification information), should be prompted to configure verification information. To prompt unregistered users, you can use a customized AD FS error page to direct users to <https://aka.ms/mfasetup> and configure verification information. After configuration, the user can reattempt their AD FS sign-in.

Microsoft Entra multifactor authentication as primary authentication is considered a single factor. After initial configuration users need to provide another factor to manage or update their verification information in Microsoft Entra ID, or to access other resources that require MFA.

ⓘ Note

With AD FS 2019, you're required to make a modification to the anchor claim type for the Active Directory Claims Provider trust and modify this from the `windowsaccountname` to User Principal Name (UPN). Run the following PowerShell cmdlet. This has no effect on the internal functioning of the AD FS farm. It's possible a few users might be re-prompted for credentials after this change is made. After logging in again, end users will see no difference.

PowerShell

```
Set-AdfsClaimsProviderTrust -AnchorClaimType  
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn" -TargetName  
"Active Directory"
```

Microsoft Entra multifactor authentication as extra authentication to Office 365

Microsoft Entra multifactor authentication adapter for AD FS enables your users to do MFA on AD FS. To secure your Microsoft Entra resource, you should require MFA through a [Conditional Access policy](#). You must also set the domain setting `SupportsMfa` to `$true` and [emit the multipleauthn claim](#) when a user performs two-step verification successfully.

As described previously, any AD FS user who isn't registered (hasn't yet configured MFA verification information) should be prompted to configure verification information. To prompt unregistered users, you can use a customized AD FS error page to direct users to <https://aka.ms/mfasetup> and configure verification information. After configuration, the user can reattempt their AD FS sign-in.

Prerequisites

The following prerequisites are required when you use Microsoft Entra multifactor authentication for authentication with AD FS:

- An Azure subscription with Microsoft Entra ID [↗](#).
- Microsoft Entra multifactor authentication.

 **Note**

Microsoft Entra ID and Microsoft Entra multifactor authentication are included in Microsoft Entra ID P1 or P2 and the Enterprise Mobility Suite (EMS). You don't need individual subscriptions if you have either of these applications installed.

- A Windows Server 2016 AD FS on-premises environment.
 - The server needs to be able to communicate with the following URLs over port 443.
 - <https://adnotifications.windowsazure.com>
 - <https://login.microsoftonline.com>
- Your on-premises environment must be [federated with Microsoft Entra ID](#).
- [Microsoft Azure Active Directory module for Windows PowerShell](#).
- Enterprise administrator credentials to configure the AD FS farm for Microsoft Entra multifactor authentication.
- You'll need either an account that has the [Application Administrator](#) role on your instance of Microsoft Entra ID to configure it by using PowerShell.

 **Note**

Azure AD and MSOnline PowerShell modules are deprecated as of March 30, 2024. To learn more, read the [deprecation update](#) [↗](#). After this date, support for these modules are limited to migration assistance to Microsoft Graph PowerShell SDK and security fixes. The deprecated modules will continue to function through March, 30 2025.

We recommend migrating to [Microsoft Graph PowerShell](#) to interact with Microsoft Entra ID (formerly Azure AD). For common migration questions, refer to the [Migration FAQ](#). Note: Versions 1.0.x of MSOnline may experience disruption after June 30, 2024.

Configure the AD FS Servers

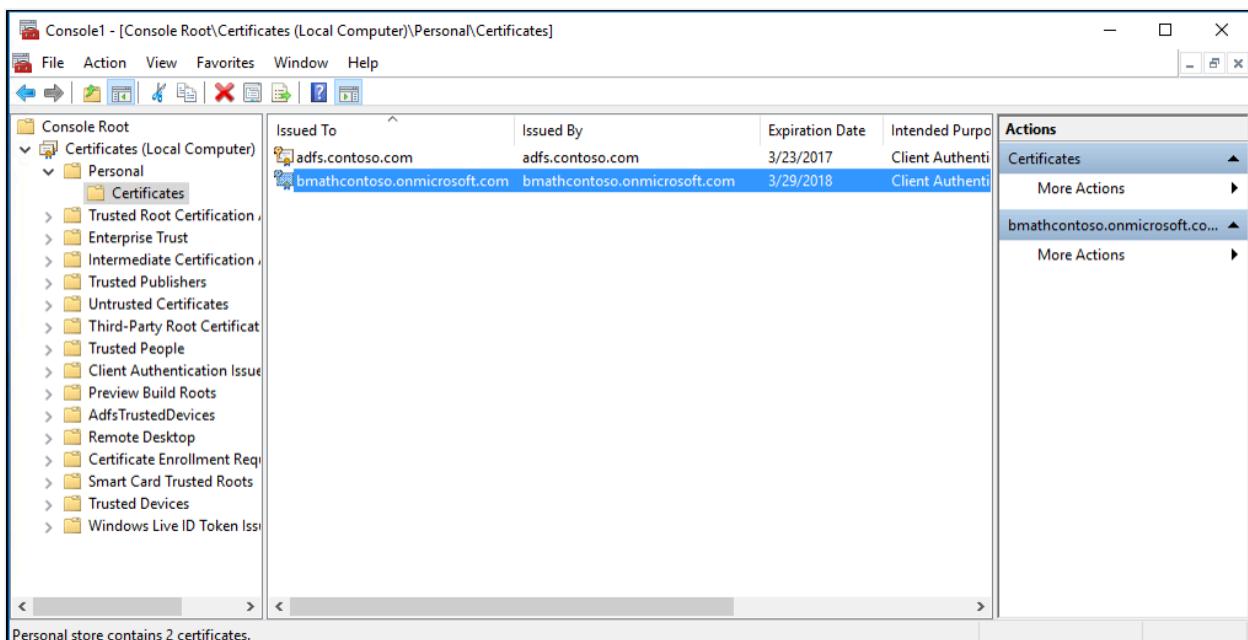
In order to complete configuration for Microsoft Entra multifactor authentication for AD FS, you need to configure each AD FS server by using the steps described here.

ⓘ Note

Ensure that these steps are performed on all AD FS servers in your farm. If you've multiple AD FS servers in your farm, you can perform the necessary configuration remotely by using Azure AD PowerShell.

Step 1: Generate a certificate for Microsoft Entra multifactor authentication on each AD FS server

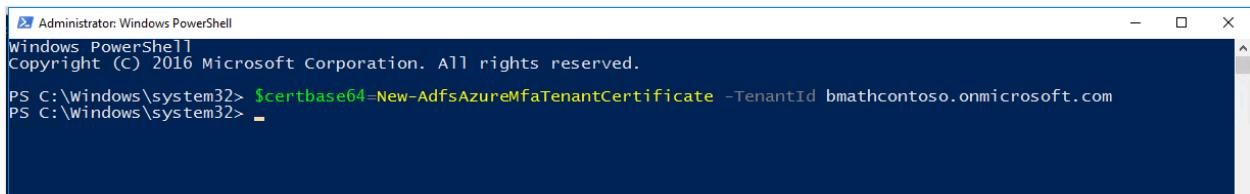
The first thing you need to do is to use the `New-AdfsAzureMfaTenantCertificate` PowerShell command to generate a certificate for Microsoft Entra multifactor authentication to use. After you generate the certificate, find it in the local machines certificate store. The certificate is marked with a subject name containing the TenantID for your Microsoft Entra directory.



The TenantID is the name of your directory in Microsoft Entra ID. Use the following PowerShell cmdlet to generate the new certificate:

PowerShell

```
$certbase64 = New-AdfsAzureMfaTenantCertificate -TenantID <tenantID>
```



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> $certbase64=New-AdfsAzureMfaTenantCertificate -TenantId bmathcontoso.onmicrosoft.com
PS C:\Windows\system32> -
```

Step 2: Add the new credentials to the Azure multifactor authentication Client Service Principal

In order to enable the AD FS servers to communicate with the Azure multifactor authentication Client, you need to add the credentials to the Service Principal for the Azure multifactor authentication Client. The certificates generated by using the New-AdfsAzureMfaTenantCertificate cmdlet serve as these credentials. Open PowerShell, and perform the following steps to add the new credentials to the Azure multifactor authentication Client Service Principal.

Step 3: Set the certificate as the new credential against the Azure multifactor authentication Client

ⓘ Note

In order to complete this step you need to connect to your instance of Microsoft Entra ID with Microsoft Graph PowerShell by using `Connect-MgGraph`. These steps assume you've already connected via PowerShell.

PowerShell

```
Connect-MgGraph -Scopes 'Application.ReadWrite.All'
$servicePrincipalId = (Get-MgServicePrincipal -Filter "appid eq '981f26a1-7f43-403b-a875-f8b09b8cd720'").Id
$keyCredentials = (Get-MgServicePrincipal -Filter "appid eq '981f26a1-7f43-403b-a875-f8b09b8cd720'").KeyCredentials
$certX509 = [System.Security.Cryptography.X509Certificates.X509Certificate2]([System.Convert]::FromBase64String($certBase64))
$newKey = @(@{
    CustomKeyIdentifier = $null
    DisplayName = $certX509.Subject
    EndDateTime = $null
    Key = $certX509.GetRawCertData()
    KeyId = [guid]::NewGuid()
    StartDateTime = $null
    Type = "AsymmetricX509Cert"
    Usage = "Verify"
    AdditionalProperties = $null
})
$keyCredentials += $newKey
```

```
Update-MgServicePrincipal -ServicePrincipalId $servicePrincipalId -  
KeyCredentials $keyCredentials
```

ⓘ Important

This command needs to be run on all of the AD FS servers in your farm. Microsoft Entra multifactor authentication will fail on servers that haven't had the certificate set as the new credential against the Azure multifactor authentication Client.

ⓘ Note

981f26a1-7f43-403b-a875-f8b09b8cd720 is the GUID for Azure multifactor authentication Client.

Configure the AD FS Farm

After you've completed the steps in the previous section for each AD FS server, set the Azure tenant information by using the [Set-AdfsAzureMfaTenant](#) cmdlet. This cmdlet needs to be executed only once for an AD FS farm.

Open PowerShell, and enter your own *tenantId* with the [Set-AdfsAzureMfaTenant](#) cmdlet. For customers that use Microsoft Azure Government cloud, add the `-Environment USGov` parameter:

ⓘ Note

You need to restart the AD FS service on each server in your farm before these changes take effect. For minimal impact, take each AD FS server out of the NLB rotation one at a time and wait for all connections to drain.

PowerShell

```
Set-AdfsAzureMfaTenant -TenantId <tenant ID> -ClientId 981f26a1-7f43-403b-  
a875-f8b09b8cd720
```

```
Windows PowerShell  
Copyright (C) 2016 Microsoft Corporation. All rights reserved.  
PS C:\Windows\system32> Set-AdfsAzureMfaTenant -TenantId bm.onmicrosoft.com -ClientId 22223333-cccc-4444-dddd-5555eeee6666  
WARNING: PS0177: The authentication provider configuration data was successfully updated. Before your changes take  
effect, you must restart the AD FS Windows Service on each server in the farm.  
PS C:\Windows\system32> -
```

Windows Server without the latest service pack doesn't support the `-Environment` parameter for the `Set-AdfsAzureMfaTenant` cmdlet. If you use Azure Government cloud and the previous steps failed to configure your Azure tenant due to the missing `-Environment` parameter, complete the following steps to manually create the registry entries. Skip these steps if the previous cmdlet correctly registered your tenant information or if you aren't in the Azure Government cloud:

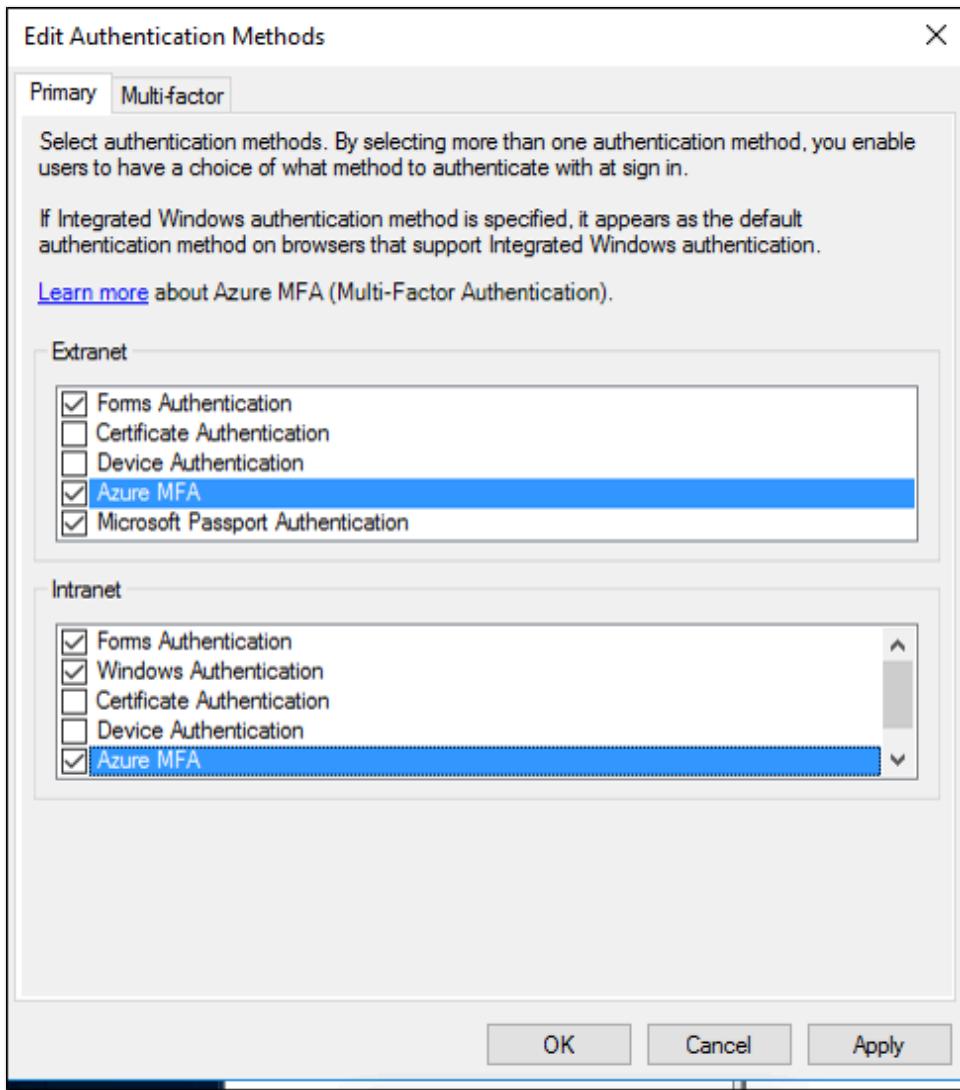
1. Open **Registry Editor** on the AD FS server.
2. Navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ADFS**. Create the following registry key values:

 Expand table

Registry key	Value
SasUrl	<code>https://adnotifications.windowsazure.us/StrongAuthenticationService.svc/Connector</code>
StsUrl	<code>https://login.microsoftonline.us</code>
ResourceUri	<code>https://adnotifications.windowsazure.us/StrongAuthenticationService.svc/Connector</code>

3. Restart the AD FS service on each server in the farm before these changes take effect. To reduce the effect on your systems, take each AD FS server out of the NLB rotation one at a time and wait for all connections to drain.

After this step, you'll see that Microsoft Entra multifactor authentication is available as a primary authentication method for intranet and extranet use.



If you want to use Microsoft Entra multifactor authentication as a secondary authentication method, on the **Edit Authentication Methods** box, select the **multifactor** tab (the Additional tab in AD FS 2019) and ensure that it's enabled. Otherwise you might receive error messages, such as, "No valid strong authentication method found. Contact your administrator to configure and enable an appropriate strong authentication provider."

Renew and Manage AD FS Microsoft Entra multifactor authentication Certificates

The following guidance is designed to help you manage the Microsoft Entra multifactor authentication certificates on your AD FS servers.

By default, when you configure AD FS with Microsoft Entra multifactor authentication, the certificates generated via the `New-AdfsAzureMfaTenantCertificate` PowerShell cmdlet are valid for two years. To determine how close to expiration your certificates are, and to renew and install new certificates, use the following procedure.

1. Assess AD FS Microsoft Entra multifactor authentication certificate expiration date.

On each AD FS server, in the local computer My store, there's a self signed certificate with "Microsoft AD FS Microsoft Entra multifactor authentication" in the Issuer and Subject area. This certificate is the Microsoft Entra multifactor authentication certificate. Check the validity period of this certificate on each AD FS server to determine the expiration date.

2. Create a new AD FS Microsoft Entra multifactor authentication Certificate on each AD FS server.

If the validity period of your certificates is nearing its end, start the renewal process by generating a new Microsoft Entra multifactor authentication certificate on each AD FS server. In PowerShell generate a new certificate on each AD FS server by using the following cmdlet:

Caution

If your certificate has already expired, don't add the `-Renew $true` parameter to the following command. In this scenario, the existing expired certificate is replaced with a new one instead of being left in place and an additional certificate created.

PowerShell

```
$newcert = New-AdfsAzureMfaTenantCertificate -TenantId <tenant id such as contoso.onmicrosoft.com> -Renew $true
```

If the certificate hasn't already expired, the command generates a new certificate that is valid from two days after the current day to two years plus two days in the future. AD FS and Microsoft Entra multifactor authentication operations aren't affected when running the cmdlet or renewing the certificate. The two-day delay is intentional and provides time to follow the next steps to configure the new certificate in the tenant before AD FS starts by using it for Microsoft Entra multifactor authentication.

3. Configure each new AD FS Microsoft Entra multifactor authentication certificate in the Microsoft Entra tenant.

Note

In order to complete this step you need to connect to your instance of Microsoft Entra ID with Microsoft Graph PowerShell by using `Connect-MgGraph`. These steps assume you've already connected via PowerShell.

PowerShell

```
Connect-MgGraph -Scopes 'Application.ReadWrite.All'
$servicePrincipalId = (Get-MgServicePrincipal -Filter "appid eq
'981f26a1-7f43-403b-a875-f8b09b8cd720").Id
$keyCredentials = (Get-MgServicePrincipal -Filter "appid eq '981f26a1-
7f43-403b-a875-f8b09b8cd720").KeyCredentials
$certX509 =
[System.Security.Cryptography.X509Certificates.X509Certificate2]
([System.Convert]::FromBase64String($newcert))
$newKey = @(@{
    CustomKeyIdentifier = $null
    DisplayName = $certX509.Subject
    EndDateTime = $null
    Key = $certX509.GetRawCertData()
    KeyId = [guid]::NewGuid()
    StartDateTime = $null
    Type = "AsymmetricX509Cert"
    Usage = "Verify"
    AdditionalProperties = $null
})
$keyCredentials += $newKey
Update-MgServicePrincipal -ServicePrincipalId $servicePrincipalId -
KeyCredentials $keyCredentials
```

If your previous certificate is expired, restart the AD FS service to pick up the new certificate. You don't need to restart the AD FS service if you renewed a certificate before it expired.

4. Verify that the new certificate(s) is used for Microsoft Entra multifactor authentication.

After the new certificate(s) become valid, AD FS will pick them up and use each respective certificate for Microsoft Entra multifactor authentication within a few hours to one day.

After AD FS uses the new certificates, on each server you'll see an event logged in the AD FS Admin event log with the following information:

Output

Log Name:	AD FS/Admin
Source:	AD FS
Date:	2/27/2018 7:33:31 PM
Event ID:	547
Task Category:	None
Level:	Information

```
Keywords: AD FS
User: DOMAIN\adfssvc
Computer: ADFS.domain.contoso.com
Description:
The tenant certificate for Azure MFA has been renewed.

TenantId: contoso.onmicrosoft.com.
Old thumbprint: 7CC103D60967318A11D8C51C289EF85214D9FC63.
Old expiration date: 9/15/2019 9:43:17 PM.
New thumbprint: 8110D7415744C9D4D5A4A6309499F7B48B5F3CCF.
New expiration date: 2/27/2020 2:16:07 AM.
```

Customize the AD FS web page to guide users to register MFA verification methods

Use the following examples to customize your AD FS web pages for users who haven't yet proofed up (configured MFA verification information).

Find the error

First, AD FS returns a couple of different error messages when the user lacks verification information. If you're using Microsoft Entra multifactor authentication as primary authentication, the unproofed user sees an AD FS error page containing the following messages:

HTML

```
<div id="errorArea">
    <div id="openingMessage" class="groupMargin bigText">
        An error occurred
    </div>
    <div id="errorMessage" class="groupMargin">
        Authentication attempt failed. Select a different sign in option
        or close the web browser and sign in again. Contact your administrator for
        more information.
    </div>
```

When Microsoft Entra ID as extra authentication is being attempted, the unproofed user sees an AD FS error page containing the following messages:

HTML

```
<div id='mfaGreetingDescription' class='groupMargin'>For security reasons, we
require additional information to verify your account (mahesh@jenfield.net)
</div>
<div id="errorArea">
```

```
<div id="openingMessage" class="groupMargin bigText">
    An error occurred
</div>
<div id="errorMessage" class="groupMargin">
    The selected authentication method is not available for
    'username@contoso.com'. Choose another authentication method or
    contact your system administrator for details.
</div>
```

Catch the error and update the page text

To catch the error and show the user custom guidance, append the JavaScript to the end of the `onload.js` file that's part of the AD FS web theme. Doing so allows you to:

- Search for the identifying error string(s).
- Provide custom web content.

Note

For guidance in general on how to customize the `onload.js` file, see [Advanced Customization of AD FS Sign-in Pages](#).

The following steps show a simple example:

1. Open Windows PowerShell on your primary AD FS server, and create a new AD FS Web Theme by running the following command.

```
PowerShell
New-AdfsWebTheme -Name ProofUp -SourceName default
```

2. Create the folder, and export the default AD FS Web Theme.

```
PowerShell
New-Item -Path 'C:\Theme' -ItemType Directory; Export-AdfsWebTheme -Name default -DirectoryPath C:\Theme
```

3. Open the `C:\Theme\script\ onload.js` file in a text editor.

4. Append the following code to the end of the `onload.js` file:

```
JavaScript
```

```

//Custom Code
//Customize MFA exception
//Begin

var domain_hint = "<YOUR_DOMAIN_NAME_HERE>";
var mfaSecondFactorErr = "The selected authentication method is not
available for";
var mfaProofupMessage = "You will be automatically redirected in 5
seconds to set up your account for additional security verification.
After you've completed the setup, please return to the application you
are attempting to access.<br><br>If you are not redirected
automatically, please click <a href='{0}'>here</a>."
var authArea = document.getElementById("authArea");
if (authArea) {
    var errorMessage = document.getElementById("errorMessage");
    if (errorMessage) {
        if (errorMessage.innerHTML.indexOf(mfaSecondFactorErr) >= 0) {

            //Hide the error message
            var openingMessage =
document.getElementById("openingMessage");
            if (openingMessage) {
                openingMessage.style.display = 'none'
            }
            var errorDetailsLink =
document.getElementById("errorDetailsLink");
            if (errorDetailsLink) {
                errorDetailsLink.style.display = 'none'
            }

            //Provide a message and redirect to Azure AD MFA
Registration Url
            var mfaRegisterUrl =
"https://account.activedirectory.windowsazure.com/proofup.aspx?
proofup=1&whr=" + domain_hint;
            errorMessage.innerHTML = "<br>" +
mfaProofupMessage.replace("{0}", mfaRegisterUrl);
            window.setTimeout(function () { window.location.href =
mfaRegisterUrl; }, 5000);
        }
    }
}

//End Customize MFA Exception
//End Custom Code

```

Important

You need to change "<YOUR_DOMAIN_NAME_HERE>"; to use your domain name. For example: var domain_hint = "contoso.com"; .

5. Save the onload.js file.
6. Import the onload.js file into your custom theme by entering the following Windows PowerShell command:

```
PowerShell
```

```
Set-AdfsWebTheme -TargetName ProofUp -AdditionalFileResource  
@{Uri='/adfs/portal/script/onload.js';path="c:\theme\script\onload.js"}
```

7. Apply the custom AD FS Web Theme by entering the following Windows PowerShell command:

```
PowerShell
```

```
Set-AdfsWebConfig -ActiveThemeName "ProofUp"
```

Related links

- [Manage SSL/TLS protocols and cipher suites for AD FS](#)
-

Feedback

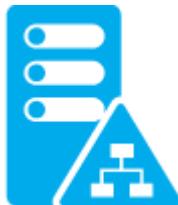
Was this page helpful?

 Yes

 No

Getting started with Microsoft Entra multifactor authentication and Active Directory Federation Services

Article • 03/04/2025



If your organization has federated your on-premises Active Directory with Microsoft Entra ID using AD FS, there are two options for using Microsoft Entra multifactor authentication.

- Secure cloud resources using Microsoft Entra multifactor authentication or Active Directory Federation Services
- Secure cloud and on-premises resources using Azure Multifactor Authentication Server

The following table summarizes the verification experience between securing resources with Microsoft Entra multifactor authentication and AD FS

[] Expand table

Verification Experience - Browser-based Apps	Verification Experience - Non-Browser-based Apps
Securing Microsoft Entra resources using Microsoft Entra multifactor authentication	<ul style="list-style-type: none">• The first verification step is performed on-premises using AD FS.• The second step is a phone-based method carried out using cloud authentication.
Securing Microsoft Entra resources using Active Directory Federation Services	<ul style="list-style-type: none">• The first verification step is performed on-premises using AD FS.• The second step is performed on-premises by honoring the claim.

Caveats with app passwords for federated users:

- App passwords are verified using cloud authentication, so they bypass federation. Federation is only actively used when setting up an app password.
- On-premises Client Access Control settings aren't honored by app passwords.

- You lose on-premises authentication-logging capability for app passwords.
- Account disable/deletion may take up to three hours for directory sync, delaying disable/deletion of app passwords in the cloud identity.

For information on setting up either Microsoft Entra multifactor authentication or the Azure Multifactor Authentication Server with AD FS, see the following articles:

- [Secure cloud resources using Microsoft Entra multifactor authentication and AD FS](#)
- [Secure cloud and on-premises resources using Azure Multifactor Authentication Server with Windows Server](#)
- [Secure cloud and on-premises resources using Azure Multifactor Authentication Server with AD FS 2.0](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Securing cloud resources with Microsoft Entra multifactor authentication and AD FS

Article • 03/04/2025

If your organization is federated with Microsoft Entra ID, use Microsoft Entra multifactor authentication or Active Directory Federation Services (AD FS) to secure resources that are accessed by Microsoft Entra ID. Use the following procedures to secure Microsoft Entra resources with either Microsoft Entra multifactor authentication or Active Directory Federation Services.

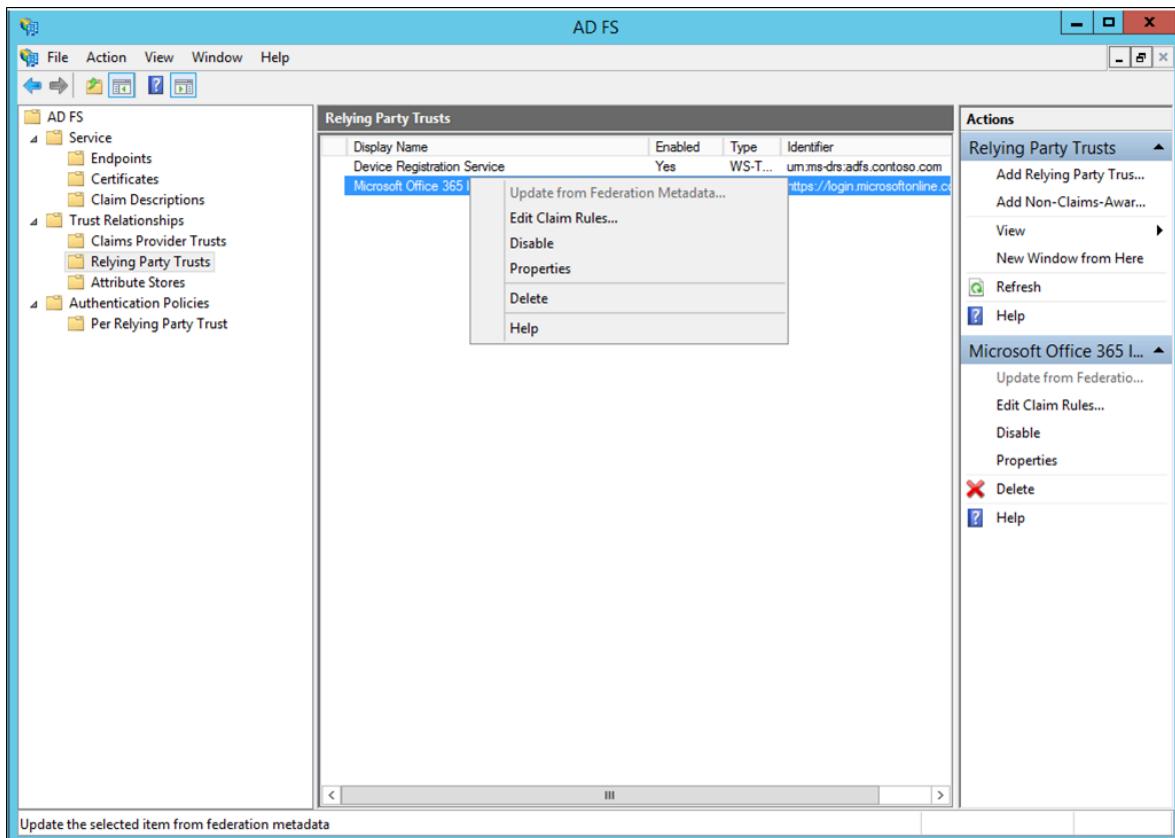
ⓘ Note

Set the domain setting `federatedIdpMfaBehavior` to `enforceMfaByFederatedIdp` (recommended) or `SupportsMFA` to `$True`. The `federatedIdpMfaBehavior` setting overrides `SupportsMFA` when both are set.

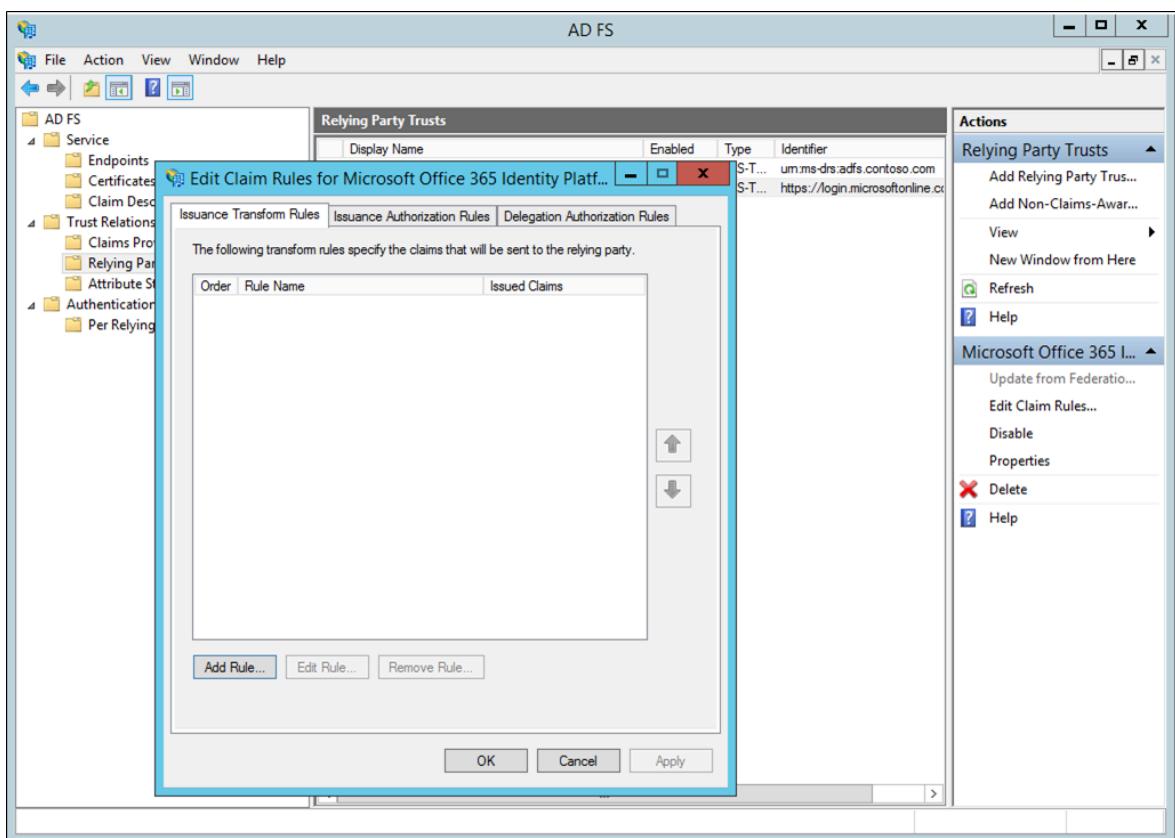
Secure Microsoft Entra resources using AD FS

To secure your cloud resource, set up a claims rule so that Active Directory Federation Services emits the `multipleauthn` claim when a user performs two-step verification successfully. This claim is passed on to Microsoft Entra ID. Follow this procedure to walk through the steps:

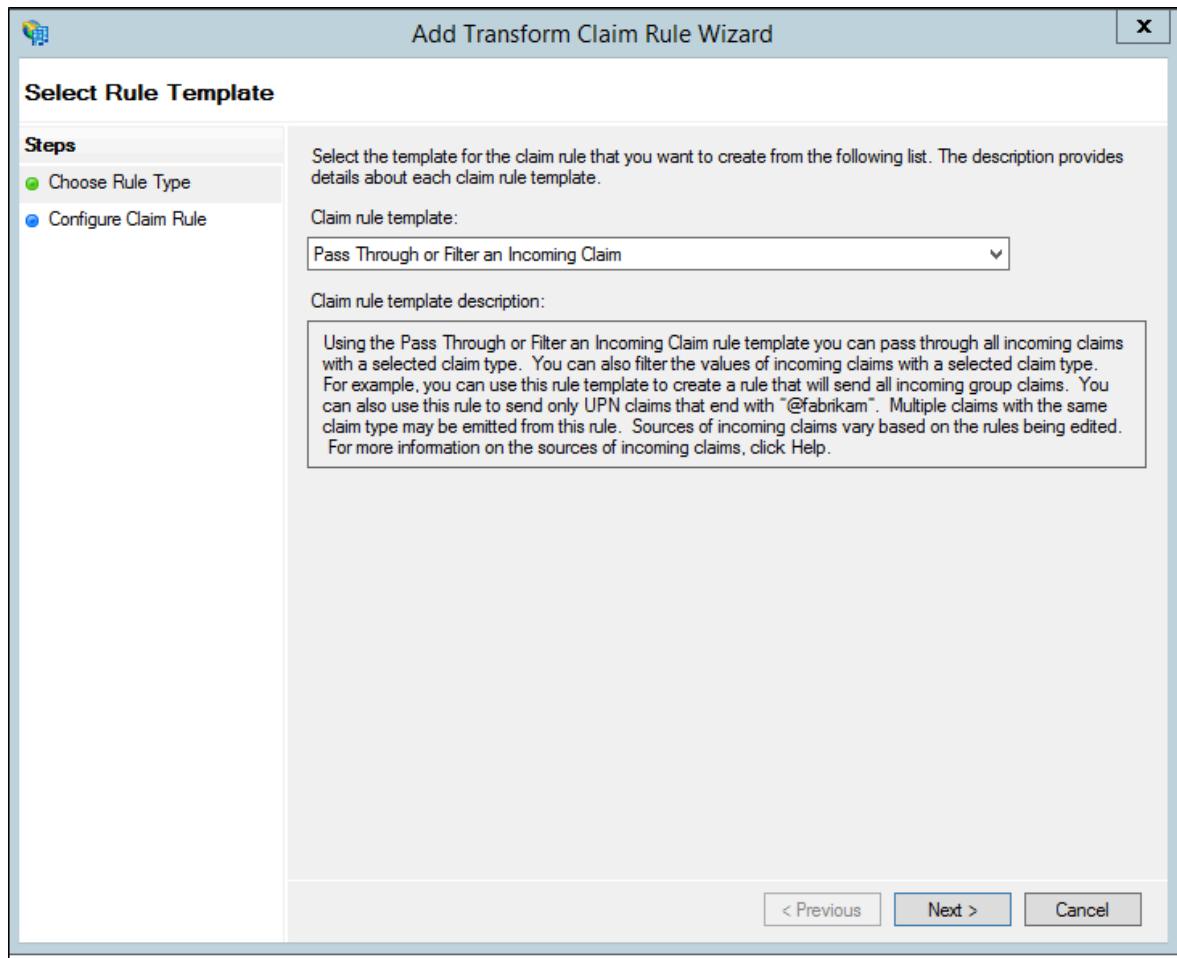
1. Open AD FS Management.
2. On the left, select **Relying Party Trusts**.
3. Right-select on **Microsoft Office 365 Identity Platform** and select **Edit Claim Rules**.



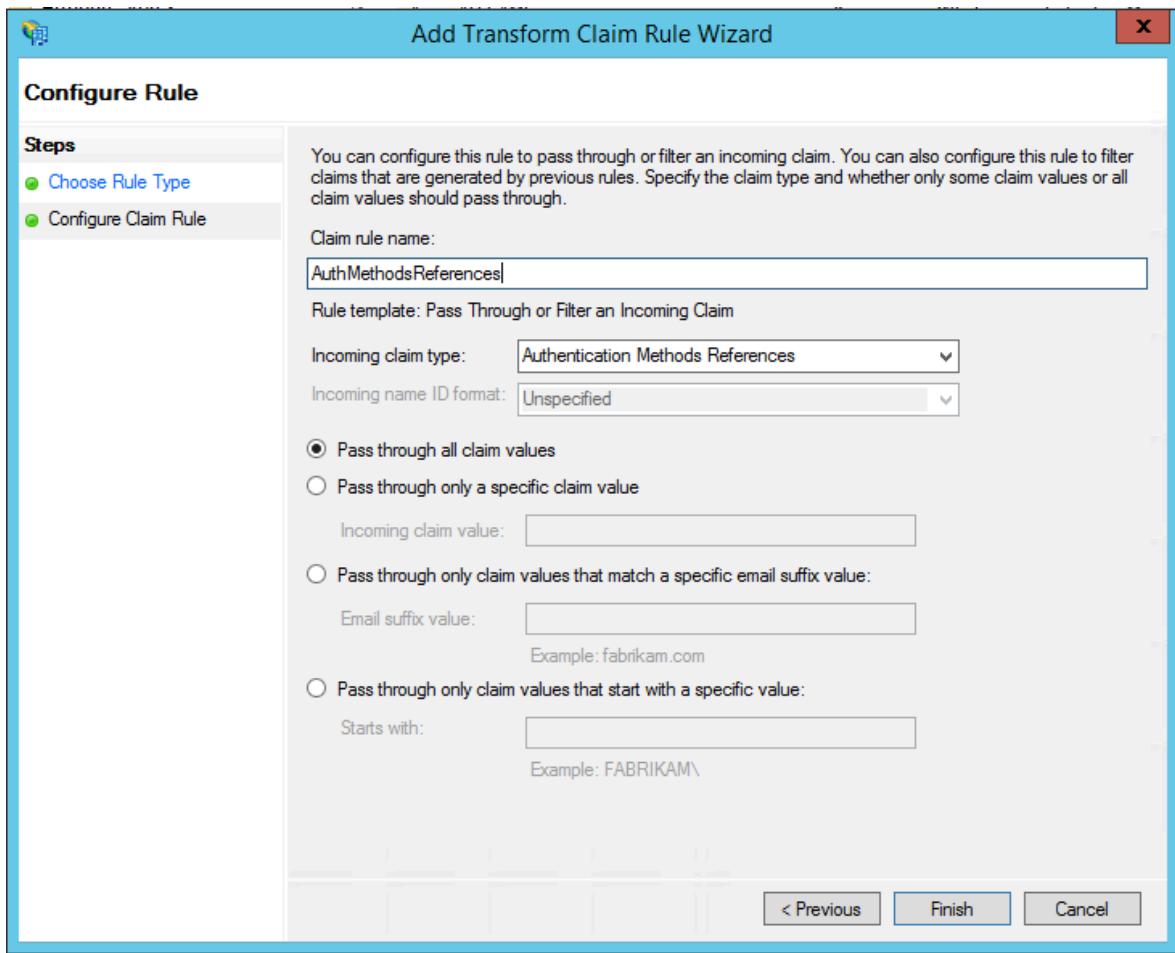
4. On Issuance Transform Rules, select Add Rule.



5. On the Add Transform Claim Rule Wizard, select Pass Through or Filter an Incoming Claim from the drop-down and select Next.



6. Give your rule a name.
7. Select **Authentication Methods References** as the Incoming claim type.
8. Select Pass through all claim values.



9. Select **Finish**. Close the AD FS Management console.

Trusted IPs for federated users

Trusted IPs allow administrators to bypass two-step verification for specific IP addresses, or for federated users who have requests originating from within their own intranet. The following sections describe how to configure the bypass using Trusted IPs. This is achieved by configuring AD FS to use a pass-through or filter an incoming claim template with the Inside Corporate Network claim type.

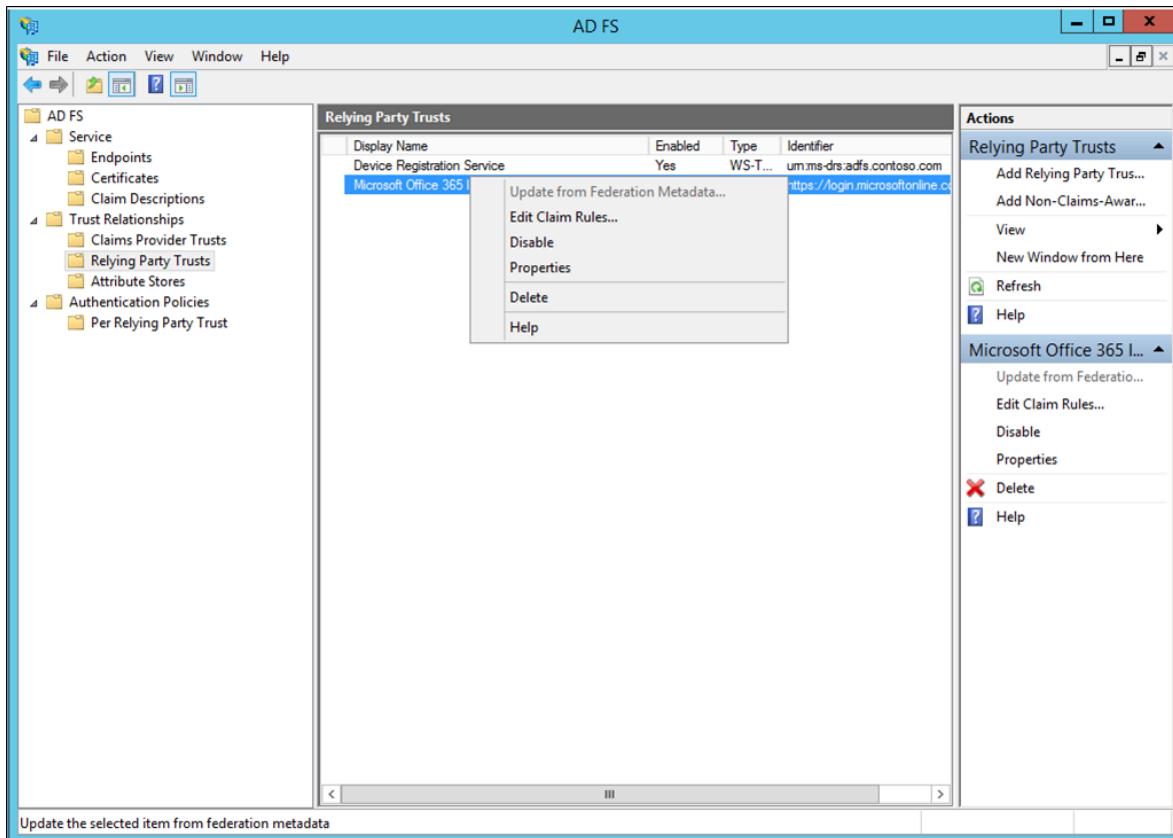
This example uses Microsoft 365 for our Relying Party Trusts.

Configure the AD FS claims rules

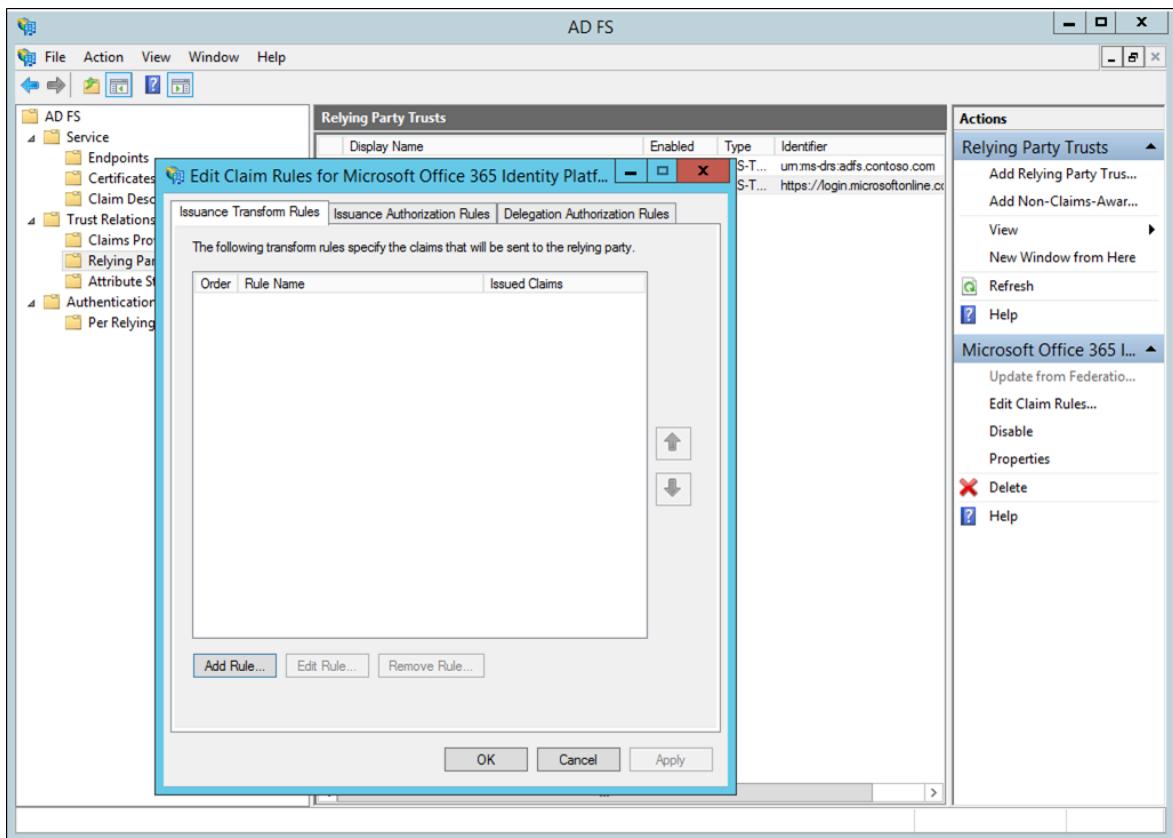
The first thing we need to do is to configure the AD FS claims. Create two claims rules, one for the Inside Corporate Network claim type and an additional one for keeping our users signed in.

1. Open AD FS Management.
2. On the left, select **Relying Party Trusts**.

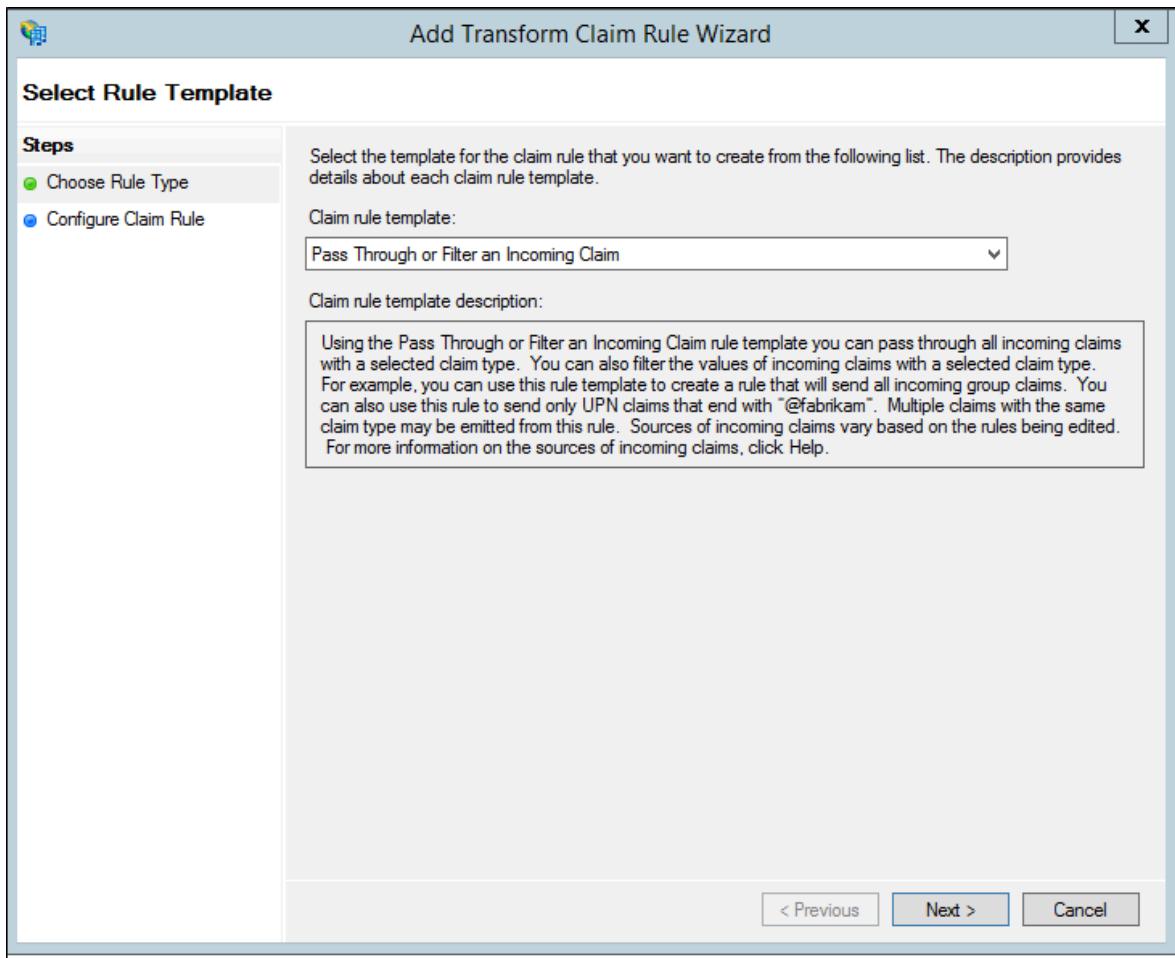
3. Right-select on Microsoft Office 365 Identity Platform and select Edit Claim Rules...



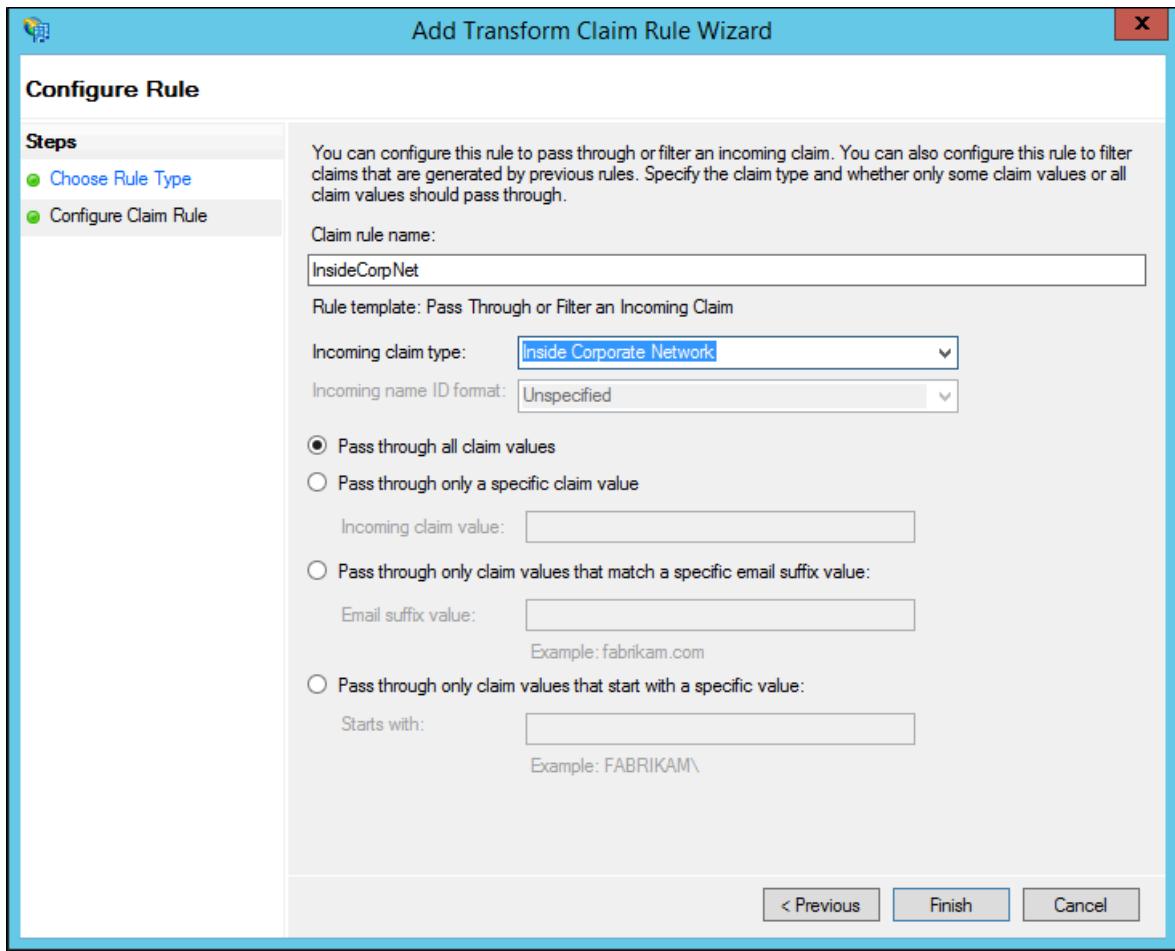
4. On Issuance Transform Rules, select Add Rule.



5. On the Add Transform Claim Rule Wizard, select Pass Through or Filter an Incoming Claim from the drop-down and select Next.



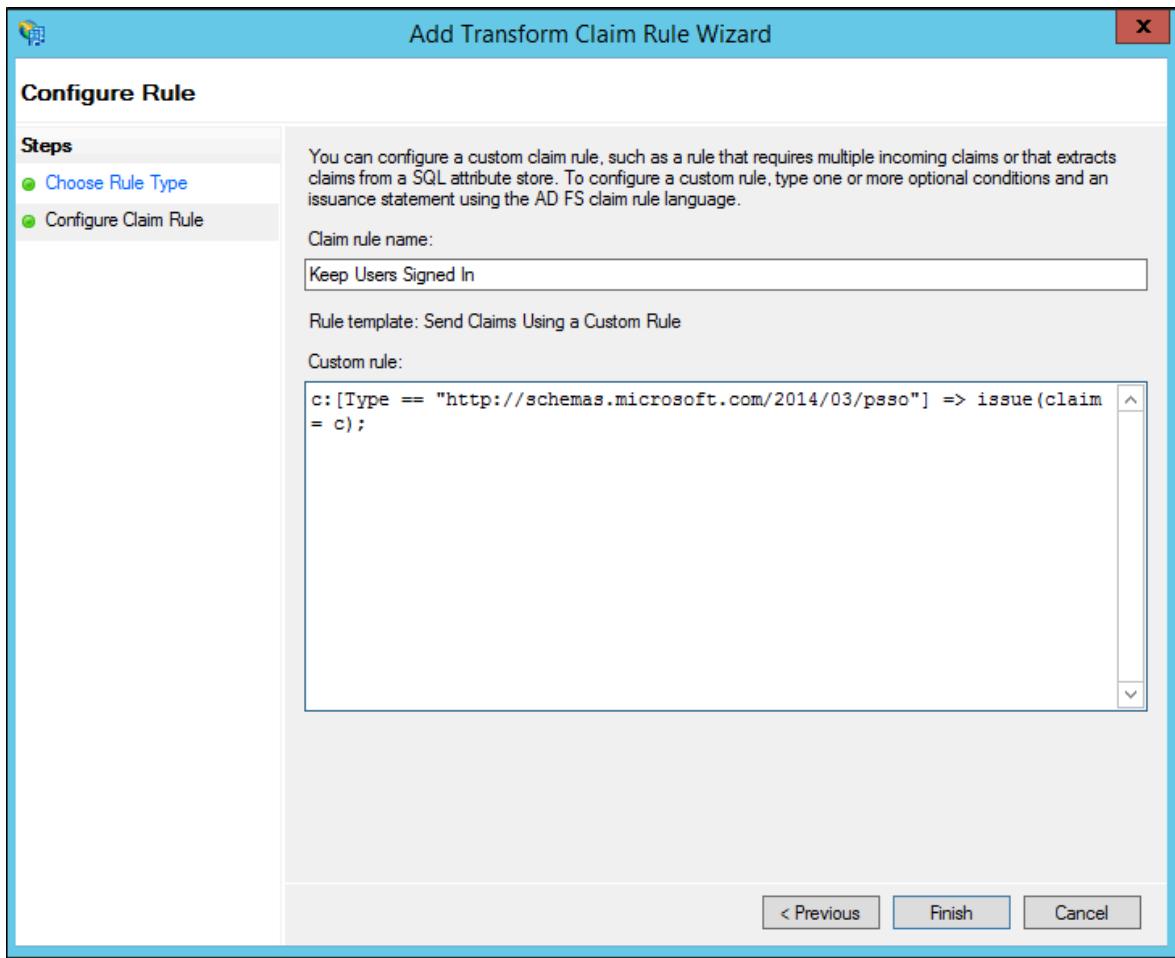
6. In the box next to Claim rule name, give your rule a name. For example:
InsideCorpNet.
7. From the drop-down, next to Incoming claim type, select **Inside Corporate Network**.



8. Select **Finish**.
9. On Issuance Transform Rules, select **Add Rule**.
10. On the Add Transform Claim Rule Wizard, select **Send Claims Using a Custom Rule** from the drop-down and select **Next**.
11. In the box under Claim rule name: enter *Keep Users Signed In*.
12. In the Custom rule box, enter:

```
ad-fs-claim-rule
```

```
c:[Type == "https://schemas.microsoft.com/2014/03/pso"]  
=> issue(claim = c);
```



13. Select Finish.

14. Select Apply.

15. Select Ok.

16. Close AD FS Management.

Configure Microsoft Entra multifactor authentication Trusted IPs with federated users

Now that the claims are in place, we can configure trusted IPs.

1. Sign in to the Microsoft Entra admin center [as at least an Authentication Policy Administrator](#).
2. Browse to **Conditional Access > Named locations**.
3. From the **Conditional Access - Named locations** blade, select **Configure MFA trusted IPs**

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation sidebar with various categories like Billing, Settings, Protection, and Conditional Access. The Conditional Access section is expanded, showing sub-options like Overview, Policies, Insights and reporting, Diagnose and solve problems, Manage, and Named locations. The 'Named locations' option is highlighted with a red box. The main content area is titled 'Conditional Access | Named locations' and contains a note about named locations being used for security reports and Conditional Access policies. It includes a search bar, filter buttons for location type and trusted status, and a table showing 0 named locations found. The table has columns for Name, Location type, Trusted, and Conditional Access policies.

4. On the Service Settings page, under **trusted IPs**, select **Skip multifactor-authentication for requests from federated users on my intranet**.

5. Select **save**.

That's it! At this point, federated Microsoft 365 users should only have to use MFA when a claim originates from outside the corporate intranet.

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Integrate your existing Network Policy Server (NPS) infrastructure with Microsoft Entra multifactor authentication

Article • 03/04/2025

The Network Policy Server (NPS) extension for Microsoft Entra multifactor authentication adds cloud-based MFA capabilities to your authentication infrastructure using your existing servers. With the NPS extension, you can add phone call, text message, or phone app verification to your existing authentication flow without having to install, configure, and maintain new servers.

The NPS extension acts as an adapter between RADIUS and cloud-based Microsoft Entra multifactor authentication to provide a second factor of authentication for federated or synced users.

How the NPS extension works

When you use the NPS extension for Microsoft Entra multifactor authentication, the authentication flow includes the following components:

1. **NAS/VPN Server** receives requests from VPN clients and converts them into RADIUS requests to NPS servers.
2. **NPS Server** connects to Active Directory Domain Services (AD DS) to perform the primary authentication for the RADIUS requests and, upon success, passes the request to any installed extensions.
3. **NPS Extension** triggers a request to Microsoft Entra multifactor authentication for the secondary authentication. Once the extension receives the response, and if the MFA challenge succeeds, it completes the authentication request by providing the NPS server with security tokens that include an MFA claim, issued by Azure STS.

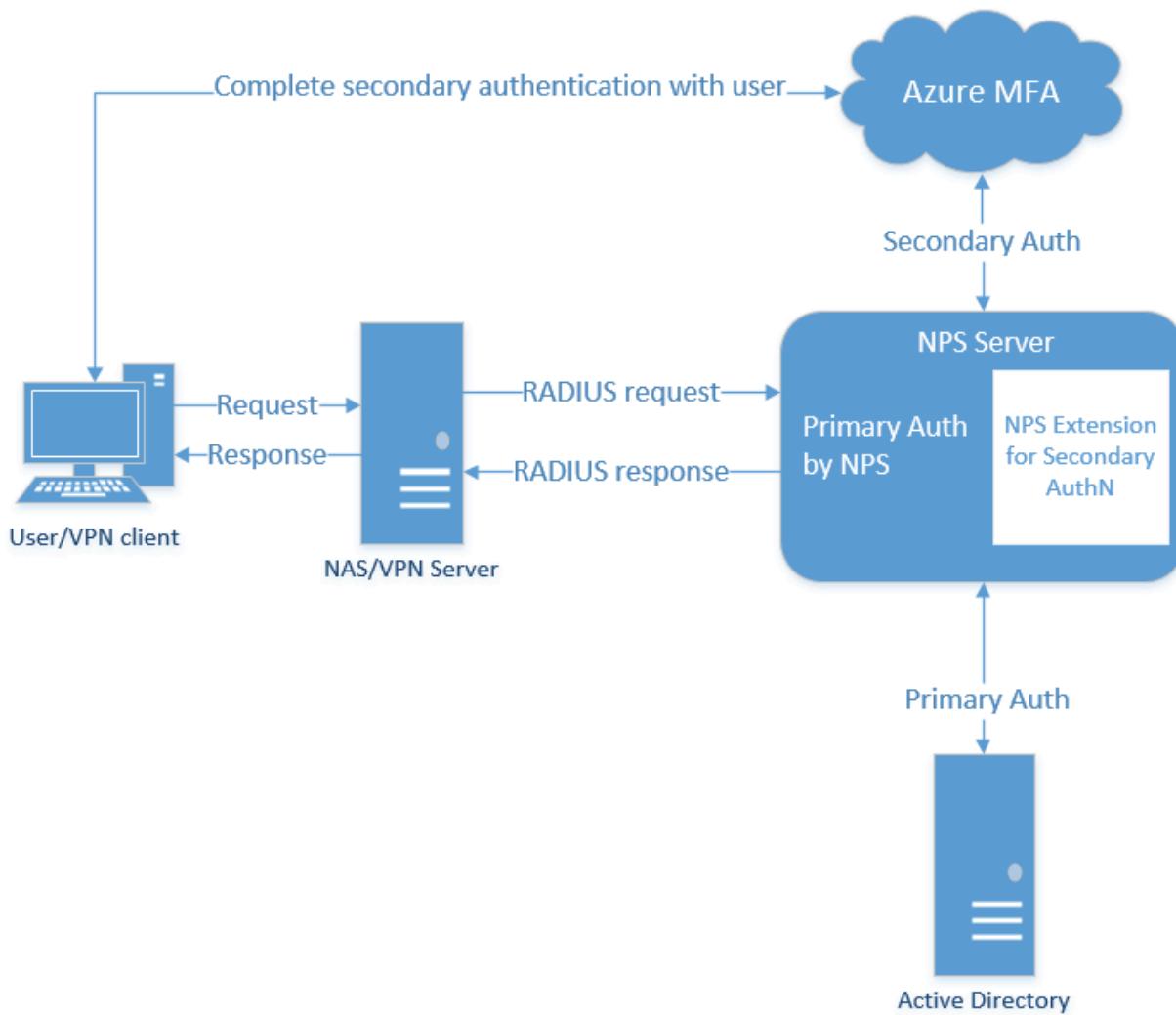
Note

Although NPS doesn't support [number matching](#), the latest NPS extension does support time-based one-time password (TOTP) methods, such as the TOTP available in Microsoft Authenticator. TOTP sign-in provides better security than the alternative **Approve/Deny** experience.

After May 8, 2023, when number matching is enabled for all users, anyone who performs a RADIUS connection with NPS extension version 1.2.2216.1 or later will be prompted to sign in with a TOTP method instead. Users must have a TOTP authentication method registered to see this behavior. Without a TOTP method registered, users continue to see **Approve/Deny**.

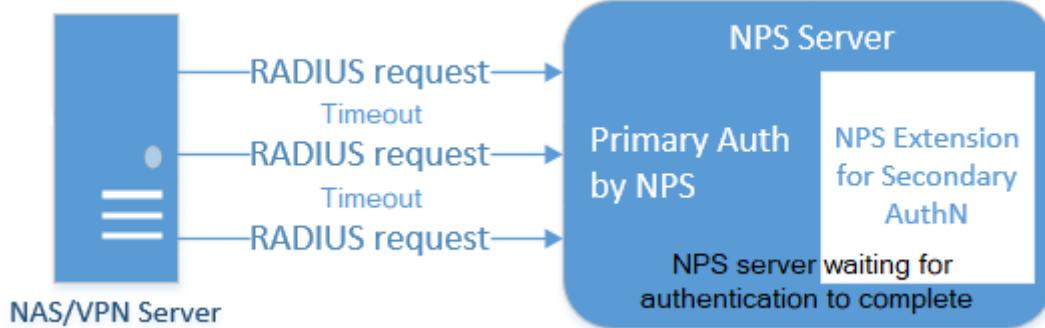
4. Microsoft Entra multifactor authentication communicates with Microsoft Entra ID to retrieve the user's details and performs the secondary authentication using a verification method configured to the user.

The following diagram illustrates this high-level authentication request flow:

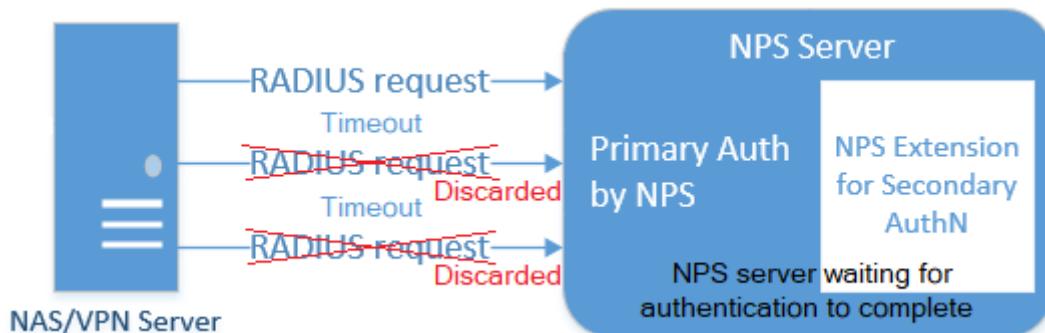


RADIUS protocol behavior and the NPS extension

As RADIUS is a UDP protocol, the sender assumes packet loss and awaits a response. After a period of time, the connection may time out. If so, the packet is resent as the sender assumes the packet didn't reach the destination. In the authentication scenario in this article, VPN servers send the request and wait for a response. If the connection times out, the VPN server sends the request again.



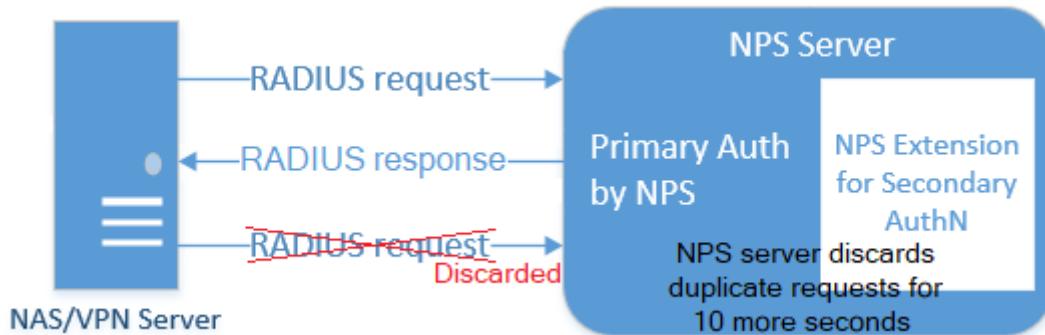
The NPS server may not respond to the VPN server's original request before the connection times out as the MFA request may still be being processed. The user may not have successfully responded to the MFA prompt, so the Microsoft Entra multifactor authentication NPS extension is waiting for that event to complete. In this situation, the NPS server identifies additional VPN server requests as a duplicate request. The NPS server discards these duplicate VPN server requests.



If you look at the NPS server logs, you may see these additional requests being discarded. This behavior is by design to protect the end user from getting multiple requests for a single authentication attempt. Discarded requests in the NPS server event log don't indicate there's a problem with the NPS server or the Microsoft Entra multifactor authentication NPS extension.

To minimize discarded requests, we recommend that VPN servers are configured with a timeout of at least 60 seconds. If needed, or to reduce discarded requests in the event logs, you can increase the VPN server timeout value to 90 or 120 seconds.

Due to this UDP protocol behavior, the NPS server could receive a duplicate request and send another MFA prompt, even after the user has already responded to the initial request. To avoid this timing condition, the Microsoft Entra multifactor authentication NPS extension continues to filter and discard duplicate requests for up to 10 seconds after a successful response has been sent to the VPN server.



Again, you may see discarded requests in the NPS server event logs, even when the Microsoft Entra multifactor authentication prompt was successful. This is expected behavior, and doesn't indicate a problem with the NPS server or Microsoft Entra multifactor authentication NPS extension.

Plan your deployment

The NPS extension automatically handles redundancy, so you don't need a special configuration.

You can create as many Microsoft Entra multifactor authentication-enabled NPS servers as you need. If you do install multiple servers, you should use a different client certificate for each one of them. Creating a certificate for each server means that you can update each cert individually, and not worry about downtime across all your servers.

VPN servers route authentication requests, so they need to be aware of the new Microsoft Entra multifactor authentication-enabled NPS servers.

Prerequisites

The NPS extension is meant to work with your existing infrastructure. Make sure you have the following prerequisites before you begin.

Licenses

The NPS Extension for Microsoft Entra multifactor authentication is available to customers with [licenses for Microsoft Entra multifactor authentication](#) (included with Microsoft Entra ID P1 and Premium P2 or Enterprise Mobility + Security). Consumption-based licenses for Microsoft Entra multifactor authentication, such as per user or per authentication licenses, aren't compatible with the NPS extension.

Software

- Windows Server 2012 or later. Please note that [Windows Server 2012 has reached end of support](#).
- .NET Framework 4.7.2 or later is required for the Microsoft Graph PowerShell module.
- PowerShell version 5.1 or later. To check the version of PowerShell, run this command:

```
PowerShell  
  
PS C:\> $PSVersionTable.PSVersion  
Major Minor Build Revision  
----- ----- -----  
5       1       16232  1000
```

Libraries

- Visual Studio 2017 C++ Redistributable (x64) will be installed by the NPS Extension installer.
- Microsoft Graph PowerShell is also installed through a configuration script you run as part of the setup process, if not already present. There's no need to install a module in advance.

Obtain the directory tenant ID

As part of the configuration of the NPS extension, you must supply administrator credentials and the ID of your Microsoft Entra tenant. To get the tenant ID, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#).
2. Browse to **Identity > Settings**.

The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation sidebar with categories like Identity, Overview, Users, Groups, Devices, Applications, Roles & admins, Billing, Settings, Protection, Identity governance, External identities, and Hybrid management. The 'Overview' tab is selected in the main content area. At the top of the content area, there's a breadcrumb trail: ... > Microsoft Entra Connect | Cloud Sync > Cloud sync | Configurations > Microsoft Entra Connect | Cloud Sync > Cloud sync | Configurations >. Below the breadcrumb, it says 'Contoso ...'. There are buttons for '+ Add', 'Manage tenants', 'What's new', 'Preview features', and 'Got feedback?'. A note says 'Azure Active Directory is now Microsoft Entra ID. Learn more'. The main content area has tabs for 'Overview', 'Monitoring', 'Properties', 'Recommendations', and 'Tutorials'. A search bar says 'Search your tenant'. Under 'Basic information', there's a table with the following data:

Name	Contoso	Users	26
Tenant ID	aaaaabbb-0000-cccc-1111-dddd22	Groups	5
Primary domain	.com	Applications	1
License	Microsoft Entra ID Free	Devices	0

At the bottom, there's an 'Alerts' section.

Network requirements

The NPS server must be able to communicate with the following URLs over TCP port 443:

- `https://login.microsoftonline.com`
- `https://login.microsoftonline.us` (Azure Government)
- `https://login.chinacloudapi.cn` (Microsoft Azure operated by 21Vianet)
- `https://credentials.azure.com`
- `https://strongauthenticationservice.auth.microsoft.com`
- `https://strongauthenticationservice.auth.microsoft.us` (Azure Government)
- `https://strongauthenticationservice.auth.microsoft.cn` (Microsoft Azure operated by 21Vianet)
- `https://adnotifications.windowsazure.com`
- `https://adnotifications.windowsazure.us` (Azure Government)
- `https://adnotifications.windowsazure.cn` (Microsoft Azure operated by 21Vianet)

Additionally, connectivity to the following URLs is required to complete the [setup of the adapter using the provided PowerShell script](#):

- `https://onegetcdn.azureedge.net`
- `https://login.microsoftonline.com`
- `https://graph.microsoft.com`
- `https://provisioningapi.microsoftonline.com`
- `https://aadcdn.msauth.net`
- `https://www.powershellgallery.com`
- `https://go.microsoft.com`
- `https://aadcdn.msftauthimages.net`

The following table describes the ports and protocols required for the NPS extension. TCP 443 (inbound and outbound) is the only port needed from the NPS Extension server to Entra ID. The RADIUS ports are needed between the access point and the NPS Extension server.

[+] Expand table

Protocol	Port	Description
HTTPS	443	Enable user authentication against Entra ID (required when installing the extension)
UDP	1812	Common port for RADIUS Authentication by NPS
UDP	1645	Uncommon port for RADIUS Authentication by NPS
UDP	1813	Common port for RADIUS Accounting by NPS
UDP	1646	Uncommon port for RADIUS Accounting by NPS

Prepare your environment

Before you install the NPS extension, prepare your environment to handle the authentication traffic.

Enable the NPS role on a domain-joined server

The NPS server connects to Microsoft Entra ID and authenticates the MFA requests. Choose one server for this role. We recommend choosing a server that doesn't handle requests from other services, because the NPS extension throws errors for any requests that aren't RADIUS. The NPS server must be set up as the primary and secondary authentication server for your environment. It can't proxy RADIUS requests to another server.

1. On your server, open **Server Manager**. Select **Add Roles and Features Wizard** from the *Quickstart* menu.
2. For your installation type, choose **Role-based or feature-based installation**.
3. Select the **Network Policy and Access Services** server role. A window may pop up to inform you of additional required features to run this role.
4. Continue through the wizard until the *Confirmation* page. When ready, select **Install**.

It may take a few minutes to install the NPS server role. When finished, continue with the following sections to configure this server to handle incoming RADIUS requests

from the VPN solution.

Configure your VPN solution to communicate with the NPS server

Depending on which VPN solution you use, the steps to configure your RADIUS authentication policy vary. Configure your VPN policy to point to your RADIUS NPS server.

Sync domain users to the cloud

This step may already be complete on your tenant, but it's good to double-check that Microsoft Entra Connect has synchronized your databases recently.

1. Sign in to the [Microsoft Entra admin center](#) as a [Hybrid Identity Administrator](#).
2. Browse to **Identity > Hybrid management > Microsoft Entra Connect**.
3. Verify that your sync status is **Enabled** and that your last sync was less than an hour ago.

If you need to kick off a new round of synchronization, see [Microsoft Entra Connect Sync: Scheduler](#).

Determine which authentication methods your users can use

There are two factors that affect which authentication methods are available with an NPS extension deployment:

- The password encryption algorithm used between the RADIUS client (VPN, Netscaler server, or other) and the NPS servers.
 - **PAP** supports all the authentication methods of Microsoft Entra multifactor authentication in the cloud: phone call, one-way text message, mobile app notification, OATH hardware tokens, and mobile app verification code.
 - **CHAPV2** and **EAP** support phone call and mobile app notification.
- The input methods that the client application (VPN, Netscaler server, or other) can handle. For example, does the VPN client have some means to allow the user to type in a verification code from a text or mobile app?

You can [disable unsupported authentication methods](#) in Azure.

Note

Regardless of the authentication protocol that's used (PAP, CHAP, or EAP), if your MFA method is text-based (SMS, mobile app verification code, or OATH hardware token) and requires the user to enter a code or text in the VPN client UI input field, the authentication might succeed. *But* any RADIUS attributes that are configured in the Network Access Policy are *not* forwarded to the RADIUS client (the Network Access Device, like the VPN gateway). As a result, the VPN client might have more access than you want it to have, or less access or no access.

As a workaround, you can run the [CrpUsernameStuffing script](#) to forward RADIUS attributes that are configured in the Network Access Policy and allow MFA when the user's authentication method requires the use of a One-Time Passcode (OTP), such as SMS, a Microsoft Authenticator passcode, or a hardware FOB.

Register users for MFA

Before you deploy and use the NPS extension, users that are required to perform Microsoft Entra multifactor authentication need to be registered for MFA. To test the extension as you deploy it, you also need at least one test account that is fully registered for Microsoft Entra multifactor authentication.

If you need to create and configure a test account, use the following steps:

1. Sign in to <https://aka.ms/mfasetup> with a test account.
2. Follow the prompts to set up a verification method.
3. Sign in to the [Microsoft Entra admin center](#) as at least an **Authentication Policy Administrator**.
4. Browse to **Protection > Multifactor authentication** and enable for the test account.

Important

Make sure that users have successfully registered for Microsoft Entra multifactor authentication. If users have previously only registered for self-service password reset (SSPR), *StrongAuthenticationMethods* is enabled for their account. Microsoft Entra multifactor authentication is enforced when *StrongAuthenticationMethods* is configured, even if the user only registered for SSPR.

Combined security registration can be enabled that configures SSPR and Microsoft Entra multifactor authentication at the same time. For more information, see [Enable](#)

combined security information registration in Microsoft Entra ID.

You can also [force users to re-register authentication methods](#) if they previously only enabled SSPR.

Users who connect to the NPS server using username and password will be required to complete a multifactor authentication prompt.

Install the NPS extension

ⓘ Important

Install the NPS extension on a different server than the VPN access point.

Download and install the NPS extension for Microsoft Entra multifactor authentication

To download and install the NPS extension, complete the following steps:

1. [Download the NPS Extension](#) from the Microsoft Download Center.
2. Copy the binary to the Network Policy Server you want to configure.
3. Run *setup.exe* and follow the installation instructions. If you encounter errors, make sure that the [libraries from the prerequisite section](#) were successfully installed.

Upgrade the NPS extension

If you later upgrade an existing NPS extension install, to avoid a reboot of the underlying server, complete the following steps:

1. Uninstall the existing version.
2. Run the new installer.
3. Restart the *Network Policy Server (IAS)* service.

Run the PowerShell script

The installer creates a PowerShell script at `C:\Program Files\Microsoft\AzureMfa\Config` (where `C:\` is your installation drive). This PowerShell script performs the following actions each time it's run:

- Creates a self-signed certificate.

- Associates the public key of the certificate to the service principal on Microsoft Entra ID.
- Stores the certificate in the local machine certificate store.
- Grants access to the certificate's private key to Network User.
- Restarts the NPS service.

Unless you want to use your own certificates (instead of the self-signed certificates that the PowerShell script generates), run the PowerShell script to complete the NPS extension installation. If you install the extension on multiple servers, each server should have its own certificate.

To provide load-balancing capabilities or for redundancy, repeat these steps on additional NPS servers as desired:

1. Open a Windows PowerShell prompt as an administrator.
2. Change directories to where the installer created the PowerShell script:

```
PowerShell
cd "C:\Program Files\Microsoft\AzureMfa\Config"
```

3. Run the PowerShell script created by the installer.

You might be required to first enable TLS 1.2 for PowerShell to be able to connect and download packages properly:

```
[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12
```

Important

For customers that use the Azure for US Government or Azure operated by 21Vianet clouds, first edit the *AzureMfaNpsExtnConfigSetup.ps1* script to include the *Environment* parameters for the required cloud. For example, specify *-Environment USGov* or *-Environment China*. Environment options: USGov, USGovDoD, Germany, China, Global. Example: Connect-MgGraph -Scopes Application.ReadWrite.All -Environment USGov -NoWelcome -Verbose -ErrorAction Stop.

```
PowerShell
```

```
. \AzureMfaNpsExtnConfigSetup.ps1
```

4. When prompted, sign in to Microsoft Entra ID. A **Global Administrator** is needed to manage this feature.
5. PowerShell prompts for your tenant ID. Use the *Tenant ID* GUID that you copied in the prerequisites section.
6. A success message is shown when the script is finished.

If your previous computer certificate has expired, and a new certificate has been generated, you should delete any expired certificates. Having expired certificates can cause issues with the NPS Extension starting.

 **Note**

If you use your own certificates instead of generating certificates with the PowerShell script, make sure that they include the Client Authentication purpose and that the private key has **READ** permission granted to the user **NETWORK SERVICE**.

If you use version 1.2.2893.1 or later, the certificate's thumbprint can be used to identify the certificate. Set

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AzureMfa\CLIENT_CERT_IDENTIFIER to the thumbprint in **Registry Settings**. There have been issues with subject name lookup for some certificates. Using thumbprint works around this issue.

If you use version 1.2.2677.2 or earlier, the certificate must align to the NPS naming convention and the subject name must be **CN=<TenantID>,OU=Microsoft NPS Extension**.

Microsoft Azure Government or Microsoft Azure operated by 21Vianet additional steps

For customers that use the Azure Government or Azure operated by 21Vianet clouds, the following additional configuration steps are required on each NPS server.

 **Important**

Only configure these registry settings if you're an Azure Government or Azure operated by 21Vianet customer.

1. If you're an Azure Government or Azure operated by 21Vianet customer, open **Registry Editor** on the NPS server.
2. Navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AzureMfa`.
3. For Azure Government customers, set the following key values:

[+] Expand table

Registry key	Value
AZURE_MFA_HOSTNAME	strongauthenticationservice.auth.microsoft.us
AZURE_MFA_RESOURCE_HOSTNAME	adnotifications.windowsazure.us
STS_URL	https://login.microsoftonline.us/

4. For Microsoft Azure operated by 21Vianet customers, set the following key values:

[+] Expand table

Registry key	Value
AZURE_MFA_HOSTNAME	strongauthenticationservice.auth.microsoft.cn
AZURE_MFA_RESOURCE_HOSTNAME	adnotifications.windowsazure.cn
STS_URL	https://login.chinacloudapi.cn/

5. Repeat the previous two steps to set the registry key values for each NPS server.
6. Restart the NPS service for each NPS server.

For minimal impact, take each NPS server out of the NLB rotation one at a time and wait for all connections to drain.

Certificate rollover

With release 1.0.1.32 of the NPS extension, reading multiple certificates is now supported. This capability helps facilitate rolling certificate updates prior to their expiration. If your organization is running a previous version of the NPS extension, upgrade to version 1.0.1.32 or higher.

Certificates created by the `AzureMfaNpsExtnConfigSetup.ps1` script are valid for 2 years. Monitor certificates for expiration. Certificates for the NPS extension are placed in the *Local Computer* certificate store under *Personal* and are *Issued To* the tenant ID provided to the installation script.

When a certificate is approaching the expiration date, a new certificate should be created to replace it. This process is accomplished by running the `AzureMfaNpsExtnConfigSetup.ps1` again and keeping the same tenant ID when prompted. This process should be repeated on each NPS server in your environment.

Configure your NPS extension

With your environment prepared, and the NPS extension now installed on the required servers, you can configure the extension.

This section includes design considerations and suggestions for successful NPS extension deployments.

Configuration limitations

- The NPS extension for Microsoft Entra multifactor authentication doesn't include tools to migrate users and settings from MFA Server to the cloud. For this reason, we suggest using the extension for new deployments, rather than existing deployment. If you use the extension on an existing deployment, your users have to perform proof-up again to populate their MFA details in the cloud.
- The NPS extension doesn't support custom phone calls configured on Phone call settings. The default phone call language will be used (EN-US).
- The NPS extension uses the UPN from the on-premises AD DS environment to identify the user on Microsoft Entra multifactor authentication for performing the Secondary Auth. The extension can be configured to use a different identifier like alternate login ID or custom AD DS field other than UPN. For more information, see the article, [Advanced configuration options for the NPS extension for multifactor authentication](#).
- Not all encryption protocols support all verification methods.
 - **PAP** supports phone call, one-way text message, mobile app notification, and mobile app verification code
 - **CHAPV2** and **EAP** support phone call and mobile app notification

Control RADIUS clients that require MFA

Once you enable MFA for a RADIUS client using the NPS extension, all authentications for this client are required to perform MFA. If you want to enable MFA for some RADIUS clients but not others, you can configure two NPS servers and install the extension on only one of them.

Configure RADIUS clients that you want to require MFA to send requests to the NPS server configured with the extension, and other RADIUS clients to the NPS server not configured with the extension.

Prepare for users that aren't enrolled for MFA

If you have users that aren't enrolled for MFA, you can determine what happens when they try to authenticate. To control this behavior, use the setting *REQUIRE_USER_MATCH* in the registry path *HKLM\Software\Microsoft\AzureMFA*. This setting has a single configuration option:

[+] Expand table

Key	Value	Default
REQUIRE_USER_MATCH	TRUE/FALSE	Not set (equivalent to TRUE)

This setting determines what to do when a user isn't enrolled for MFA. When the key doesn't exist, is not set, or is set to *TRUE*, and the user isn't enrolled, the extension fails the MFA challenge.

When the key is set to *FALSE* and the user isn't enrolled, authentication proceeds without performing MFA. If a user is enrolled in MFA, they must authenticate with MFA even if *REQUIRE_USER_MATCH* is set to *FALSE*.

You can choose to create this key and set it to *FALSE* while your users are onboarding, and may not all be enrolled for Microsoft Entra multifactor authentication yet. However, since setting the key permits users that aren't enrolled for MFA to sign in, you should remove this key before going to production.

Troubleshooting

NPS extension health check script

The [Microsoft Entra multifactor authentication NPS Extension health check script ↗](#) performs a basic health check when troubleshooting the NPS extension. Run the script and choose one of available options.

How to fix the error "Service principal was not found" while running `AzureMfaNpsExtnConfigSetup.ps1` script?

If for any reason the "Azure Multi-factor Auth Client" service principal was not created in the tenant, it can be manually created by running PowerShell.

PowerShell

```
Connect-MgGraph -Scopes 'Application.ReadWrite.All'  
New-MgServicePrincipal -AppId 00001111-aaaa-2222-bbbb-3333cccc4444 -  
DisplayName "Azure Multi-Factor Auth Client"
```

1. Sign in to the [Microsoft Entra admin center](#) as at least an **Application Administrator**.
2. Browse to **Identity > Applications > Enterprise applications** > and search for "Azure Multi-factor Auth Client".
3. Click **Check properties for this app**. Confirm if the service principal is enabled or disabled.
4. Click the application entry > **Properties**.
5. If the option **Enabled for users to sign-in?** is set to **No**, set it to **Yes**.

Run the `AzureMfaNpsExtnConfigSetup.ps1` script again and it should not return the **Service principal was not found** error.

How do I verify that the client cert is installed as expected?

Look for the self-signed certificate created by the installer in the cert store, and check that the private key has "READ" permission granted to user *NETWORK SERVICE*. The cert has a subject name of **CN <tenantid>, OU = Microsoft NPS Extension**

Self-signed certificates generated by the `AzureMfaNpsExtnConfigSetup.ps1` script have a validity lifetime of two years. When verifying that the certificate is installed, you should also check that the certificate hasn't expired.

How can I verify that my client certificate is associated to my tenant in Microsoft Entra ID?

Open PowerShell command prompt and run the following commands:

PowerShell

```
Connect-MgGraph -Scopes 'Application.Read.All'  
(Get-MgServicePrincipal -Filter "appid eq '00001111-aaaa-2222-bbbb-  
3333cccc4444'" -Property "KeyCredentials").KeyCredentials |  
Format-List KeyId, DisplayName, StartDateTime, EndDateTime,  
@{Name = "Key"; Expression = {[System.Convert]::ToBase64String($_.Key) }},  
@{Name = "Thumbprint"; Expression = {  
[Convert]::ToBase64String($_.CustomKeyIdentifier)}}}
```

These commands print all the certificates associating your tenant with your instance of the NPS extension in your PowerShell session. Look for your certificate by exporting your client cert as a *Base-64 encoded X.509(.cer)* file without the private key, and compare it with the list from PowerShell. Compare the thumbprint of the certificate installed on the server to this one. The certificate thumbprints should match.

StartDateTime and *EndDateTime* timestamps, which are in human-readable form, can be used to filter out obvious misfits if the command returns more than one cert.

Why cannot I sign in?

Check that your password hasn't expired. The NPS extension doesn't support changing passwords as part of the sign-in workflow. Contact your organization's IT Staff for further assistance.

Why are my requests failing with security token error?

This error could be due to one of several reasons. Use the following steps to troubleshoot:

1. Restart your NPS server.
2. Verify that client cert is installed as expected.
3. Verify that the certificate is associated with your tenant on Microsoft Entra ID.
4. Verify that <https://login.microsoftonline.com/> is accessible from the server running the extension.

Why does authentication fail with an error in HTTP logs stating that the user is not found?

Verify that AD Connect is running, and that the user is present in both the on-premises AD DS environment and in Microsoft Entra ID.

Why do I see HTTP connect errors in logs with all my authentications failing?

Verify that <https://adnotifications.windowsazure.com>, <https://strongauthenticationservice.auth.microsoft.com> is reachable from the server running the NPS extension.

Why is authentication not working, despite a valid certificate being present?

If your previous computer certificate has expired, and a new certificate has been generated, delete any expired certificates. Expired certificates can cause issues with the NPS extension starting.

To check if you have a valid certificate, check the local *Computer Account's Certificate Store* using MMC, and ensure the certificate hasn't passed its expiry date. To generate a newly valid certificate, rerun the steps from [Run the PowerShell installer script](#).

Why do I see discarded requests in the NPS server logs?

A VPN server may send repeated requests to the NPS server if the timeout value is too low. The NPS server detects these duplicate requests and discards them. This behavior is by design, and doesn't indicate a problem with the NPS server or the Microsoft Entra multifactor authentication NPS extension.

For more information on why you see discarded packets in the NPS server logs, see [RADIUS protocol behavior and the NPS extension](#) at the start of this article.

How do I get Microsoft Authenticator number matching to work with NPS?

Although NPS doesn't support number matching, the latest NPS extension does support time-based one-time password (TOTP) methods such as the TOTP available in Microsoft Authenticator, other software tokens, and hardware FOBs. TOTP sign-in provides better security than the alternative Approve/Deny experience. Make sure you run the latest version of the [NPS extension](#).

After May 8, 2023, when number matching is enabled for all users, anyone who performs a RADIUS connection with NPS extension version 1.2.2216.1 or later will be prompted to sign in with a TOTP method instead.

Users must have a TOTP authentication method registered to see this behavior. Without a TOTP method registered, users continue to see [Approve/Deny](#).

Prior to the release of NPS extension version 1.2.2216.1 after May 8, 2023, organizations that run earlier versions of NPS extension can modify the registry to require users to enter a TOTP. For more information, see [NPS extension](#).

Managing the TLS/SSL Protocols and Cipher Suites

It's recommended that older and weaker cipher suites be disabled or removed unless required by your organization. Information on how to complete this task can be found in the article, [Managing SSL/TLS Protocols and Cipher Suites for AD FS](#)

Additional troubleshooting

Additional troubleshooting guidance and possible solutions can be found in the article, [Resolve error messages from the NPS extension for Microsoft Entra multifactor authentication](#).

Next steps

- Overview and configuration of Network Policy Server in Windows Server
- Configure alternate IDs for login, or set up an exception list for IPs that shouldn't perform two-step verification in [Advanced configuration options for the NPS extension for multifactor authentication](#)
- Learn how to integrate [Remote Desktop Gateway](#) and [VPN servers](#) using the NPS extension
- [Resolve error messages from the NPS extension for Microsoft Entra multifactor authentication](#)

Feedback

Was this page helpful?



[Provide product feedback ↗](#)

Advanced configuration options for the NPS extension for multifactor authentication

Article • 03/04/2025

The Network Policy Server (NPS) extension extends your cloud-based Microsoft Entra multifactor authentication features into your on-premises infrastructure. This article assumes that you already have the extension installed, and now want to know how to customize the extension for your needs.

Alternate sign-in ID

Since the NPS extension connects to both your on-premises and cloud directories, you might encounter an issue where your on-premises user principal names (UPNs) don't match the names in the cloud. To solve this problem, use alternate sign-in IDs.

Within the NPS extension, you can designate an Active Directory attribute to be used as the UPN for Microsoft Entra multifactor authentication. This enables you to protect your on-premises resources with two-step verification without modifying your on-premises UPNs.

To configure alternate sign-in IDs, go to `HKLM\SOFTWARE\Microsoft\AzureMfa` and edit the following registry values:

[+] Expand table

Name	Type	Default value	Description
LDAP_ALTERNATE_LOGINID_ATTRIBUTE	string	Empty	Designate the name of Active Directory attribute that you want to use as the UPN. This attribute is used as the AlternateLoginId attribute. If this registry value is set to a valid Active Directory attribute (for example, mail or displayName), then the attribute's value is used as the user's UPN for authentication. If this registry value is empty or not configured, then AlternateLoginId is disabled

Name	Type	Default value	Description
			and the user's UPN is used for authentication.
LDAP_FORCE_GLOBAL_CATALOG	boolean	False	<p>Use this flag to force the use of Global Catalog for LDAP searches when looking up AlternateLoginId. Configure a domain controller as a Global Catalog, add the AlternateLoginId attribute to the Global Catalog, and then enable this flag.</p> <p>If LDAP_LOOKUP_FORESTS is configured (not empty), this flag is enforced as true, regardless of the value of the registry setting. In this case, the NPS extension requires the Global Catalog to be configured with the AlternateLoginId attribute for each forest.</p>
LDAP_LOOKUP_FORESTS	string	Empty	Provide a semi-colon separated list of forests to search. For example, <i>contoso.com;foobar.com</i> . If this registry value is configured, the NPS extension iteratively searches all the forests in the order in which they were listed, and returns the first successful AlternateLoginId value. If this registry value isn't configured, the AlternateLoginId lookup is confined to the current domain.

To troubleshoot problems with alternate sign-in IDs, use the recommended steps for [Alternate sign-in ID errors](#).

IP exceptions

If you need to monitor server availability, like if load balancers verify which servers are running before sending workloads, you don't want verification requests to block these checks. Instead, create a list of IP addresses that you know are used by service accounts, and disable multifactor authentication requirements for that list.

To configure an IP allowed list, go to `HKLM\SOFTWARE\Microsoft\AzureMfa` and configure the following registry value:

[+] Expand table

Name	Type	Default value	Description
IP_WHITELIST	string	Empty	<p>Provide a semi-colon separated list of IP addresses. Include the IP addresses of machines where service requests originate, like the NAS/VPN server. IP ranges and subnets aren't supported.</p> <p>For example, <code>10.0.0.1;10.0.0.2;10.0.0.3</code>.</p>

ⓘ Note

This registry key isn't created by default by the installer and an error appears in the AuthZOptCh log when the service is restarted. This error in the log can be ignored, but if this registry key is created and left empty if not needed then the error message doesn't return.

When a request comes in from an IP address that exists in the `IP_WHITELIST`, two-step verification is skipped. The IP list is compared to the IP address that is provided in the `ratNASIPAddress` attribute of the RADIUS request. If a RADIUS request comes in without the `ratNASIPAddress` attribute, a warning is logged: "IP_WHITE_LIST_WARNING::IP Whitelist is being ignored as the source IP is missing in the RADIUS request NasIpAddress attribute."

Next steps

- [Resolve error messages from the NPS extension for Microsoft Entra multifactor authentication](#)
- [Use REQUIRE_USER_MATCH to prepare for users that aren't enrolled for MFA](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Integrate P2S RADIUS authentication with NPS for multifactor authentication

Article • 11/19/2024

The article helps you integrate Network Policy Server (NPS) with Azure VPN Gateway RADIUS authentication to deliver multifactor authentication (MFA) for point-to-site (P2S) VPN connections.

Prerequisites

- **Microsoft Entra ID:** In order to enable MFA, the users must be in Microsoft Entra ID, which must be synced from either the on-premises environment, or the cloud environment.
 - The user must have completed the autoenrollment process for MFA. For more information, see [Set up my account for two-step verification ↗](#).
 - If your MFA is text-based (SMS, mobile app verification code, etc.) and requires the user to enter a code or text in the VPN client UI, authentication won't succeed and isn't a supported scenario.
- **Route-based VPN gateway:** You must already have a route-based VPN gateway. For steps to create a route-based VPN gateway, see [Tutorial: Create and manage a VPN gateway](#).
- **NPS:** You must already have installed the Network Policy Server and configured the VPN policy for RADIUS.
 - For steps to install the Network Policy Server, see [Install the Network Policy Server \(NPS\)](#).
 - For steps to create a VPN policy for RADIUS, see [Create a VPN policy for RADIUS](#).

Create RADIUS client

1. Create the RADIUS client by specifying the following settings:

- **Friendly Name:** Type any name.
- **Address (IP or DNS):** Use the value specified for your VPN gateway Gateway Subnet. For example, 10.1.255.0/27.

- **Shared secret:** Type any secret key, and remember it for later use.
2. On the **Advanced** tab, set the vendor name to **RADIUS Standard** and make sure that the **Additional Options** check box isn't selected. Then, select **OK**.
 3. Go to **Policies > Network Policies**. Double-click **Connections to Microsoft Routing and Remote Access server** policy. Select **Grant access**, and then select **OK**.

Configure the VPN gateway

1. In the Azure portal, open your virtual network gateway (VPN gateway).
2. On the **Overview** page, verify that the **Gateway type** is set to **VPN** and that the **VPN type** is **route-based**.
3. In the left pane, expand **Settings** and select **Point to site configuration > Configure now**.
4. View the **Point-to-site configuration** page.

The screenshot shows the 'Point-to-site configuration' page for a virtual network gateway named 'VNet1GW'. The left sidebar lists various configuration options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Configuration and Connections), Point-to-site configuration (selected), NAT Rules, Maintenance, Properties, Locks, Monitoring, and Automation. The main content area has several input fields and dropdown menus. At the top, there's a search bar, save, discard, delete, and download VPN client buttons. The 'Address pool' field contains '172.16.201.0/24'. The 'Tunnel type' dropdown is set to 'IKEv2 and OpenVPN (SSL)'. Under 'IPsec / IKE policy', 'Default' is selected. The 'Authentication type' dropdown is set to 'RADIUS authentication'. Below these, there's a section for 'Public IP address for User VPN configuration' with a note: 'A third public IP address is required to use a User VPN configuration with an availability zone SKU gateway in active-active mode.' It shows two radio button options: 'Create new' (selected) and 'Use existing'. A magnifying glass icon is at the bottom right of the form.

5. On the **Point-to-site configuration** page, configure the following settings:

- **Address pool:** This value specifies the client address pool from which the VPN clients receive an IP address when they connect to the VPN gateway. The address pool must be a private IP address range that doesn't overlap with the virtual network address range. For example, **172.16.201.0/24**.
- **Tunnel type:** Select the tunnel type. For example, select **IKEv2 and OpenVPN (SSL)**.
- **Authentication type:** Select **RADIUS authentication**.

- If you have an active-active VPN gateway, a third public IP address is required. You can create a new public IP address using the example value **VNet1GWpip3**.
- **Primary Server IP address:** Type the IP address of the Network Policy Server (NPS).
- **Primary Server secret:** Type the shared secret that you specified when you created the RADIUS client on the NPS.

6. At the top of the page, **Save** the configuration settings.

After the settings are saved, you can click **Download VPN Client** to download the VPN client configuration package and use the settings to configure the VPN client. For more information about P2S VPN client configuration, see the [Point-to-site client configuration requirements](#) table.

Integrate NPS with Microsoft Entra MFA

Use the following links to integrate your NPS infrastructure with Microsoft Entra multifactor authentication:

- [How it works: Microsoft Entra multifactor authentication](#)
- [Integrate your existing NPS infrastructure with Microsoft Entra multifactor authentication](#)

Next steps

For steps to configure your VPN client, see the [Point-to-site client configuration requirements](#) table.

Feedback

Was this page helpful?



[Provide product feedback ↗](#) | [Get help at Microsoft Q&A](#)

Integrate your Remote Desktop Gateway infrastructure using the Network Policy Server (NPS) extension and Microsoft Entra ID

Article • 03/04/2025

This article provides details for integrating your Remote Desktop Gateway infrastructure with Microsoft Entra multifactor authentication using the Network Policy Server (NPS) extension for Microsoft Azure.

The Network Policy Server (NPS) extension for Azure allows customers to safeguard Remote Authentication Dial-In User Service (RADIUS) client authentication using Azure's cloud-based [multifactor authentication](#). This solution provides two-step verification for adding a second layer of security to user sign-ins and transactions.

This article provides step-by-step instructions for integrating the NPS infrastructure with Microsoft Entra multifactor authentication using the NPS extension for Azure. This enables secure verification for users attempting to sign in to a Remote Desktop Gateway.

Note

This article shouldn't be used with MFA Server deployments and should only be used with Microsoft Entra multifactor authentication (Cloud-based) deployments.

The Network Policy and Access Services (NPS) gives organizations the ability to do the following:

- Define central locations for the management and control of network requests by specifying who can connect, what times of day connections are allowed, the duration of connections, and the level of security that clients must use to connect, and so on. Rather than specifying these policies on each VPN or Remote Desktop (RD) Gateway server, these policies can be specified once in a central location. The RADIUS protocol provides the centralized Authentication, Authorization, and Accounting (AAA).
- Establish and enforce Network Access Protection (NAP) client health policies that determine whether devices are granted unrestricted or restricted access to network resources.
- Provide a means to enforce authentication and authorization for access to 802.1x-capable wireless access points and Ethernet switches.

Typically, organizations use NPS (RADIUS) to simplify and centralize the management of VPN policies. However, many organizations also use NPS to simplify and centralize the management of RD Desktop Connection Authorization Policies (RD CAPs).

Organizations can also integrate NPS with Microsoft Entra multifactor authentication to enhance security and provide a high level of compliance. This helps ensure that users establish two-step verification to sign in to the Remote Desktop Gateway. For users to be granted access, they must provide their username/password combination along with information that the user has in their control. This information must be trusted and not easily duplicated, such as a cell phone number, landline number, application on a mobile device, and so on. RDG currently supports phone call and **Approve/Deny** push notifications from Microsoft authenticator app methods for 2FA. For more information about supported authentication methods, see the section [Determine which authentication methods your users can use](#).

If your organization uses Remote Desktop Gateway and the user is registered for a TOTP code along with Authenticator push notifications, the user can't meet the MFA challenge and the Remote Desktop Gateway sign-in fails. In that case, you can override this behaviour by creating a new registry key (**OVERRIDE_NUMBER_MATCHING_WITH OTP**) to fallback to push notifications to Approve/Deny with Authenticator. To perform it, follow [NPS extension override number matching](#) procedure, assuming final value will be **OVERRIDE_NUMBER_MATCHING_WITH OTP = FALSE**.

Prior to the availability of the NPS extension for Azure, customers who wished to implement two-step verification for integrated NPS and Microsoft Entra multifactor authentication environments had to configure and maintain a separate MFA Server in the on-premises environment as documented in [Remote Desktop Gateway and Azure Multi-Factor Authentication Server using RADIUS](#).

The availability of the NPS extension for Azure now gives organizations the choice to deploy either an on-premises based MFA solution or a cloud-based MFA solution to secure RADIUS client authentication.

Authentication Flow

For users to be granted access to network resources through a Remote Desktop Gateway, they must meet the conditions specified in one RD Connection Authorization Policy (RD CAP) and one RD Resource Authorization Policy (RD RAP). RD CAPs specify who is authorized to connect to RD Gateways. RD RAPs specify the network resources, such as remote desktops or remote apps, that the user is allowed to connect to through the RD Gateway.

An RD Gateway can be configured to use a central policy store for RD CAPs. RD RAPs can't use a central policy, as they're processed on the RD Gateway. An example of an RD Gateway

configured to use a central policy store for RD CAPs is a RADIUS client to another NPS server that serves as the central policy store.

When the NPS extension for Azure is integrated with the NPS and Remote Desktop Gateway, the successful authentication flow is as follows:

1. The Remote Desktop Gateway server receives an authentication request from a remote desktop user to connect to a resource, such as a Remote Desktop session. Acting as a RADIUS client, the Remote Desktop Gateway server converts the request to a RADIUS Access-Request message and sends the message to the RADIUS (NPS) server where the NPS extension is installed.
2. The username and password combination is verified in Active Directory and the user is authenticated.
3. If all the conditions as specified in the NPS Connection Request and the Network Policies are met (for example, time of day or group membership restrictions), the NPS extension triggers a request for secondary authentication with Microsoft Entra multifactor authentication.
4. Microsoft Entra multifactor authentication communicates with Microsoft Entra ID, retrieves the user's details, and performs the secondary authentication using supported methods.
5. Upon success of the MFA challenge, Microsoft Entra multifactor authentication communicates the result to the NPS extension.
6. The NPS server, where the extension is installed, sends a RADIUS Access-Accept message for the RD CAP policy to the Remote Desktop Gateway server.
7. The user is granted access to the requested network resource through the RD Gateway.

Prerequisites

This section details the prerequisites necessary before integrating Microsoft Entra multifactor authentication with the Remote Desktop Gateway. Before you begin, you must have the following prerequisites in place.

- Remote Desktop Services (RDS) infrastructure
- Microsoft Entra multifactor authentication License
- Windows Server software
- Network Policy and Access Services (NPS) role
- Microsoft Entra synced with on-premises Active Directory
- Microsoft Entra GUID ID

Remote Desktop Services (RDS) infrastructure

You must have a working Remote Desktop Services (RDS) infrastructure in place. If you don't, then you can quickly create this infrastructure in Azure using the following quickstart template: [Create Remote Desktop Session Collection deployment](#).

If you wish to manually create an on-premises RDS infrastructure quickly for testing purposes, follow the steps to deploy one. Learn more: [Deploy RDS with Azure quickstart](#) and [Basic RDS infrastructure deployment](#).

Windows Server software

The NPS extension requires Windows Server 2008 R2 SP1 or above with the NPS role service installed. All the steps in this section were performed using Windows Server 2016.

Network Policy and Access Services (NPS) role

The NPS role service provides the RADIUS server and client functionality and Network Access Policy health service. This role must be installed on at least two computers in your infrastructure: The Remote Desktop Gateway and another member server or domain controller. By default, the role is already present on the computer configured as the Remote Desktop Gateway. You must also install the NPS role on at least one another computer, such as a domain controller or member server.

For information on installing the NPS role service Windows Server 2012 or older, see [Install a NAP Health Policy Server](#). For a description of best practices for NPS, including the recommendation to install NPS on a domain controller, see [Best Practices for NPS](#).

Microsoft Entra synced with on-premises Active Directory

To use the NPS extension, on-premises users must be synced with Microsoft Entra ID and enabled for MFA. This section assumes that on-premises users are synced with Microsoft Entra ID using AD Connect. For information on Microsoft Entra Connect, see [Integrate your on-premises directories with Microsoft Entra ID](#).

Microsoft Entra GUID ID

To install NPS extension, you need to know the GUID of the Microsoft Entra ID. The following provides instructions for finding the GUID of the Microsoft Entra ID.

Configure multifactor authentication

This section provides instructions for integrating Microsoft Entra multifactor authentication with the Remote Desktop Gateway. As an administrator, you must configure the Microsoft Entra multifactor authentication service before users can self-register their multifactor devices or applications.

Follow the steps in [Getting started with Microsoft Entra multifactor authentication in the cloud](#) to enable MFA for your Microsoft Entra users.

Configure accounts for two-step verification

Once an account has been enabled for MFA, you can't sign in to resources governed by the MFA policy until you have successfully configured a trusted device to use for the second authentication factor and have authenticated using two-step verification.

Follow the steps in [What does Microsoft Entra multifactor authentication mean for me?](#) to understand and properly configure your devices for MFA with your user account.

Important

The sign-in behavior for Remote Desktop Gateway doesn't provide the option to enter a verification code with Microsoft Entra multifactor authentication. A user account must be configured for phone verification or the Microsoft Authenticator App with **Approve/Deny** push notifications.

If neither phone verification or the Microsoft Authenticator App with **Approve/Deny** push notifications is configured for a user, the user won't be able to complete the Microsoft Entra multifactor authentication challenge and sign in to Remote Desktop Gateway.

The SMS text method doesn't work with Remote Desktop Gateway because it doesn't provide the option to enter a verification code.

Install and configure NPS extension

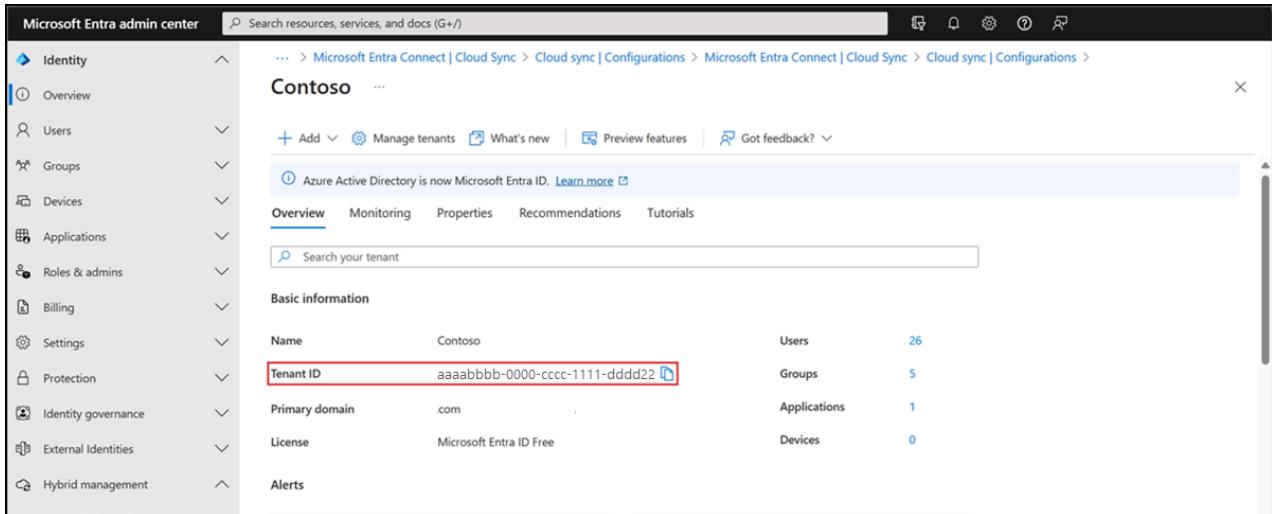
This section provides instructions for configuring RDS infrastructure to use Microsoft Entra multifactor authentication for client authentication with the Remote Desktop Gateway.

Obtain the directory tenant ID

As part of the configuration of the NPS extension, you must supply administrator credentials and the ID of your Microsoft Entra tenant. To get the tenant ID, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#).

2. Browse to **Entra ID > Overview > Properties.**



The screenshot shows the Microsoft Entra admin center interface. On the left is a navigation sidebar with categories like Identity, Overview, Users, Groups, Devices, Applications, Roles & admins, Billing, Settings, Protection, Identity governance, External Identities, and Hybrid management. The 'Overview' section is selected. In the center, the tenant 'Contoso' is displayed. At the top of the main area, there's a breadcrumb trail: ... > Microsoft Entra Connect | Cloud Sync > Cloud sync | Configurations > Microsoft Entra Connect | Cloud Sync > Cloud sync | Configurations >. Below the breadcrumb is a toolbar with Add, Manage tenants, What's new, Preview features, and Got feedback? buttons. A status message says 'Azure Active Directory is now Microsoft Entra ID. [Learn more](#)'. The 'Overview' tab is active, followed by Monitoring, Properties, Recommendations, and Tutorials. A search bar is present. Under 'Basic information', there's a table with the following data:

Name	Contoso	Users	26
Tenant ID	aaaabbbb-0000-cccc-1111-dddd22	Groups	5
Primary domain	.com	Applications	1
License	Microsoft Entra ID Free	Devices	0

Install the NPS extension

Install the NPS extension on a server that has the Network Policy and Access Services (NPS) role installed. This functions as the RADIUS server for your design.

i Important

Don't install the NPS extension on your Remote Desktop Gateway (RDG) server. The RDG server doesn't use the RADIUS protocol with its client, so the extension can't interpret and perform the MFA.

When the RDG server and NPS server with NPS extension are different servers, RDG uses NPS internally to talk to other NPS servers and uses RADIUS as the protocol to correctly communicate.

1. Download the [NPS extension](#).
2. Copy the setup executable file (NpsExtnForAzureMfaInstaller.exe) to the NPS server.
3. On the NPS server, double-select **NpsExtnForAzureMfaInstaller.exe**. If prompted, select **Run**.
4. In the NPS Extension For Microsoft Entra multifactor authentication Setup dialog box, review the software license terms, check **I agree to the license terms and conditions**, and select **Install**.
5. In the NPS Extension For Microsoft Entra multifactor authentication Setup dialog box, select **Close**.

Configure certificates for use with the NPS extension using a PowerShell script

Next, you need to configure certificates for use by the NPS extension to ensure secure communications and assurance. The NPS components include a PowerShell script that configures a self-signed certificate for use with NPS.

The script performs the following actions:

- Creates a self-signed certificate
- Associates public key of certificate to service principal on Microsoft Entra ID
- Stores the cert in the local machine store
- Grants access to the certificate's private key to the network user
- Restarts Network Policy Server service

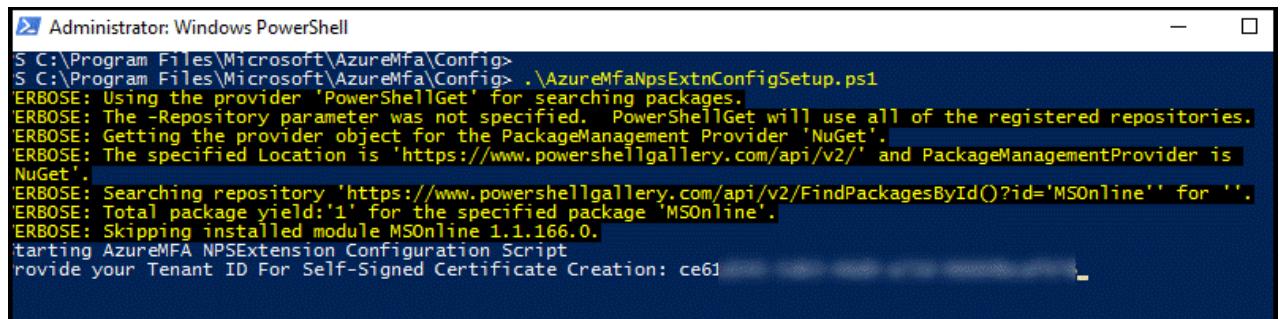
If you want to use your own certificates, you need to associate the public key of your certificate to the service principal on Microsoft Entra ID, and so on.

To use the script, provide the extension with your Microsoft Entra Admin credentials and the Microsoft Entra tenant ID that you copied earlier. Run the script on each NPS server where you installed the NPS extension. Then do the following:

1. Open an administrative Windows PowerShell prompt.
2. At the PowerShell prompt, type `cd 'c:\Program Files\Microsoft\AzureMfa\Config'`, and press **ENTER**.
3. Type `.\AzureMfaNpsExtnConfigSetup.ps1`, and press **ENTER**. The script checks to see if the PowerShell module is installed. If not installed, the script installs the module for you.

```
S C:\Program Files\Microsoft\AzureMfa\Config>
S C:\Program Files\Microsoft\AzureMfa\Config> .\AzureMfaNpsExtnConfigSetup.ps1
[ERBOSE]: Using the provider 'PowerShellGet' for searching packages.
[ERBOSE]: The -Repository parameter was not specified. PowerShellGet will use all of the registered repositories.
[ERBOSE]: Getting the provider object for the PackageManagement Provider 'NuGet'.
[ERBOSE]: The specified Location is 'https://www.powershellgallery.com/api/v2/' and PackageManagementProvider is 'NuGet'.
[ERBOSE]: Searching repository 'https://www.powershellgallery.com/api/v2/FindPackagesById()?id='MSOnline'' for ''.
[ERBOSE]: Total package yield:1 for the specified package 'MSOnline'.
[ERBOSE]: Skipping installed module MSOnline 1.1.166.0.
```

4. After the script verifies the installation of the PowerShell module, it displays the PowerShell module dialog box. In the dialog box, enter your Microsoft Entra admin credentials and password, and select **Sign In**.
5. When prompted, paste the *Tenant ID* you copied to the clipboard earlier, and press **ENTER**.



```
S C:\Program Files\Microsoft\AzureMfa\Config>
S C:\Program Files\Microsoft\AzureMfa\Config> .\AzureMfaNpsExtnConfigSetup.ps1
[ERBOSE: Using the provider 'PowerShellGet' for searching packages.
[ERBOSE: The -Repository parameter was not specified. PowerShellGet will use all of the registered repositories.
[ERBOSE: Getting the provider object for the PackageManagement Provider 'NuGet'.
[ERBOSE: The specified Location is 'https://www.powershellgallery.com/api/v2/' and PackageManagementProvider is NuGet'.
[ERBOSE: Searching repository 'https://www.powershellgallery.com/api/v2/FindPackagesById()?id='MSOnline'' for ''.
[ERBOSE: Total package yield:'1' for the specified package 'MSOnline'.
[ERBOSE: Skipping installed module MSOnline 1.1.166.0.
Starting AzureMFA NPSExtension Configuration Script
Provide your Tenant ID For Self-Signed Certificate Creation: ce61
```

6. The script creates a self-signed certificate and performs other configuration changes.

Configure NPS components on Remote Desktop Gateway

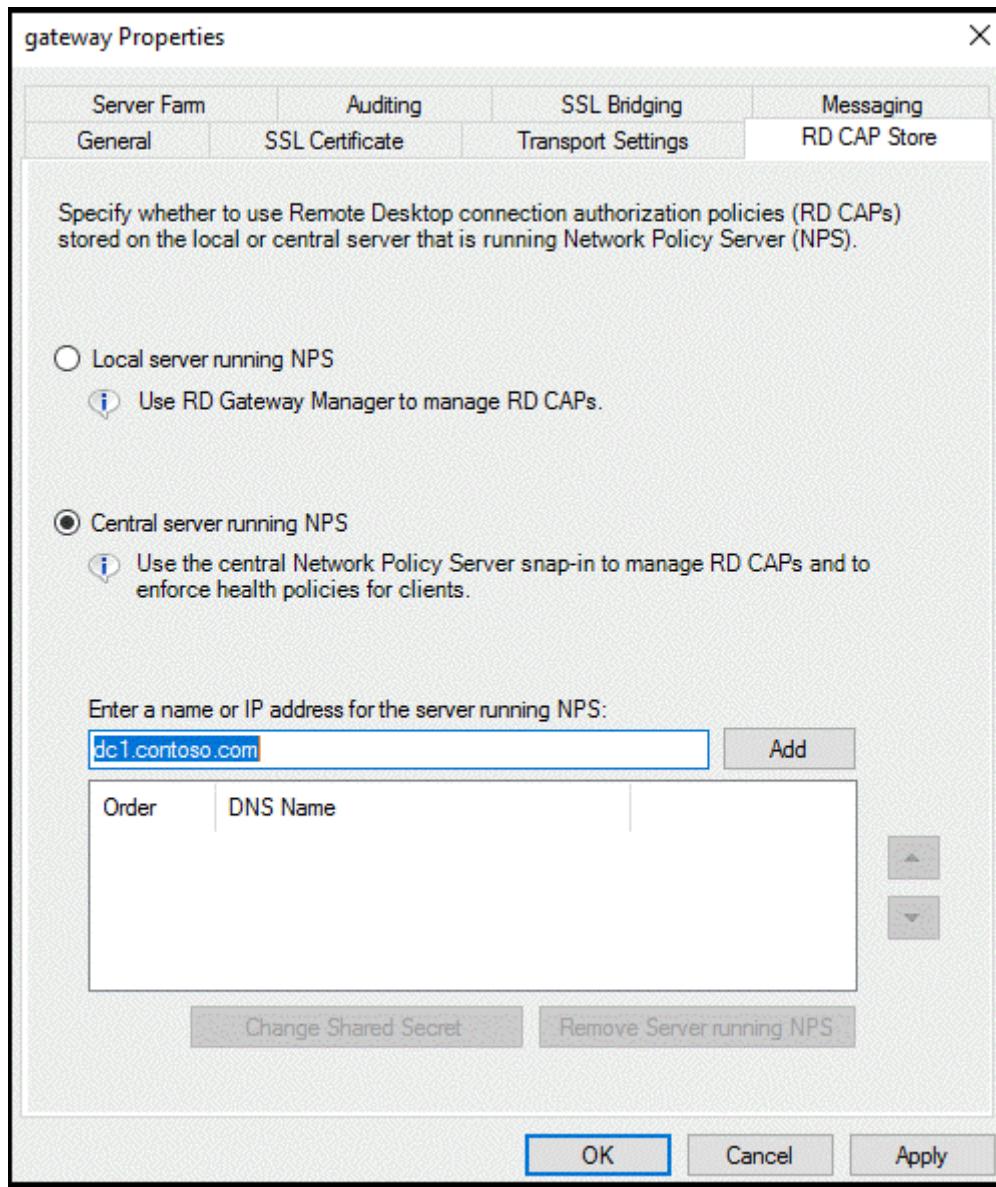
In this section, you configure the Remote Desktop Gateway connection authorization policies and other RADIUS settings.

The authentication flow requires that RADIUS messages be exchanged between the Remote Desktop Gateway and the NPS server where the NPS extension is installed. This means that you must configure RADIUS client settings on both Remote Desktop Gateway and the NPS server where the NPS extension is installed.

Configure Remote Desktop Gateway connection authorization policies to use central store

Remote Desktop connection authorization policies (RD CAPs) specify the requirements for connecting to a Remote Desktop Gateway server. RD CAPs can be stored locally (default) or they can be stored in a central RD CAP store that is running NPS. To configure integration of Microsoft Entra multifactor authentication with RDS, you need to specify the use of a central store.

1. On the RD Gateway server, open Server Manager.
2. On the menu, select Tools, point to **Remote Desktop Services**, and then select **Remote Desktop Gateway Manager**.
3. In the RD Gateway Manager, right-select [Server Name] (Local), and select **Properties**.
4. In the Properties dialog box, select the **RD CAP Store** tab.
5. On the RD CAP Store tab, select **Central server running NPS**.
6. In the **Enter a name or IP address for the server running NPS** field, type the IP address or server name of the server where you installed the NPS extension.



7. Select **Add**.

8. In the **Shared Secret** dialog box, enter a shared secret, and then select **OK**. Ensure you record this shared secret and store the record securely.

① Note

Shared secret is used to establish trust between the RADIUS servers and clients. Create a long and complex secret.

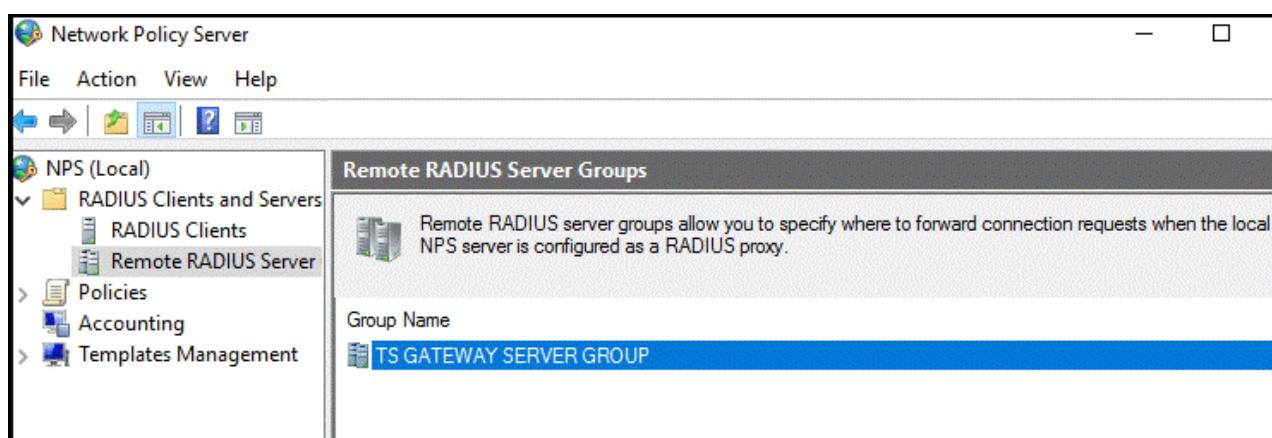


9. Select **OK** to close the dialog box.

Configure RADIUS timeout value on Remote Desktop Gateway NPS

To ensure there is time to validate users' credentials, perform two-step verification, receive responses, and respond to RADIUS messages, it's necessary to adjust the RADIUS timeout value.

1. On the RD Gateway server, open Server Manager. On the menu, select **Tools**, and then select **Network Policy Server**.
2. In the **NPS (Local)** console, expand **RADIUS Clients and Servers**, and select **Remote RADIUS Server**.

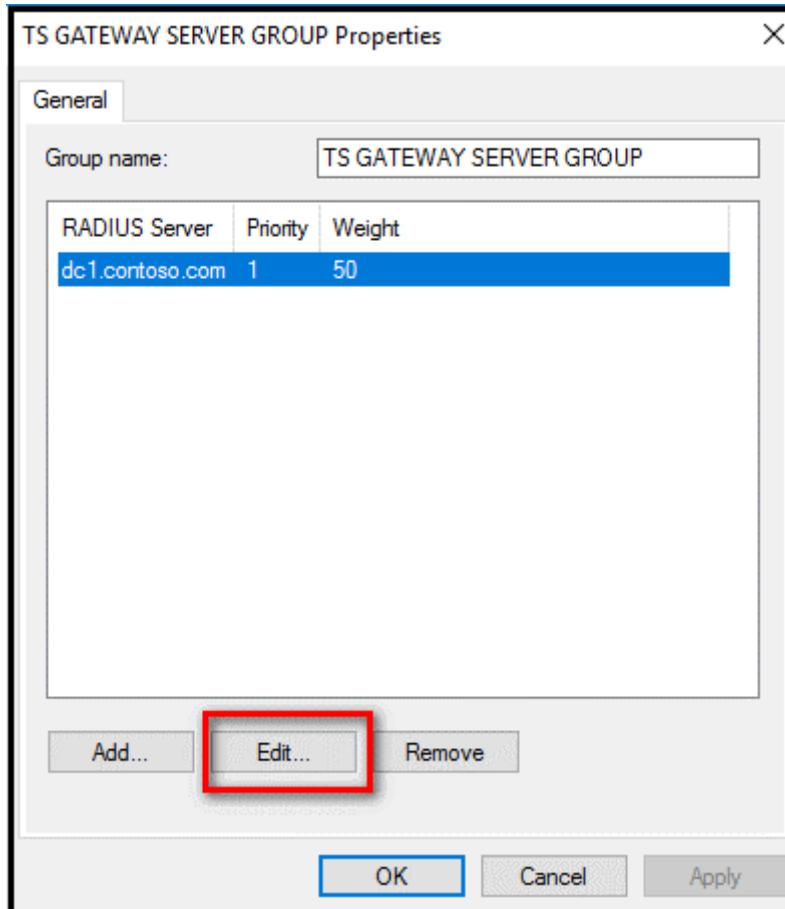


3. In the details pane, double-select **TS GATEWAY SERVER GROUP**.

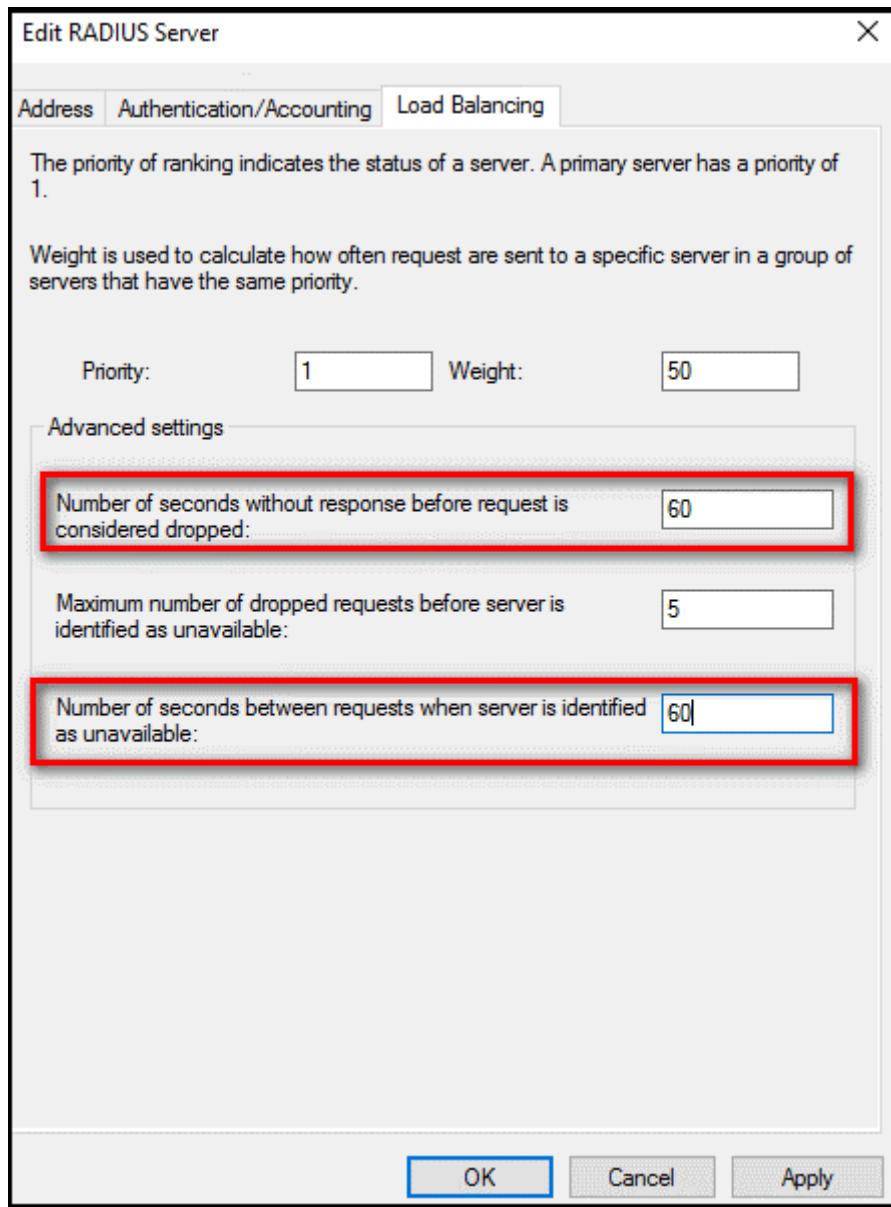
Note

This RADIUS Server Group was created when you configured the central server for NPS policies. The RD Gateway forwards RADIUS messages to this server or group of servers, if more than one in the group.

4. In the **TS GATEWAY SERVER GROUP Properties** dialog box, select the IP address or name of the NPS server you configured to store RD CAPs, and then select **Edit**.



5. In the **Edit RADIUS Server** dialog box, select the **Load Balancing** tab.
6. In the **Load Balancing** tab, in the **Number of seconds without response before request is considered dropped** field, change the default value from 3 to a value between 30 and 60 seconds.
7. In the **Number of seconds between requests when server is identified as unavailable** field, change the default value of 30 seconds to a value that is equal to or greater than the value you specified in the previous step.



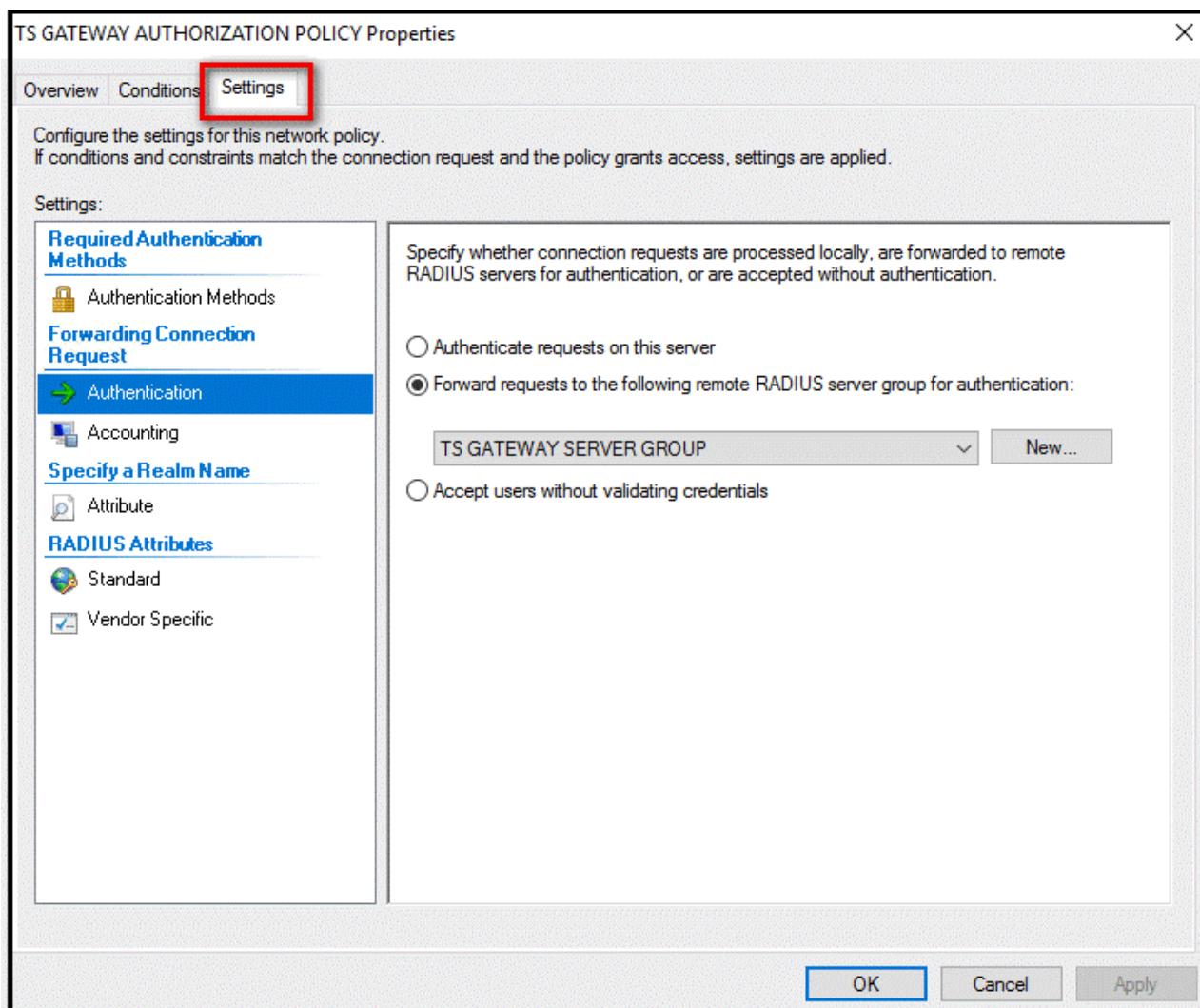
8. Select OK two times to close the dialog boxes.

Verify Connection Request Policies

By default, when you configure the RD Gateway to use a central policy store for connection authorization policies, the RD Gateway is configured to forward CAP requests to the NPS server. The NPS server with the Microsoft Entra multifactor authentication extension installed, processes the RADIUS access request. The following steps show you how to verify the default connection request policy.

1. On the RD Gateway, in the NPS (Local) console, expand **Policies**, and select **Connection Request Policies**.
2. Double-select **TS GATEWAY AUTHORIZATION POLICY**.
3. In the **TS GATEWAY AUTHORIZATION POLICY** properties dialog box, select the **Settings** tab.

4. On **Settings** tab, under Forwarding Connection Request, select **Authentication**. RADIUS client is configured to forward requests for authentication.



5. Select Cancel.

! Note

For more information about creating a connection request policy, see the article [Configure connection request policies](#) documentation for the same.

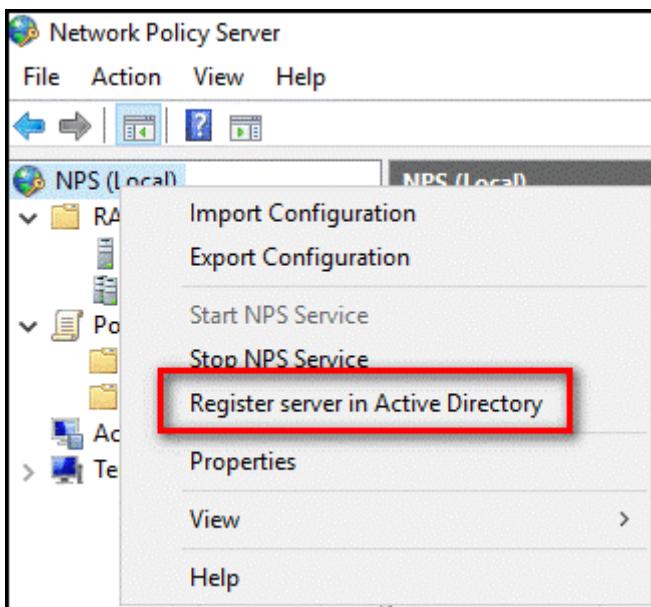
Configure NPS on the server where the NPS extension is installed

The NPS server where the NPS extension is installed needs to be able to exchange RADIUS messages with the NPS server on the Remote Desktop Gateway. To enable this message exchange, you need to configure the NPS components on the server where the NPS extension service is installed.

Register Server in Active Directory

To function properly in this scenario, the NPS server needs to be registered in Active Directory.

1. On the NPS server, open **Server Manager**.
2. In Server Manager, select **Tools**, and then select **Network Policy Server**.
3. In the Network Policy Server console, right-select **NPS (Local)**, and then select **Register server in Active Directory**.
4. Select **OK** two times.

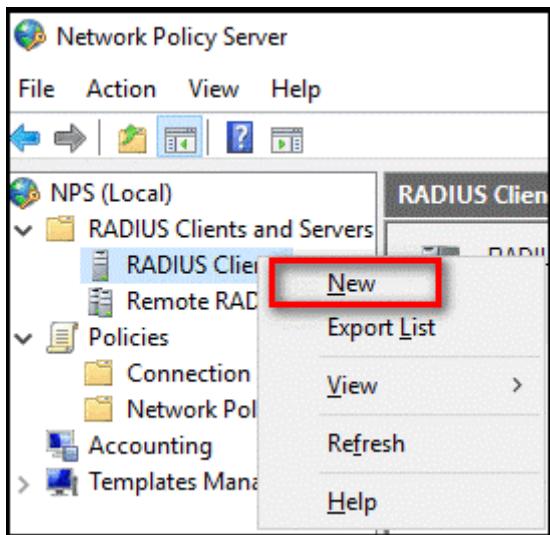


5. Leave the console open for the next procedure.

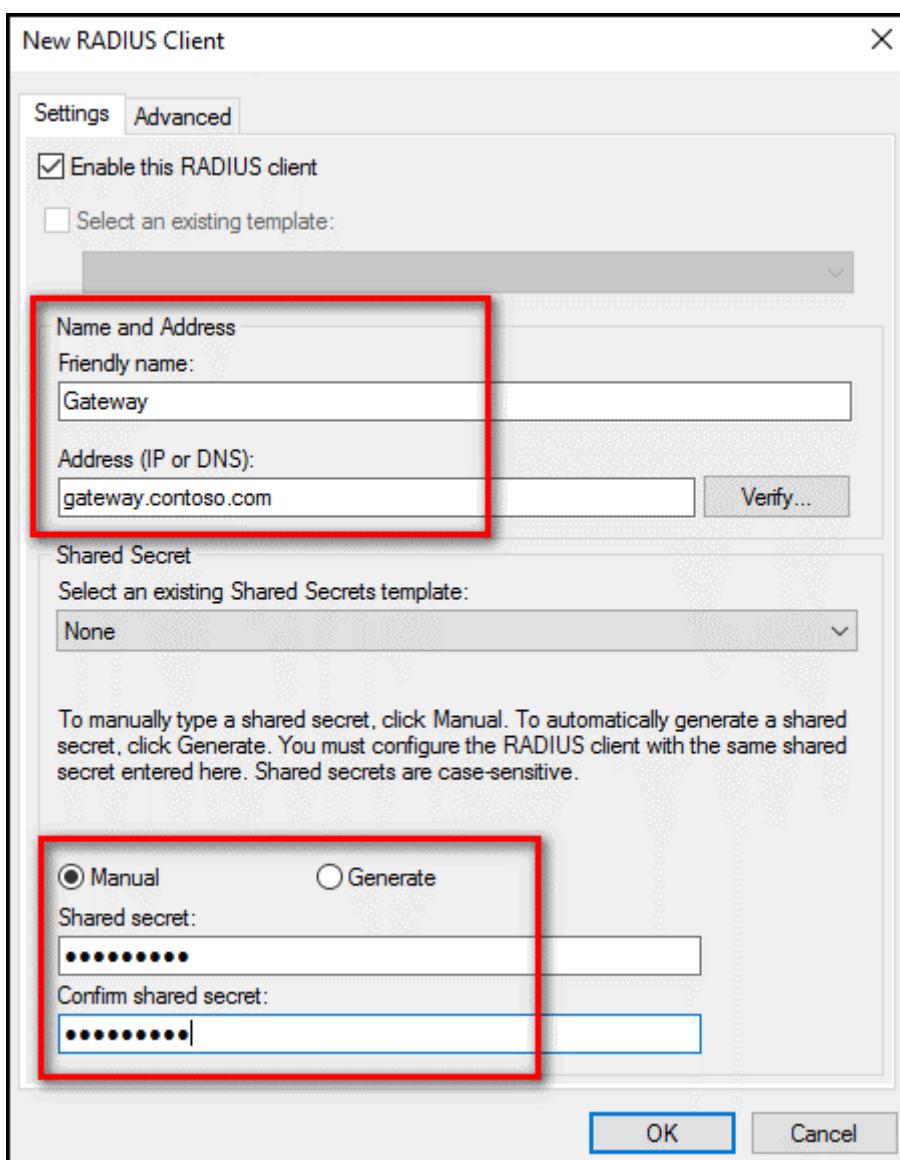
Create and configure RADIUS client

The Remote Desktop Gateway needs to be configured as a RADIUS client to the NPS server.

1. On the NPS server where the NPS extension is installed, in the **NPS (Local)** console, right-select **RADIUS Clients** and select **New**.



2. In the **New RADIUS Client** dialog box, provide a friendly name, such as *Gateway*, and the IP address or DNS name of the Remote Desktop Gateway server.
3. In the **Shared secret** and the **Confirm shared secret** fields, enter the same secret that you used before.

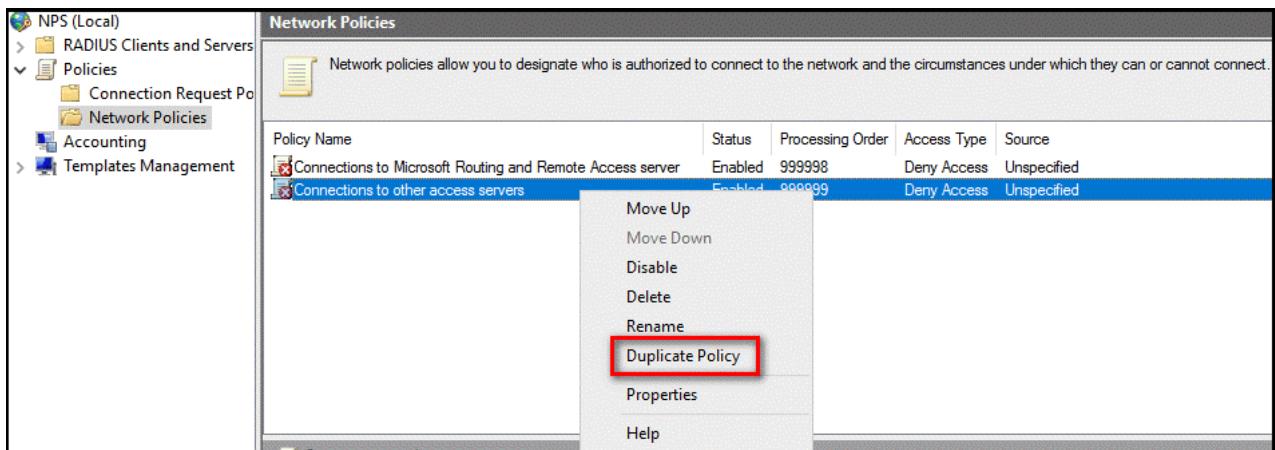


4. Select **OK** to close the New RADIUS Client dialog box.

Configure Network Policy

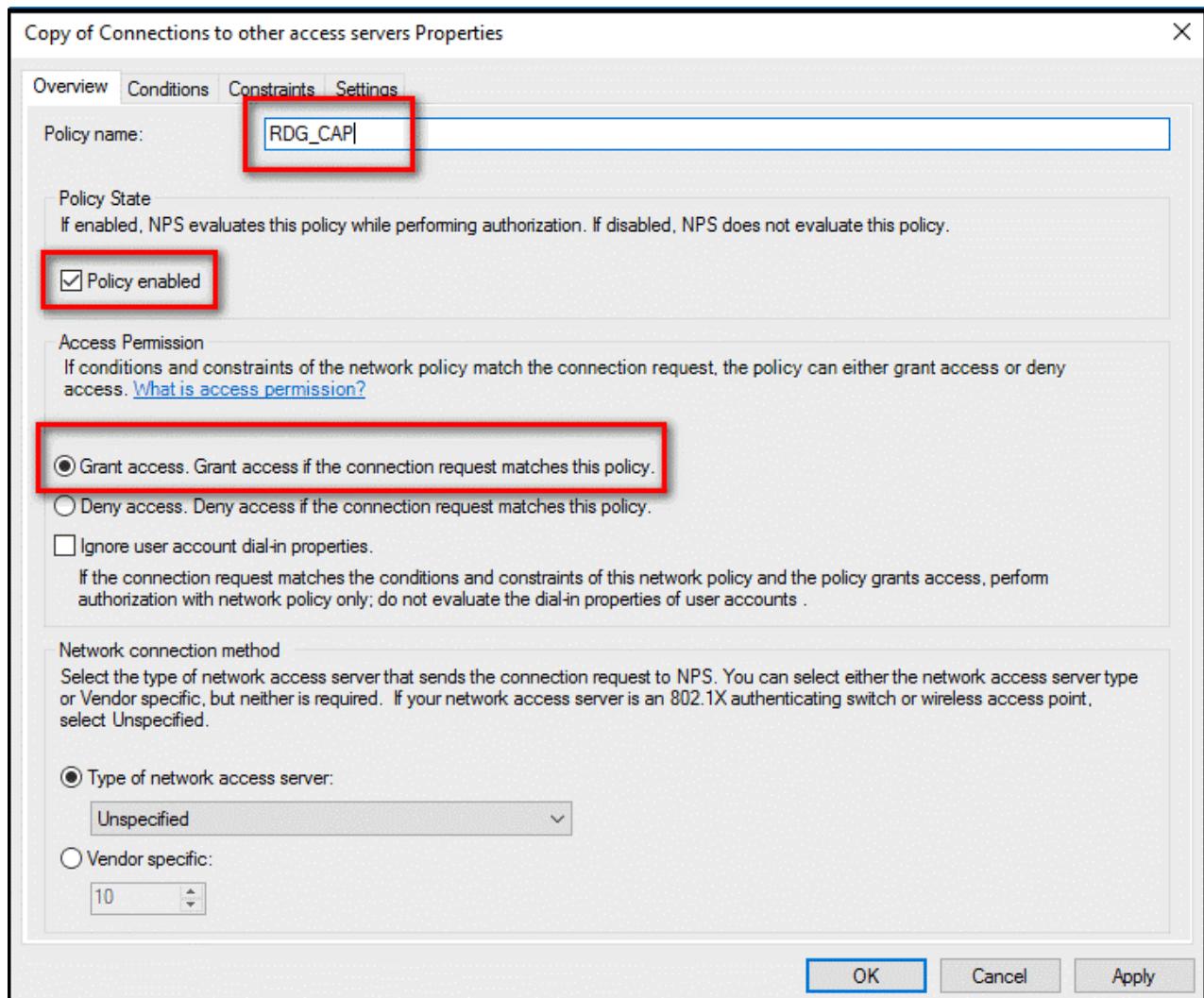
Recall that the NPS server with the Microsoft Entra multifactor authentication extension is the designated central policy store for the Connection Authorization Policy (CAP). Therefore, you need to implement a CAP on the NPS server to authorize valid connections requests.

1. On the NPS Server, open the NPS (Local) console, expand **Policies**, and select **Network Policies**.
2. Right-select **Connections to other access servers**, and select **Duplicate Policy**.

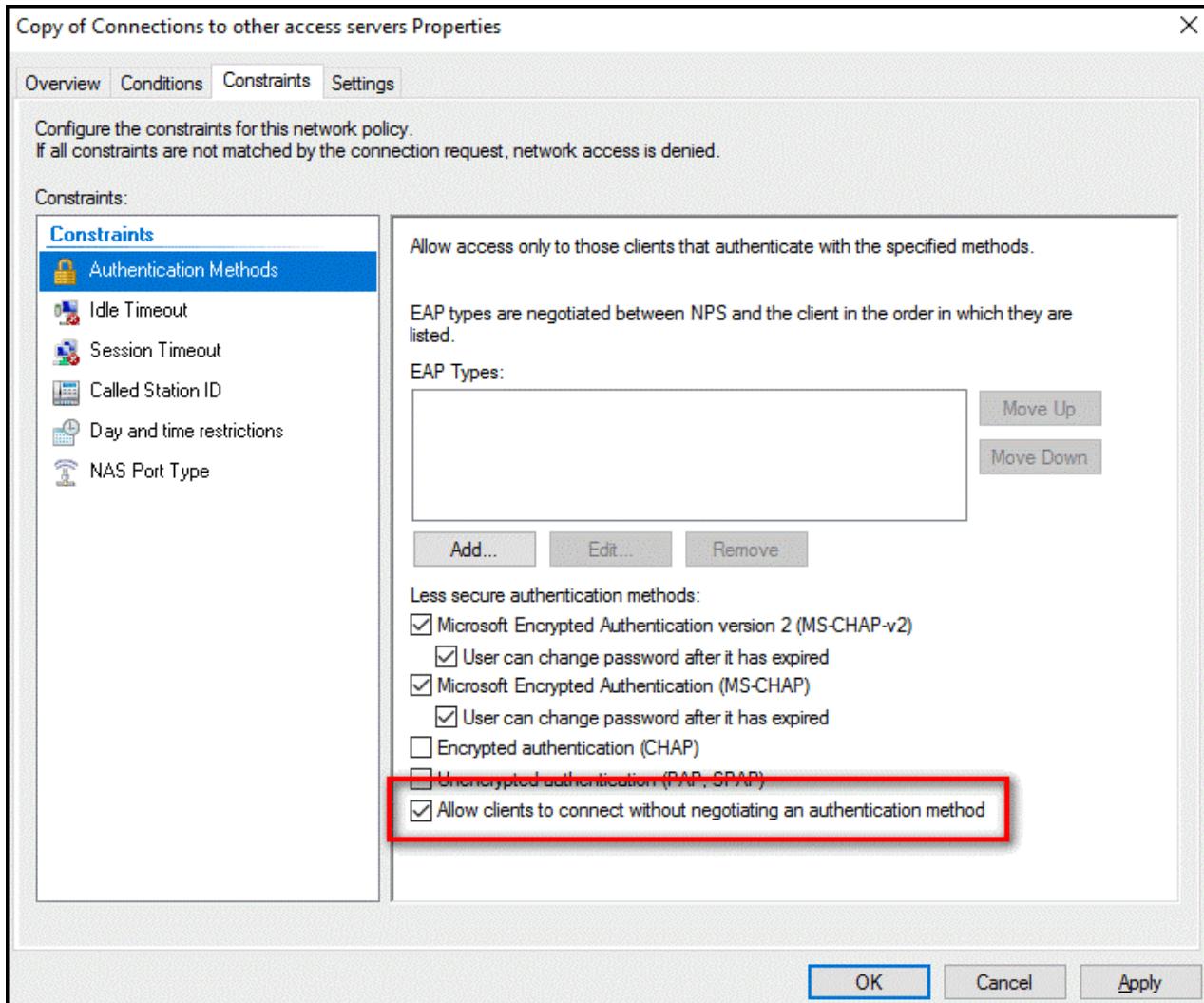


3. Right-select **Copy of Connections to other access servers**, and select **Properties**.

4. In the **Copy of Connections to other access servers** dialog box, in **Policy name**, enter a suitable name, such as *RDG_CAP*. Check **Policy enabled**, and select **Grant access**. Optionally, in **Type of network access server**, select **Remote Desktop Gateway**, or you can leave it as **Unspecified**.



5. Select the **Constraints** tab, and check **Allow clients to connect without negotiating an authentication method**.



6. Optionally, select the **Conditions** tab and add conditions that must be met for the connection to be authorized, for example, membership in a specific Windows group.

RDG_CAP Properties

Overview Conditions Constraints Settings

Configure the conditions for this network policy.

If conditions match the connection request, NPS uses this policy to authorize the connection request. If conditions do not match the connection request, NPS skips this policy and evaluates other policies, if additional policies are configured.

Condition	Value
Day and time restrictions	Sunday 00:00-24:00 Monday 00:00-24:00 Tuesday 00:00-24:00 Wednesday 00:00-24:00 Thursday 00:00-24:00 Friday 00:00-24:00 Saturday 00:00-24:00
Windows Groups	CONTOSO\RDGUsers

Condition description:
The Windows Groups condition specifies that the connecting user or computer must belong to one of the selected groups.

Add... Edit... Remove

OK Cancel Apply

This screenshot shows the 'RDG_CAP Properties' dialog box. The 'Conditions' tab is selected. It displays two conditions: 'Day and time restrictions' (set to Sunday through Saturday from 00:00 to 24:00) and 'Windows Groups' (set to the group 'CONTOSO\RDGUsers'). Below the conditions, a 'Condition description' section states that the Windows Groups condition specifies that the connecting user or computer must belong to one of the selected groups. At the bottom, there are buttons for 'Add...', 'Edit...', and 'Remove', and standard 'OK', 'Cancel', and 'Apply' buttons.

7. Select **OK**. When prompted to view the corresponding Help topic, select **No**.
8. Ensure that your new policy is at the top of the list, that the policy is enabled, and that it grants access.

Network Policies

Network policies allow you to designate who is authorized to connect to the network and the circumstances under which they can do so.

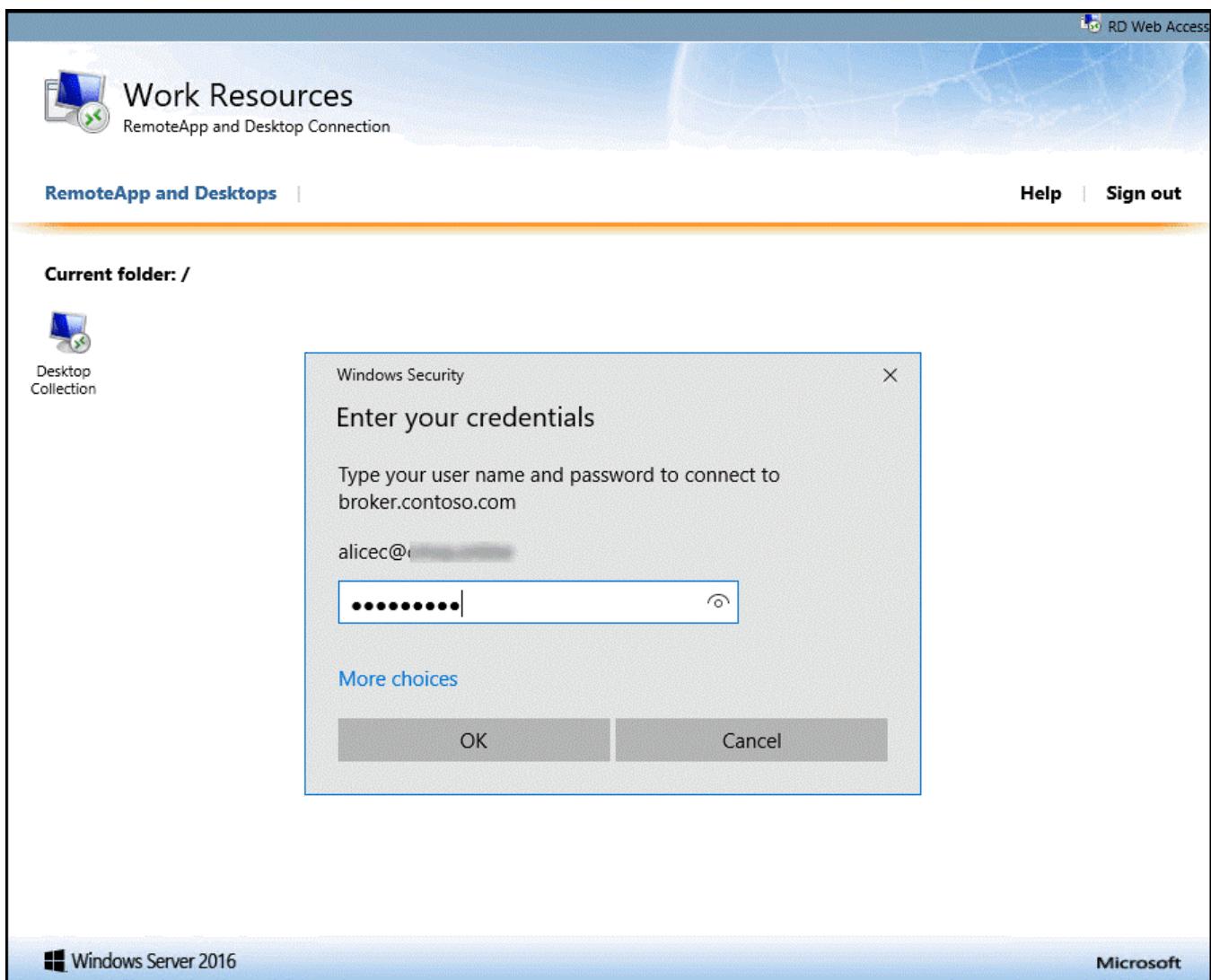
Policy Name	Status	Processing Order	Access Type	Source
RDG_CAP	Enabled	1	Grant Access	Unspecified
Connections to Microsoft Routing and Remote Access server	Enabled	2	Deny Access	Unspecified
Connections to other access servers	Enabled	3	Deny Access	Unspecified

This screenshot shows the 'Network Policies' window. It displays a table of policies. The first policy, 'RDG_CAP', is highlighted with a blue selection bar. The table columns are 'Policy Name', 'Status', 'Processing Order', 'Access Type', and 'Source'. The 'RDG_CAP' policy is enabled, has a processing order of 1, grants access, and has an unspecified source. The other two policies listed are 'Connections to Microsoft Routing and Remote Access server' and 'Connections to other access servers', both of which are also enabled and have processing orders of 2 and 3 respectively, with deny access types and unspecified sources.

Verify configuration

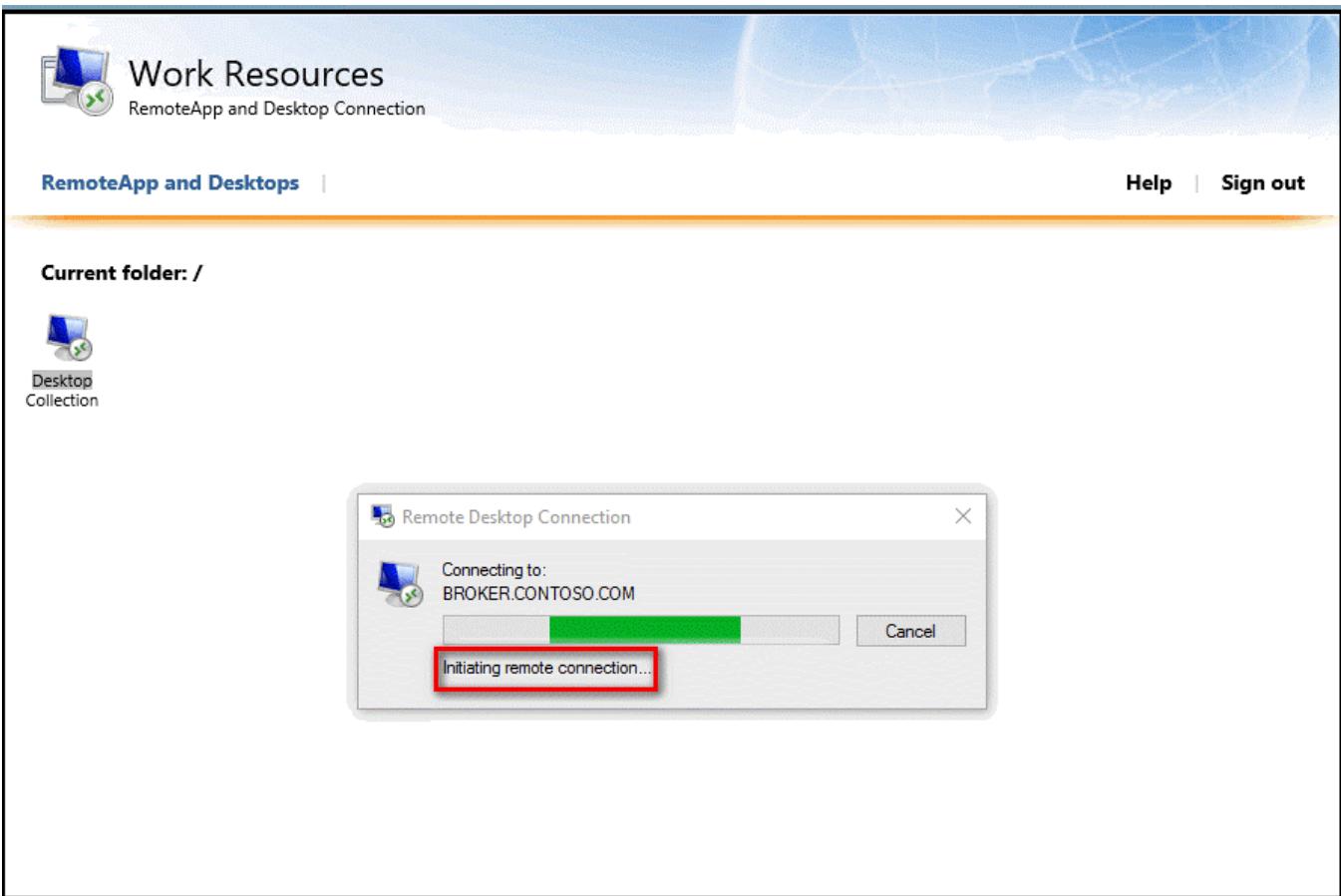
To verify the configuration, you need to sign in to the Remote Desktop Gateway with a suitable RDP client. Be sure to use an account that is allowed by your Connection Authorization Policies and is enabled for Microsoft Entra multifactor authentication.

As shown in the following image, you can use the Remote Desktop Web Access page.

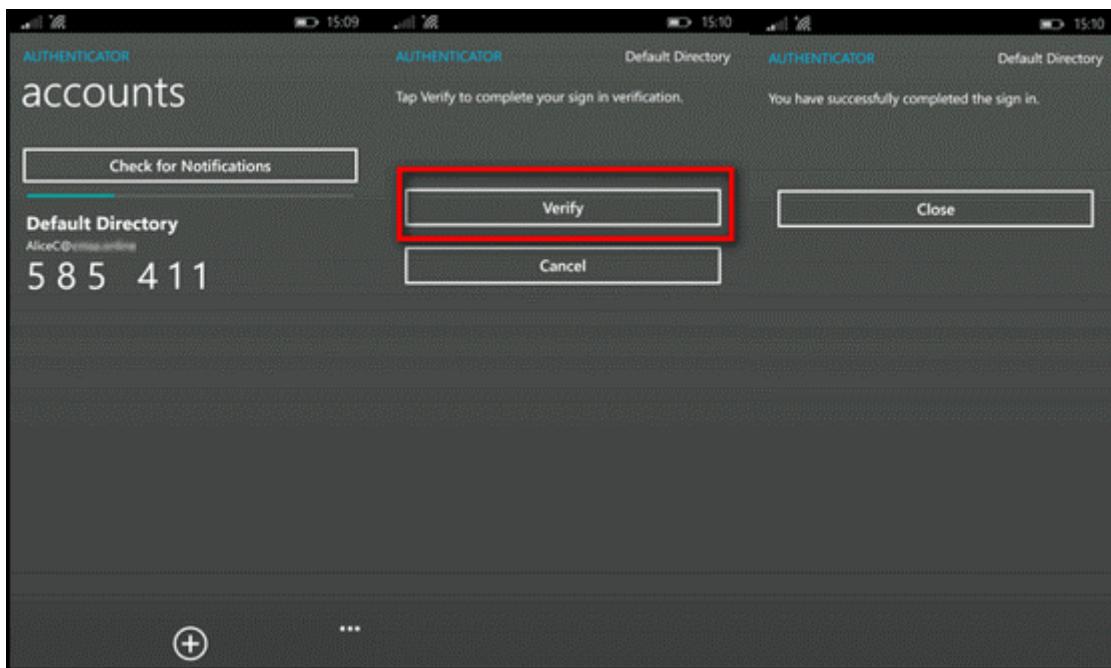


When you successfully enter your credentials for primary authentication, the Remote Desktop Connect dialog box shows a status of Initiating remote connection, as shown in the following section.

If you successfully authenticate with the secondary authentication method you previously configured in Microsoft Entra multifactor authentication, you're connected to the resource. However, if the secondary authentication isn't successful, you're denied access to the resource.



In the following example, the Authenticator app on a Windows phone is used to provide the secondary authentication.



Once you have successfully authenticated using the secondary authentication method, you're logged into the Remote Desktop Gateway as normal. However, because you're required to use a secondary authentication method using a mobile app on a trusted device, the sign in process is more secure than it would be otherwise.

View Event Viewer logs for successful logon events

To view the successful sign-in events in the Windows Event Viewer logs, you can issue the following PowerShell command to query the Windows Terminal Services and Windows Security logs.

To query successful sign-in events in the Gateway operational logs (*Event Viewer\Applications and Services Logs\Microsoft\Windows\TerminalServices-Gateway\Operational*), use the following PowerShell commands:

- `Get-WinEvent -Logname Microsoft-Windows-TerminalServices-Gateway/Operational | where {$_.ID -eq '300'} | FL`
- This command displays Windows events that show the user met resource authorization policy requirements (RD RAP) and was granted access.

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\vmadmin> Get-WinEvent -Logname Microsoft-Windows-TerminalServices-Gateway/Operational | where {$_.ID -eq '300'} | FL

TimeCreated : 6/6/2017 9:01:40 PM
ProviderName : Microsoft-Windows-TerminalServices-Gateway
Id : 300
Message : The user "CONTOSO\alicec", on client computer "104. [REDACTED]", met resource authorization policy requirements and was therefore authorized to connect to resource "rdsh-0.contoso.com".
```

- `Get-WinEvent -Logname Microsoft-Windows-TerminalServices-Gateway/Operational | where {$_.ID -eq '200'} | FL`
- This command displays the events that show when user met connection authorization policy requirements.

```
PS C:\Users\vmadmin> Get-WinEvent -Logname Microsoft-Windows-TerminalServices-Gateway/Operational | where {$_.ID -eq '200'} | FL

TimeCreated : 6/6/2017 9:01:40 PM
ProviderName : Microsoft-Windows-TerminalServices-Gateway
Id : 200
Message : The user "CONTOSO\alicec", on client computer "104. [REDACTED]", met connection authorization policy requirements and was therefore authorized to access the RD Gateway server. The authentication method used was: "NTLM" and connection protocol used: "HTTP".
```

You can also view this log and filter on event IDs, 300 and 200. To query successful logon events in the Security event viewer logs, use the following command:

- `Get-WinEvent -Logname Security | where {$_.ID -eq '6272'} | FL`
- This command can be run on either the central NPS or the RD Gateway Server.

```
PS C:\Users\vmadmin> Get-WinEvent -Logname Security | where {$_.ID -eq '6272'} | FL

TimeCreated : 6/6/2017 9:01:39 PM
ProviderName : Microsoft-Windows-Security-Auditing
Id : 6272
Message : Network Policy Server granted access to a user.

User:
Security ID:
Account Name:
Account Domain:
Fully Qualified Account Name:

Client Machine:
Security ID:
Account Name: WIN10-TEST
Fully Qualified Account Name: -
Called Station Identifier: UserAuthType:PW
Calling Station Identifier: -

NAS:
NAS IPv4 Address: -
NAS IPv6 Address: -
NAS Identifier: -
NAS Port-Type: Virtual
NAS Port: -

RADIUS Client:
Client Friendly Name: Gateway
Client IP Address:

Authentication Details:
Connection Request Policy Name: Use Windows authentication for all users
Network Policy Name: R
Authentication Provider: Windows
Authentication Server:
Authentication Type: Extension
EAP Type: -
Account Session Identifier: -
Logging Results: Accounting information was written to the local log file.
```

You can also view the Security log or the Network Policy and Access Services custom view:

The screenshot shows the Windows Event Viewer interface. On the left, the navigation pane displays categories like Event Viewer (Local), Custom Views, Server Roles, Administrative Events, Windows Logs, Applications and Services Logs, and Microsoft logs. The Microsoft section is expanded, showing sub-categories such as AppV, User Experience Virtualization, and Windows, which further expand into various application-specific logs.

The main pane displays a list of events under the title "Network Policy and Access Services Number of events: 241". The first few events are listed as Information level, occurring on 5/31/2017 at 5:10:00 PM. The event details for the first event are shown in a large window below:

Event 6272, Microsoft Windows security auditing.

General **Details**

Network Policy Server granted access to a user.

User:

- Security ID: NULL SID
- Account Name: CONTOSO\AliceC
- Account Domain: -
- Fully Qualified Account Name: -

Client Machine:

- Security ID: NULL SID
- Account Name: -
- Fully Qualified Account Name: -
- Called Station Identifier: UserAuthType:PW
- Calling Station Identifier: -

NAS:

- NAS IPv4 Address: -
- NAS IPv6 Address: -
- NAS Identifier: -
- NAS Port-Type: Virtual
- NAS Port: -

RADIUS Client:

- Client Friendly Name: -
- Client IP Address: -

Log Name: Security

Source: Microsoft Windows security **Logged:** 5/31/2017 5:05:52 PM

Event ID: **6272** **Task Category:** Network Policy Server

Level: Information **Keywords:** Audit Success

User: N/A **Computer:** gateway.contoso.com

OpCode: Info

On the server where you installed the NPS extension for Microsoft Entra multifactor authentication, you can find Event Viewer application logs specific to the extension at *Application and Services Logs\Microsoft\AzureMfa*.

The screenshot shows the Windows Event Viewer interface. On the left, the navigation pane lists various log categories under 'Event Viewer (Local)'. The 'Windows Logs' section is expanded, showing 'Application', 'Security', 'Setup', 'System', and 'Forwarded Events'. The 'Applications and Services Logs' section is also expanded, showing 'Active Directory Web Services', 'DFS Replication', 'Directory Service', 'DNS Server', 'Hardware Events', 'Internet Explorer', 'Key Management Service', 'Microsoft' (which is expanded to show 'AppV', 'AzureMfa', 'AuthN', 'AuthZ', 'AuthZAdminCh', and 'AuthZOptCh'), 'User Experience Virtualization', 'Windows', and 'WindowsAzure'. On the right, a table titled 'AuthZOptCh Number of events: /4' displays six log entries. The first five entries are 'Information' level events from 'AuthZ' source, dated 5/31/2017 at various times between 5:17:05 PM and 5:18:44 PM. The sixth entry is partially visible. Below the table, a detailed view of the second event ('Event Properties - Event 2, AuthZ') is shown. The 'General' tab is selected, displaying the message 'NPS Extension for Azure MFA: CID: user with Azure MFA response Success message'. The 'Details' tab shows log properties: Log Name: Microsoft-, Source: AuthZ, Event ID: 2, Level: Information, User: NETWORK SERVICE, OpCode: Info, and Task Category: None. A note indicates 'Access Accepted for session'. At the bottom of the details window are 'Copy' and 'Close' buttons.

Level	Date and Time	Source	Event ID	Task Category
Information	5/31/2017 5:17:16 PM	AuthZ	1	None
Information	5/31/2017 5:17:05 PM	AuthZ	1	None
Information	5/31/2017 5:17:05 PM	AuthZ	2	None
Information	5/31/2017 5:16:44 PM	AuthZ	2	None
Information	5/31/2017 5:15:30 PM	AuthZ	2	None
Information	5/31/2017 5:13:29 PM	AuthZ	2	None
	5/31/2017 5:06:02 PM	AuthZ	1	None

Troubleshoot Guide

If the configuration isn't working as expected, the first place to start to troubleshoot is to verify that the user is configured to use Microsoft Entra multifactor authentication. Have the user sign in to the [Microsoft Entra admin center](#). If users are prompted for secondary verification and can successfully authenticate, you can eliminate an incorrect configuration of Microsoft Entra multifactor authentication.

If Microsoft Entra multifactor authentication is working for the user(s), you should review the relevant Event logs. These include the Security Event, Gateway operational, and Microsoft Entra multifactor authentication logs that are discussed in the previous section.

See the following example output of Security log showing a failed logon event (Event ID 6273).

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\admin > Get-WinEvent -Logname Security | where {$_.ID -eq '6273'} | FL

TimeCreated : 5/31/2017 5:16:44 PM
ProviderName : Microsoft-Windows-Security-Auditing
Id : 6273
Message : Network Policy Server denied access to a user.

    Contact the Network Policy Server administrator for more information.

User:
    Security ID:
    Account Name:
    Account Domain:
    Fully Qualified Account Name:   -

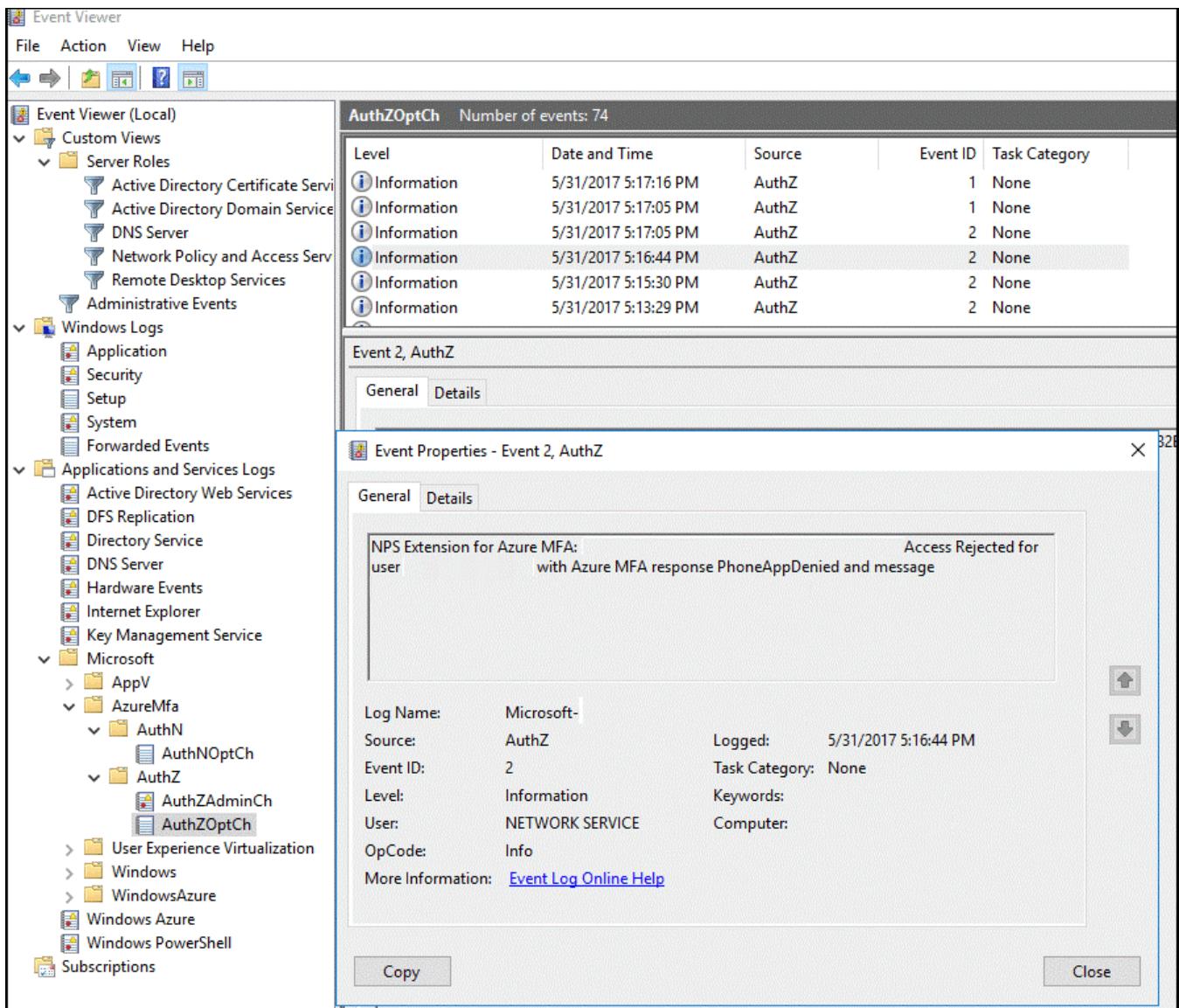
Client Machine:
    Security ID:
    Account Name:
    Fully Qualified Account Name:   -
    Called Station Identifier:      UserAuthType:
    Calling Station Identifier:     -

NAS:
    NAS IPv4 Address:           -
    NAS IPv6 Address:           -
    NAS Identifier:             -
    NAS Port-Type:              Virtual
    NAS Port:                  -

RADIUS Client:
    Client Friendly Name:       Gateway
    Client IP Address:         

Authentication Details:
    Connection Request Policy Name: Forward messages to RDGW
    Network Policy Name:          -
    Authentication Provider:       RADIUS Proxy
    Authentication Server:        -
    Authentication Type:          Extension
    EAP Type:                   -
    Account Session Identifier:   -
    Logging Results:              Accounting information was written to the local log file.
    Reason Code:                 21
    Reason:                      An NPS extension dynamic link library (DLL) that is installed on the NPS
server rejected
the connection request.
```

What follows is a related event from the AzureMFA logs:



To perform advanced troubleshoot options, consult the NPS database format log files where the NPS service is installed. These log files are created in %SystemRoot%\System32\Logs folder as comma-delimited text files.

For a description of these log files, see [Interpret NPS Database Format Log Files](#). The entries in these log files can be difficult to interpret without importing them into a spreadsheet or a database. You can find several IAS parsers online to assist you in interpreting the log files.

The following image shows the output of one such downloadable [shareware application](#).

IAS Log Viewer Trial v3.16								
File Edit View Reports Tools Help								
Records	Connects	Alerts	Scheduled Tasks					
Start Date/Time	User Name	Stop Date/Time	Duration	User IP	Output Octets	Input Octets	Connect Request	Connect Re...
05/21/2017 20:38:56	CONTOSO\AliceC	05/21/2017 20:48:11	00:09:15		827,926	115,805	The request was discarded by a third-party ext...	Finished
05/21/2017 20:39:04		05/21/2017 20:48:11	00:09:07		121,902	58,804	The request was discarded by a third-party ext...	Finished
05/21/2017 21:06:50	CONTOSO\AliceC	05/21/2017 21:06:50	00:00:00					
05/21/2017 21:07:51	CONTOSO\AliceC	05/25/2017 22:54:22	4 days 0...		1,119,524	160,95	Properties	x
05/21/2017 21:08:00		05/21/2017 21:10:09	00:02:09		95,065	127,62		
05/21/2017 21:10:16	CONTOSO\AliceC	05/21/2017 21:13:53	00:03:37		90,661	69,37		
05/21/2017 21:10:51		05/21/2017 21:14:53	00:04:02		392,679	146,35		
05/21/2017 21:14:02	CONTOSO\AliceC	05/21/2017 21:15:53	00:01:51		90,661	69,37	Name	Value
05/21/2017 21:15:18	CONTOSO\AliceC	05/21/2017 21:21:30	00:06:12		10,157	5,94	Called Station Id	UserAuthType:Pw
05/22/2017 21:45:26	CONTOSO\AliceC	05/22/2017 21:45:39	00:00:13		5,227	5,74	Client Friendly Name	Gateway
05/22/2017 21:46:14	CONTOSO\AliceC	05/22/2017 21:46:24	00:00:10		5,227	5,74	Client IP Address	10.0.0.6
05/22/2017 21:46:26		05/22/2017 21:47:21	00:00:55		5,227	5,74	Connect Request	An IAS extension dynamic link library (DLL) that is installed on the NPS server rejected the connection request.
05/22/2017 21:47:22	CONTOSO\AliceC	05/22/2017 21:48:34	00:01:12		5,227	5,74	Connect Result	Rejected
05/22/2017 21:47:26		05/22/2017 21:53:42	00:06:16		4,953	5,74	Duration	00:00:00
05/22/2017 21:50:29	CONTOSO\AliceC	05/22/2017 21:50:42	00:00:13		4,953	5,74	Record Count	2
05/22/2017 21:54:26	CONTOSO\AliceC	05/22/2017 21:54:36	00:00:10		5,039	5,74	Server Name	DC1
05/22/2017 21:55:26		05/22/2017 21:57:36	00:02:10		5,039	5,74	Server NasPort	0
05/22/2017 22:01:10	CONTOSO\AliceC	05/22/2017 22:24:42	00:23:32		210,220	69,10	Session Time	0
05/22/2017 22:28:35	CONTOSO\AliceC	05/22/2017 22:58:19	00:29:44		267,150	86,50	Start DateTime	05/31/2017 17:16:44
05/22/2017 22:29:43		05/31/2017 16:20:47	8 days 1...		5,227	5,74	Stop DateTime	05/31/2017 17:16:44
05/22/2017 23:03:54	CONTOSO\AliceC	05/22/2017 23:29:48	00:25:54		237,649	39,54	Terminate Cause	An IAS extension dynamic link library (DLL) that is installed on the NPS server rejected the connection request.
05/25/2017 20:23:08	CONTOSO\AliceC	05/25/2017 22:55:23	02:32:15		1,146,503	63,47	User Name	CONTOSO\AliceC
05/31/2017 15:53:55	CONTOSO\AliceC	05/31/2017 15:53:55	00:00:00				Start Date	05/31/2017
05/31/2017 16:17:35	CONTOSO\AliceC	05/31/2017 17:17:16	00:59:41		1,474,497	132,07	Start Time	17:16:44
05/31/2017 17:05:51	CONTOSO\AliceC	05/31/2017 17:13:23	00:07:32		115,367	107,49		
05/31/2017 17:06:59		05/31/2017 17:14:23	00:07:24		115,367	107,480	The request was discarded by a third-party ext...	Finished
05/31/2017 17:13:29	CONTOSO\AliceC	05/31/2017 17:13:29	00:00:00				An IAS extension dynamic link library (DLL) th...	Rejected
05/31/2017 17:15:30	CONTOSO\AliceC	05/31/2017 17:15:30	00:00:00				An IAS extension dynamic link library (DLL) th...	Rejected
05/31/2017 17:16:44	CONTOSO\AliceC	05/31/2017 17:16:44	00:00:00				An IAS extension dynamic link library (DLL) th...	Rejected
05/31/2017 17:17:05	CONTOSO\AliceC	05/31/2017 17:17:12	00:00:07		10,157	5,941	The request was discarded by a third-party ext...	Online
05/31/2017 17:17:16		05/31/2017 17:20:16	00:03:00		10,157	5,941	The request was discarded by a third-party ext...	Online

Next steps

How to get Microsoft Entra multifactor authentication

Remote Desktop Gateway and Azure Multi-Factor Authentication Server using RADIUS

Integrate your on-premises directories with Microsoft Entra ID

Integrate your VPN infrastructure with Microsoft Entra multifactor authentication by using the Network Policy Server extension for Azure

Article • 03/04/2025

The Network Policy Server (NPS) extension for Azure allows organizations to safeguard Remote Authentication Dial-In User Service (RADIUS) client authentication using cloud-based [Microsoft Entra multifactor authentication](#), which provides two-step verification.

This article provides instructions for integrating NPS infrastructure with MFA by using the NPS extension for Azure. This process enables secure two-step verification for users who attempt to connect to your network by using a VPN.

ⓘ Note

Although the NPS MFA extension supports time-based one-time password (TOTP), certain VPN clients like Windows VPN don't. Make sure the VPN clients that you're using support TOTP as an authentication method before you enable it in the NPS extension.

Network Policy and Access Services gives organizations the ability to:

- Assign a central location for the management and control of network requests to specify:
 - Who can connect
 - What times of day connections are allowed
 - The duration of connections
 - The level of security that clients must use to connect

Rather than specify policies on each VPN or Remote Desktop Gateway server, do so after they're in a central location. The RADIUS protocol is used to provide centralized Authentication, Authorization, and Accounting (AAA).

- Establish and enforce Network Access Protection (NAP) client health policies that determine whether devices are granted unrestricted or restricted access to network

resources.

- Provide a way to enforce authentication and authorization for access to 802.1x-capable wireless access points and Ethernet switches. For more information, see [Network Policy Server](#).

To enhance security and provide a high level of compliance, organizations can integrate NPS with Microsoft Entra multifactor authentication to ensure that users use two-step verification to connect to the virtual port on the VPN server. For users to be granted access, they must provide their username and password combination and other information that they control. This information must be trusted and not easily duplicated. It can include a cell phone number, a landline number, or an application on a mobile device.

If your organization uses a VPN and the user is registered for a TOTP code along with Authenticator push notifications, the user can't meet the MFA challenge and the remote sign-in fails. In that case, you can set OVERRIDE_NUMBER_MATCHING_WITH OTP = FALSE to fallback to push notifications to Approve/Deny with Authenticator.

In order for an NPS extension to continue working for VPN users, this registry key must be created on the NPS server. On the NPS server, open the registry editor. Navigate to:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\AzureMfa

Create the following String/Value pair:

Name: OVERRIDE_NUMBER_MATCHING_WITH OTP

Value = FALSE

Prior to the availability of the NPS extension for Azure, customers who wanted to implement two-step verification for integrated NPS and MFA environments had to configure and maintain a separate MFA server in an on-premises environment. Remote Desktop Gateway and Azure Multi-Factor Authentication Server offer this type of authentication using RADIUS.

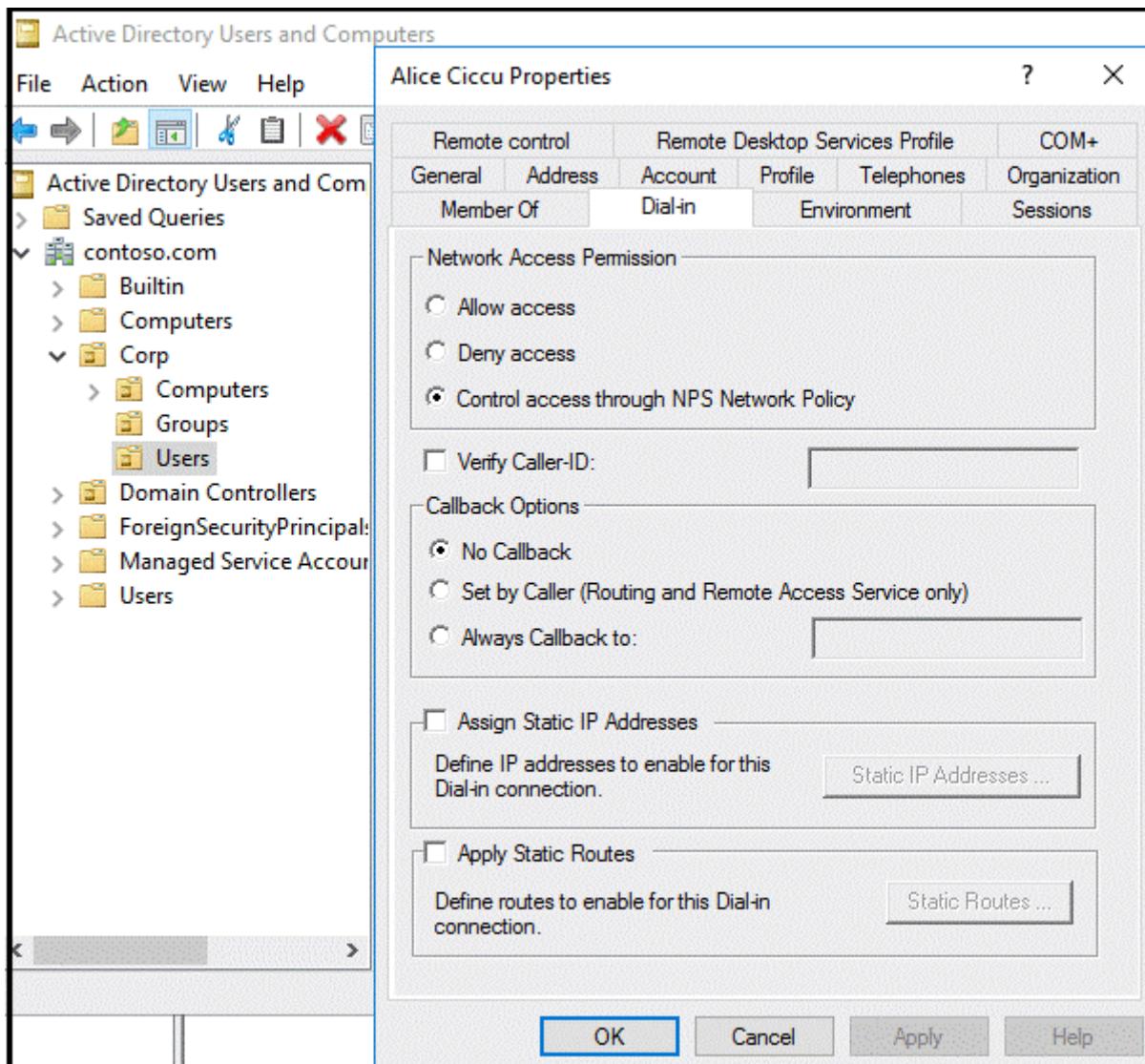
With the NPS extension for Azure, organizations can secure RADIUS client authentication by deploying either an on-premises based MFA solution or a cloud-based MFA solution.

Authentication flow

When users connect to a virtual port on a VPN server, they must first authenticate by using a variety of protocols. The protocols allow the use of a combination of user name

and password and certificate-based authentication methods.

In addition to authenticating and verifying their identity, users must have the appropriate dial-in permissions. In simple implementations, dial-in permissions that allow access are set directly on the Active Directory user objects.



In simple implementations, each VPN server grants or denies access based on policies that are defined on each local VPN server.

In larger and more scalable implementations, the policies that grant or deny VPN access are centralized on RADIUS servers. In these cases, the VPN server acts as an access server (RADIUS client) that forwards connection requests and account messages to a RADIUS server. To connect to the virtual port on the VPN server, users must be authenticated and meet the conditions that are defined centrally on RADIUS servers.

When the NPS extension for Azure is integrated with the NPS, a successful authentication flow results, as follows:

1. The VPN server receives an authentication request from a VPN user that includes the username and password for connecting to a resource, such as a Remote

Desktop session.

2. Acting as a RADIUS client, the VPN server converts the request to a RADIUS *Access-Request* message and sends it (with an encrypted password) to the RADIUS server where the NPS extension is installed.
3. The username and password combination is verified in Active Directory. If either the username or password is incorrect, the RADIUS Server sends an *Access-Reject* message.
4. If the conditions in the NPS Connection Request and Network Policies are met (like time of day or group membership restrictions), the NPS extension will request secondary authentication with Microsoft Entra multifactor authentication.
5. Microsoft Entra multifactor authentication communicates with Microsoft Entra ID, retrieves the user's details, and uses the user configured method (cell phone call, text message, or mobile app) to perform the secondary authentication.
6. When the MFA challenge is successful, Microsoft Entra multifactor authentication communicates the result to the NPS extension.
7. After the connection attempt is both authenticated and authorized, the NPS where the extension is installed sends a RADIUS *Access-Accept* message to the VPN server (RADIUS client).
8. The user is granted access to the virtual port on the VPN server and establishes an encrypted VPN tunnel.

Prerequisites

This section details the prerequisites that must be completed before you can integrate MFA with the VPN. Before you begin, you must have the following prerequisites in place:

- VPN infrastructure
- Network Policy and Access Services role
- Microsoft Entra multifactor authentication license
- Windows Server software
- Libraries
- Microsoft Entra ID synced with on-premises Active Directory
- Microsoft Entra GUID ID

VPN infrastructure

This article assumes that you have a working VPN infrastructure that uses Microsoft Windows Server 2016 and that your VPN server is currently not configured to forward connection requests to a RADIUS server. In the article, you configure the VPN infrastructure to use a central RADIUS server.

If you don't have a working VPN infrastructure in place, you can quickly create one by following the guidance in numerous VPN setup tutorials that you can find on the Microsoft and third-party sites.

The Network Policy and Access Services role

Network Policy and Access Services provides the RADIUS server and client functionality. This article assumes that you have installed the Network Policy and Access Services role on a member server or domain controller in your environment. In this guide, you configure RADIUS for a VPN configuration. Install the Network Policy and Access Services role on a server *other than* your VPN server.

For information about installing the Network Policy and Access Services role service Windows Server 2012 or later, see [Install a NAP Health Policy Server](#). NAP is deprecated in Windows Server 2016. For a description of best practices for NPS, including the recommendation to install NPS on a domain controller, see [Best practices for NPS](#).

Windows Server software

The NPS extension requires Windows Server 2008 R2 SP1 or later, with the Network Policy and Access Services role installed. All the steps in this guide were performed with Windows Server 2016.

Libraries

The following library is installed automatically with the NPS extension:

- [Visual C++ Redistributable Packages for Visual Studio 2013 \(X64\)](#) ↗

If Microsoft Graph PowerShell module isn't already present, it's installed with a configuration script that you run as part of the setup process. There's no need to install Graph PowerShell in advance.

Microsoft Entra ID synced with on-premises Active Directory

To use the NPS extension, on-premises users must be synced with Microsoft Entra ID and enabled for MFA. This guide assumes that on-premises users are synced with Microsoft Entra ID via Microsoft Entra Connect. Instructions for enabling users for MFA are provided in the following section.

For information about Microsoft Entra Connect, see [Integrate your on-premises directories with Microsoft Entra ID](#).

Microsoft Entra GUID ID

To install the NPS extension, you need to know the GUID of the Microsoft Entra ID. Instructions for finding the GUID of the Microsoft Entra ID are provided in the next section.

Configure RADIUS for VPN connections

If you installed the NPS role on a member server, you need to configure it to authenticate and authorize the VPN client that requests VPN connections.

This section assumes that you installed the Network Policy and Access Services role but haven't configured it for use in your infrastructure.

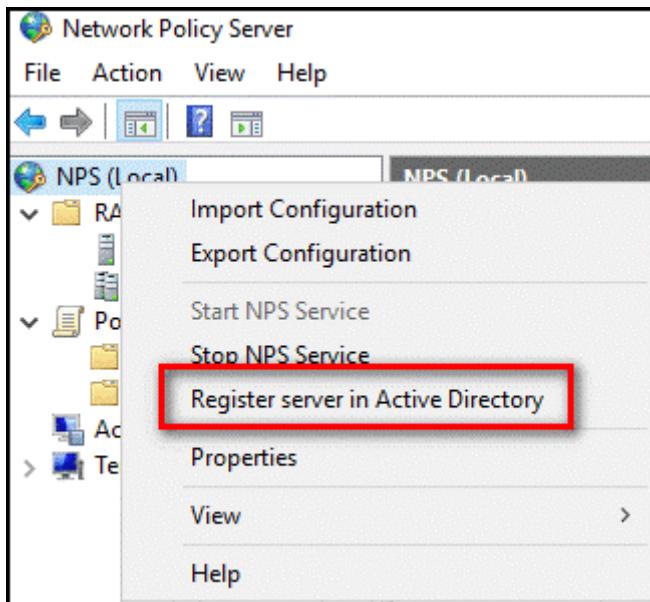
 **Note**

If you already have a working VPN server that uses a centralized RADIUS server for authentication, you can skip this section.

Register Server in Active Directory

To function properly in this scenario, the NPS server must be registered in Active Directory.

1. Open Server Manager.
2. In Server Manager, select **Tools**, and then select **Network Policy Server**.
3. In the Network Policy Server console, right-click **NPS (Local)**, and then select **Register server in Active Directory**. Select **OK** two times.

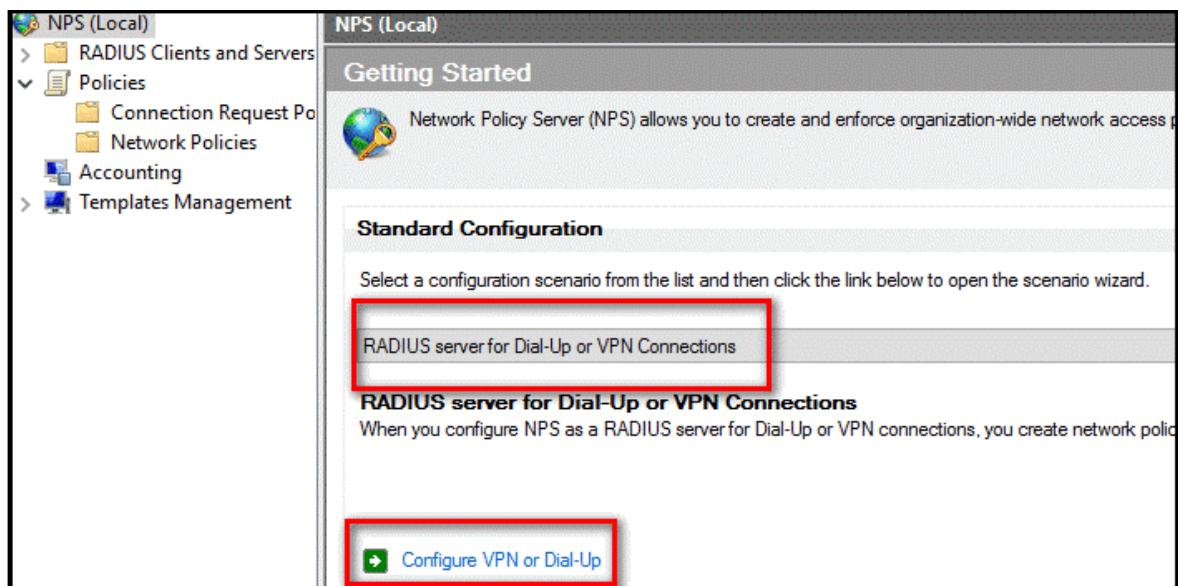


4. Leave the console open for the next procedure.

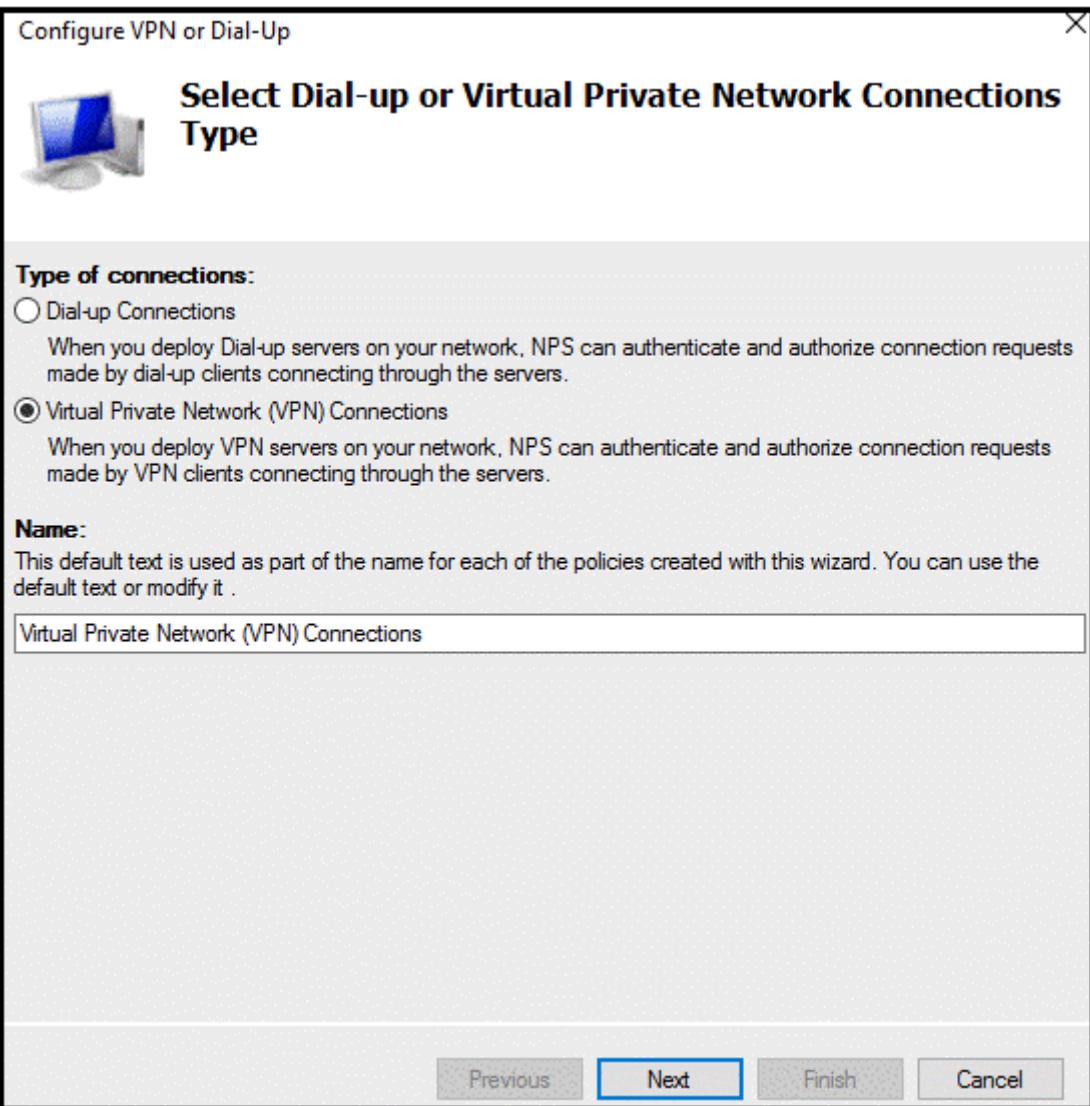
Use wizard to configure the RADIUS server

You can use a standard (wizard-based) or advanced configuration option to configure the RADIUS server. This section assumes that you're using the wizard-based standard configuration option.

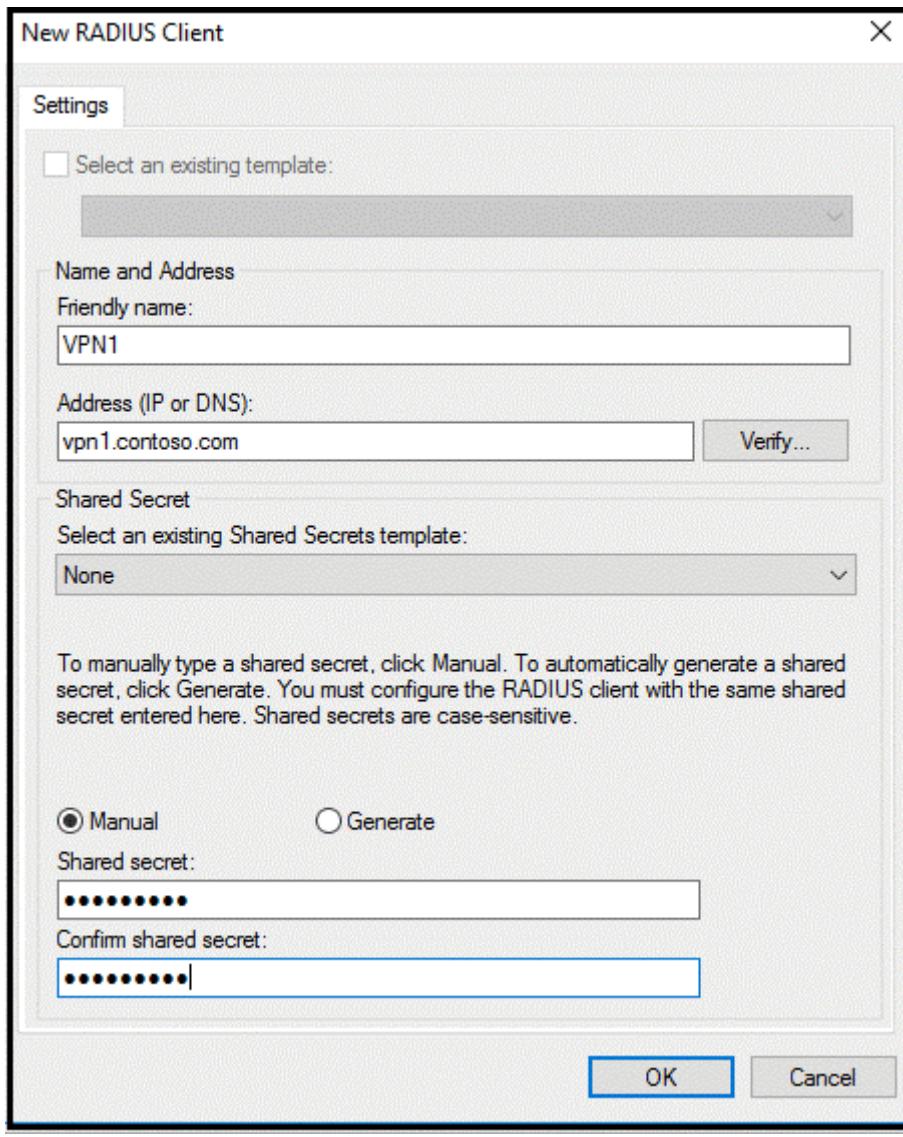
1. In the Network Policy Server console, select **NPS (Local)**.
2. Under **Standard Configuration**, select **RADIUS Server for Dial-Up or VPN Connections**, and then select **Configure VPN or Dial-Up**.



3. In the **Select Dial-up or Virtual Private Network Connections Type** window, select **Virtual Private Network Connections**, and then select **Next**.



4. In the Specify Dial-Up or VPN Server window, select Add.
5. In the New RADIUS client window, provide a friendly name, enter the resolvable name or IP address of the VPN server, and then enter a shared secret password. Make the shared secret password long and complex. Record it, because you'll need it in the next section.



6. Select **OK**, and then select **Next**.
7. In the **Configure Authentication Methods** window, accept the default selection (**Microsoft Encrypted Authentication version 2 [MS-CHAPv2]**) or choose another option, and select **Next**.

! Note

If you configure Extensible Authentication Protocol (EAP), you must use either Microsoft Challenge-Handshake Authentication Protocol (CHAPv2) or Protected Extensible Authentication Protocol (PEAP). No other EAP is supported.

8. In the **Specify User Groups** window, select **Add**, and then select an appropriate group. If no group exists, leave the selection blank to grant access to all users.

Specify User Groups



Users that are members of the selected group or groups will be allowed or denied access based on the network policy Access Permission setting.

To select User Groups, click Add. If no groups are selected, this policy applies to all users.

Groups

CONTOSO\VPNUsers

Add...

Remove

Previous

Next

Finish

Cancel

9. Select Next.

10. In the Specify IP Filters window, select Next.

11. In the Specify Encryption Settings window, accept the default settings, and then select Next.



Specify Encryption Settings

Specify the allowed encryption strengths used for traffic between access clients and the network access server.

If you are using Routing and Remote Access Service configured as a dial-up or VPN server, you can configure encryption strength.

The encryption settings are supported by computers running Microsoft Routing and Remote Access Service.

If you use different network access servers for dial-up or VPN connections, ensure that the encryptions settings you select are supported by your servers.

If No encryption is the only option selected, traffic from access clients to the network access server is not secured by encryption. This configuration is not recommended.

- Basic encryption (MPPE 40-bit)
- Strong encryption (MPPE 56-bit)
- Strongest encryption (MPPE 128-bit)

Previous

Next

Finish

Cancel

12. In the **Specify a Realm Name** window, leave the realm name blank, accept the default setting, and then select **Next**.



Specify a Realm Name

If you specify a realm name, the user account location supplied by users in log on credentials, such as a domain name, is replaced by the value you choose.

Your ISP uses a portion of the user name to identify which connection requests to route to this server. This part of the user name is the realm name.

If you do not know your realm name, contact your ISP. If you do not care about realm name, please click next.

Type the realm name, including the separator character (the period or the forward slash), that your ISP uses to forward requests.

Realm name:

Example: ISP.

Before authentication, remove the realm name from the user name

If the realm name is an identifier added to the existing Windows user name, it must be removed before Windows can authenticate the connection request.

Previous

Next

Finish

Cancel

13. In the Completing New Dial-up or Virtual Private Network Connections and RADIUS clients window, select Finish.



Completing New Dial-up or Virtual Private Network Connections and RADIUS clients

You have successfully created the following policies and configured the following RADIUS clients.

- To view the configuration details in your default browser, click Configuration Details.
- To change the configuration, click Previous.
- To save the configuration and close this wizard, click Finish.

RADIUS clients:

VPN1 (vpn1.contoso.com)

Connection Request Policy:

Virtual Private Network (VPN) Connections

Network Policies:

Virtual Private Network (VPN) Connections

[Configuration Details](#)

Previous

Next

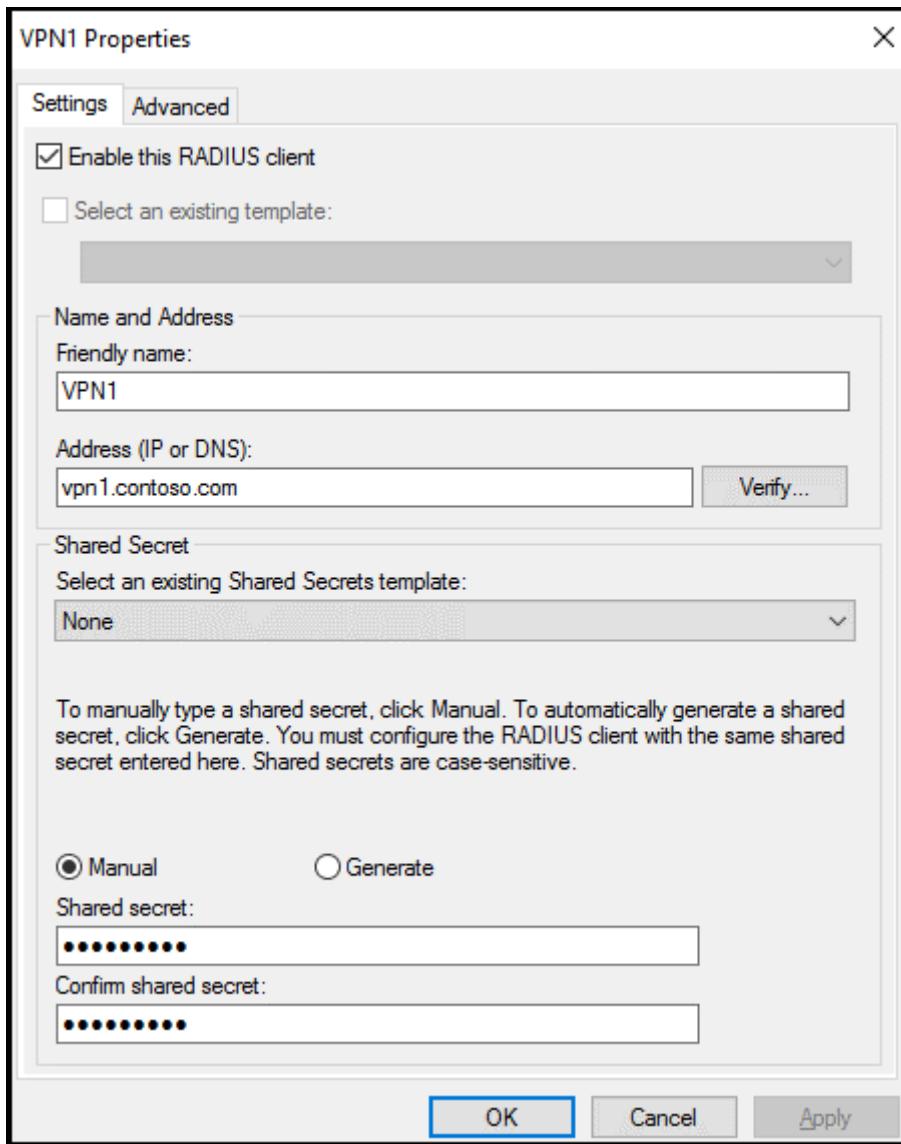
Finish

Cancel

Verify the RADIUS configuration

This section details the configuration you created by using the wizard.

1. On the Network Policy Server, in the NPS (local) console, expand **RADIUS Clients**, and then select **RADIUS Clients**.
2. In the details pane, right-click the RADIUS client that you created, and then select **Properties**. The properties of your RADIUS client (the VPN server) should be like those shown here:



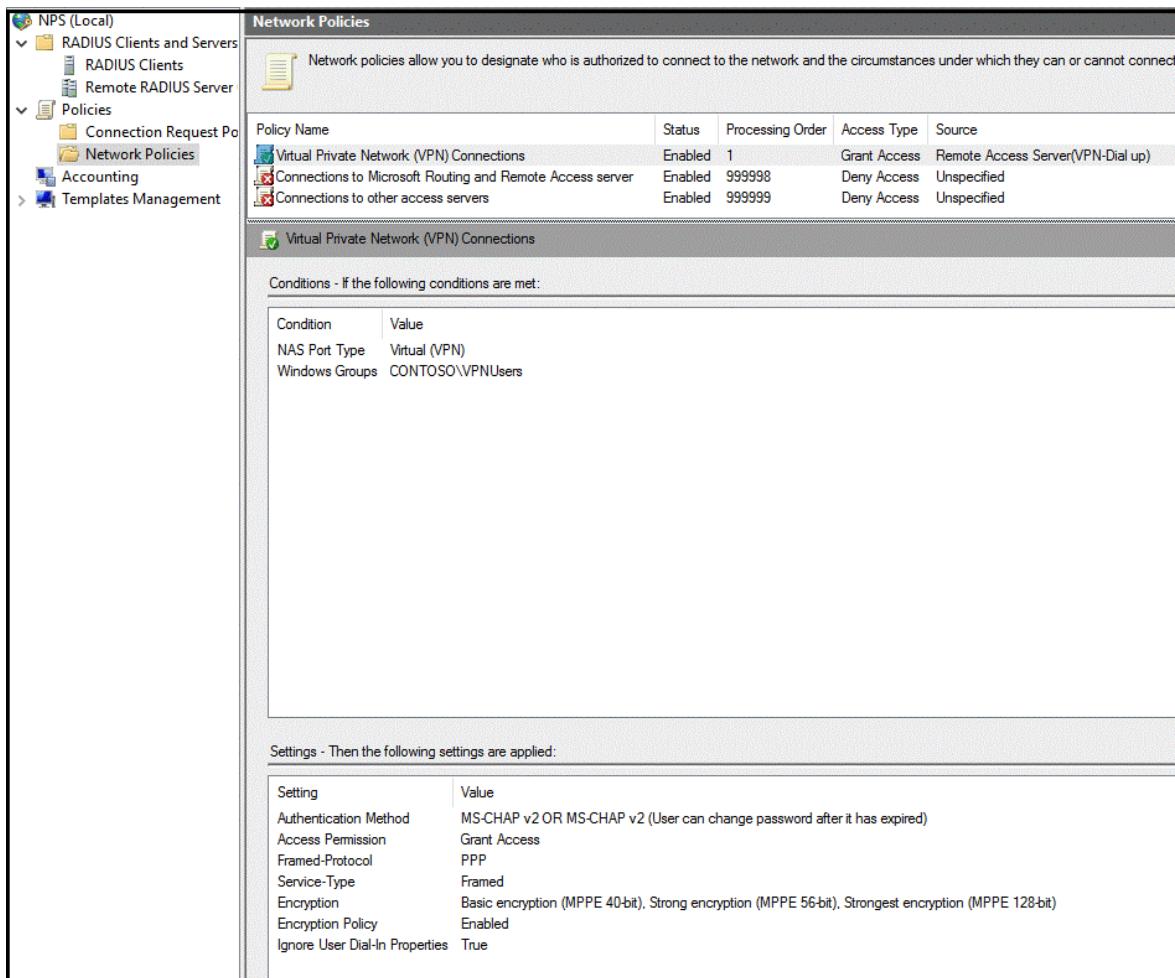
3. Select Cancel.

4. On the Network Policy Server, in the NPS (local) console, expand **Policies**, and then select **Connection Request Policies**. The VPN Connections policy is displayed as shown in the following image:

Policy Name	Status	Processing Order	Source
Virtual Private Network (VPN) Connections	Enabled	1	Remote Access Server(VPN-Dial up)
Use Windows authentication for all users	Enabled	999999	Unspecified

Condition	Value
NAS Port Type	Virtual (VPN)

5. Under **Policies**, select **Network Policies**. You should see a Virtual Private Network (VPN) Connections policy that resembles the policy shown in the following image:



Configure your VPN server to use RADIUS authentication

In this section, you configure your VPN server to use RADIUS authentication. The instructions assume that you have a working configuration of a VPN server but haven't configured it to use RADIUS authentication. After you configure the VPN server, confirm that your configuration is working as expected.

Note

If you already have a working VPN server configuration that uses RADIUS authentication, you can skip this section.

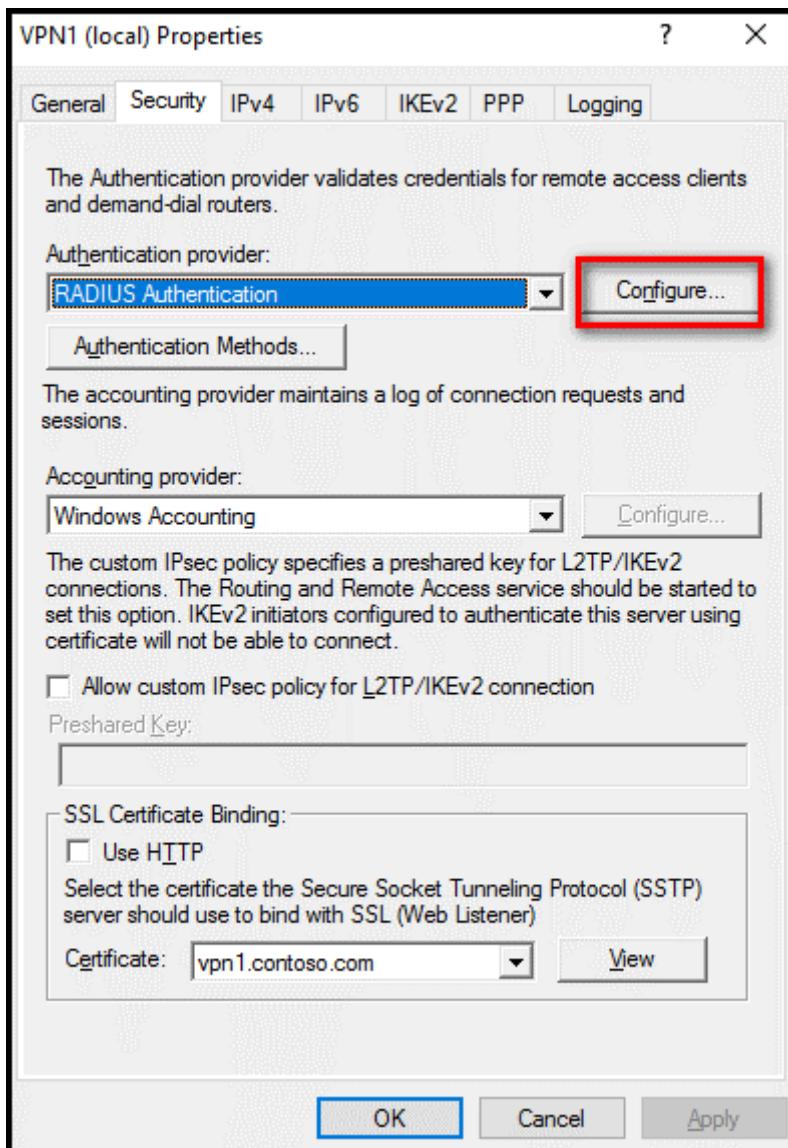
Configure authentication provider

1. On the VPN server, open Server Manager.
2. In Server Manager, select **Tools**, and then select **Routing and Remote Access**.

3. In the Routing and Remote Access window, right-click <server name> (local), and then select **Properties**.

4. In the <server name> (local) Properties window, select the **Security** tab.

5. On the **Security** tab, under **Authentication provider**, select **RADIUS Authentication**, and then select **Configure**.



6. In the RADIUS Authentication window, select **Add**.

7. In the Add RADIUS Server window, do the following:

- a. In the **Server name** box, enter the name or IP address of the RADIUS server that you configured in the previous section.
- b. For the **Shared secret**, select **Change**, and then enter the shared secret password that you created and recorded earlier.
- c. In the **Time-out (seconds)** box, enter a value of **60**. To minimize discarded requests, we recommend that VPN servers are configured with a timeout of at

least 60 seconds. If needed, or to reduce discarded requests in the event logs, you can increase the VPN server timeout value to 90 or 120 seconds.

8. Select OK.

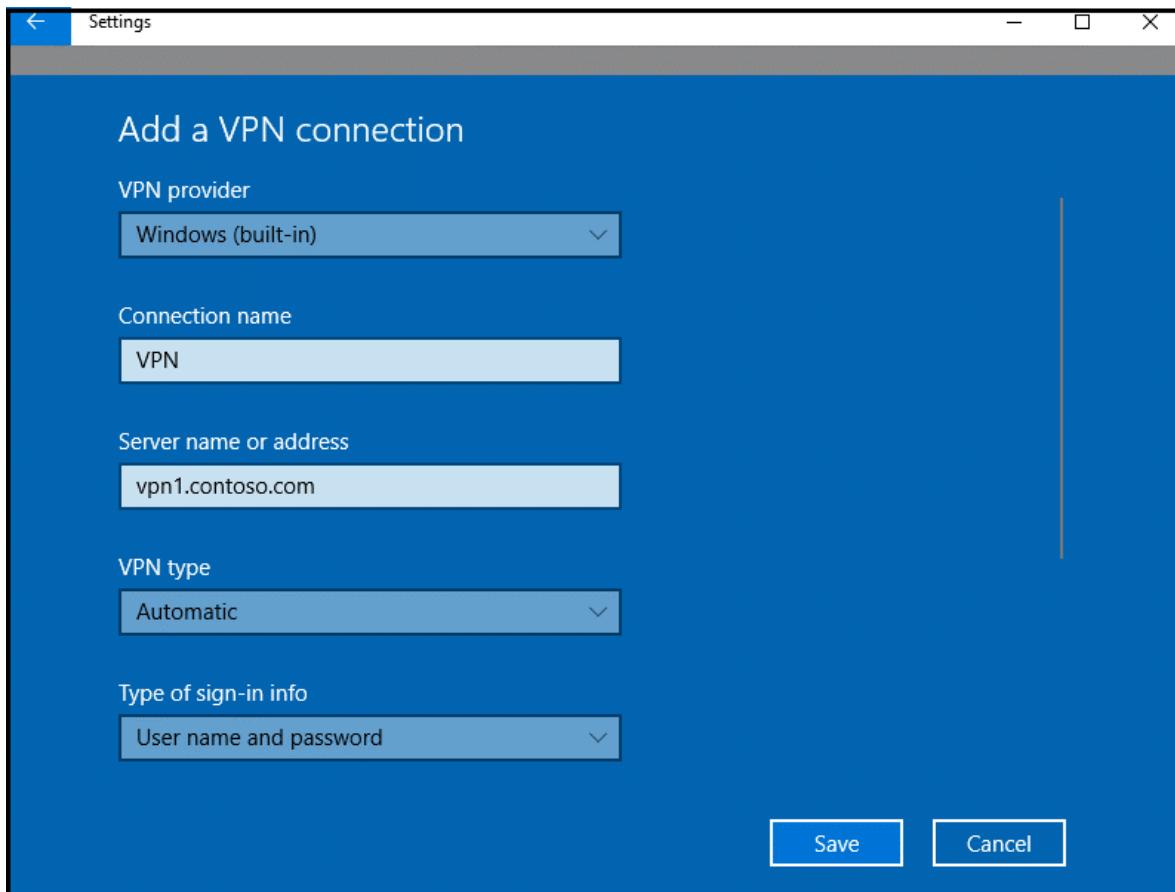
Test VPN connectivity

In this section, you confirm that the RADIUS server authenticates and authorizes the VPN client when you attempt to connect to the VPN virtual port. The instructions assume you're using Windows 10 as a VPN client.

Note

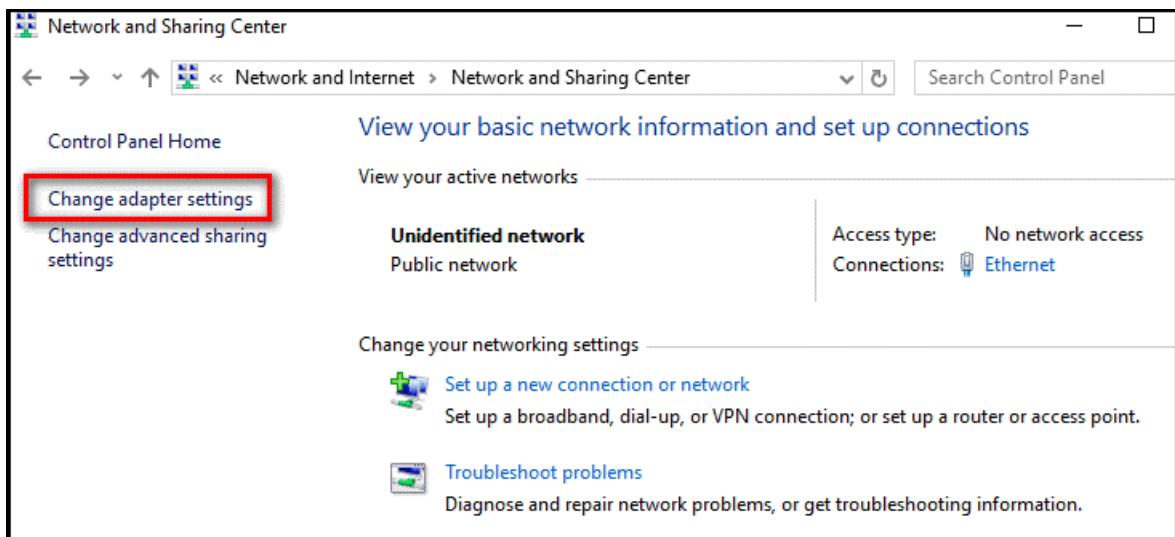
If you already configured a VPN client to connect to the VPN server and saved the settings, you can skip the steps related to configuring and saving a VPN connection object.

1. On your VPN client computer, select the **Start** button, and then select the **Settings** button.
2. In the **Windows Settings** window, select **Network & Internet**.
3. Select **VPN**.
4. Select **Add a VPN connection**.
5. In the **Add a VPN connection** window, in the **VPN provider** box, select **Windows (built-in)**, complete the remaining fields, as appropriate, and then select **Save**.



6. Go to Control Panel, and then select Network and Sharing Center.

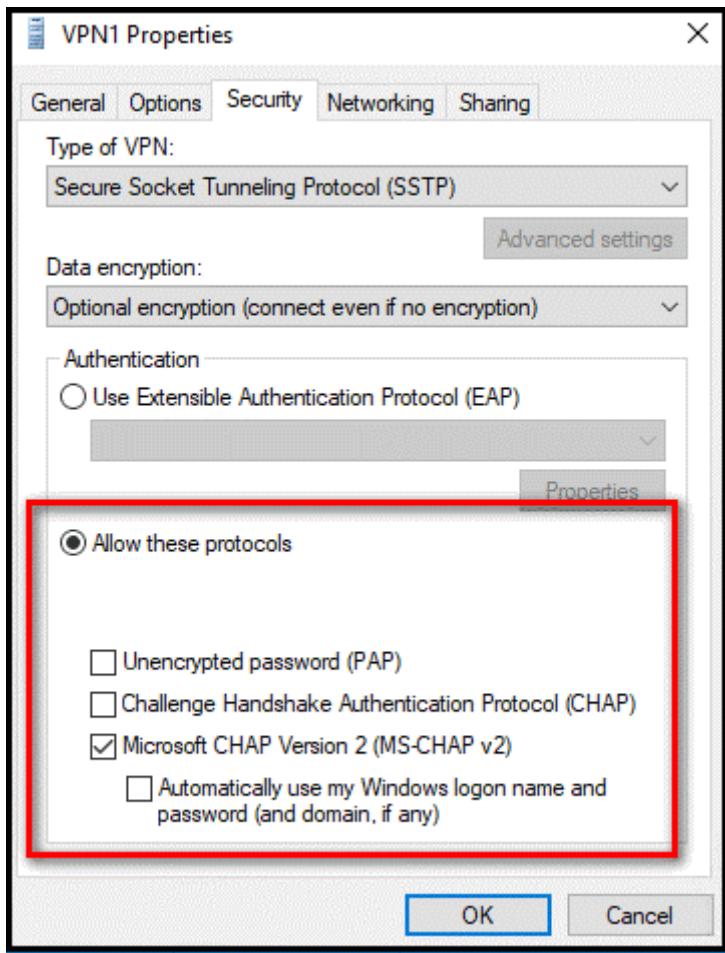
7. Select Change adapter settings.



8. Right-click the VPN network connection, and then select Properties.

9. In the VPN properties window, select the Security tab.

10. On the Security tab, ensure that only Microsoft CHAP Version 2 (MS-CHAP v2) is selected, and then select OK.



11. Right-click the VPN connection, and then select **Connect**.

12. In the **Settings** window, select **Connect**.

A successful connection appears in the Security log, on the RADIUS server, as Event ID 6272, as shown here:

Event Properties - Event 6272, Microsoft Windows security auditing.	
General Details	
Network Policy Server granted access to a user.	
User:	
Security ID:	CONTOSO\AliceC
Account Name:	alicec
Account Domain:	CONTOSO
Fully Qualified Account Name:	CONTOSO\alicec
Client Machine:	
Security ID:	NULL SID
Account Name:	-
Fully Qualified Account Name:	-
Called Station Identifier:	192.168.102.253
Calling Station Identifier:	192.168.102.2
NAS:	
NAS IPv4 Address:	192.168.101.253
NAS IPv6 Address:	-
NAS Identifier:	VPN1
NAS Port-Type:	Virtual
NAS Port:	513
RADIUS Client:	
Client Friendly Name:	VPN1
Client IP Address:	192.168.101.253
Authentication Details:	
Connection Request Policy Name:	Virtual Private Network (VPN) Connections
Network Policy Name:	Virtual Private Network (VPN) Connections
Authentication Provider:	Windows
Authentication Server:	DC1.contoso.com
Authentication Type:	MS-CHAPv2
EAP Type:	-
Account Session Identifier:	34
Logging Results:	Accounting information was written to the local log file.

Troubleshooting RADIUS

Assume that your VPN configuration was working before you configured the VPN server to use a centralized RADIUS server for authentication and authorization. If the configuration was working, it's likely that a misconfiguration of the RADIUS server or the use of an invalid username or password caused the issue. For example, if you use the alternate UPN suffix in the username, the sign-in attempt might fail. Use the same account name for best results.

To troubleshoot these issues, an ideal place to start is to examine the Security event logs on the RADIUS server. To save time searching for events, you can use the role-based Network Policy and Access Server custom view in Event Viewer, as shown here. "Event ID 6273" indicates events where the NPS denied access to a user.

Network Policy and Access Services Number of events: 8 (!) New events available		
Number of events: 8		
Level	Date and Time	Source
Information	6/11/2017 10:09:30 AM	NPS
Information	6/11/2017 10:08:32 AM	Microsoft Windows security audit...
Information	6/10/2017 4:10:19 PM	Microsoft Windows security audit...
Information	6/10/2017 4:05:53 PM	Microsoft Windows security audit...

<

Event 6273, Microsoft Windows security auditing.
--

General Details

Account Name:	CONTOSO\alicec
Account Domain:	CONTOSO
Fully Qualified Account Name:	CONTOSO\alicec
Client Machine:	
Security ID:	NULL SID
Account Name:	-
Fully Qualified Account Name:	-
Called Station Identifier:	192.168.102.253
Calling Station Identifier:	192.168.102.2
NAS:	
NAS IPv4 Address:	192.168.101.253
NAS IPv6 Address:	-
NAS Identifier:	VPN1
NAS Port-Type:	Virtual
NAS Port:	513
RADIUS Client:	
Client Friendly Name:	VPN1
Client IP Address:	192.168.101.253
Authentication Details:	
Connection Request Policy Name:	Virtual Private Network (VPN) Connections
Network Policy Name:	Virtual Private Network (VPN) Connections
Authentication Provider:	Windows
Authentication Server:	DC1.contoso.com
Authentication Type:	EAP
EAP Type:	-
Account Session Identifier:	3133
Logging Results:	Accounting information was written to the local log file.
Reason Code:	66
Reason:	The user attempted to use an authentication method that is not enabled on the matching network policy.

Configure multifactor authentication

For assistance configuring users for multifactor authentication, see the articles [Planning a cloud-based Microsoft Entra multifactor authentication deployment](#) and [Set up my account for two-step verification](#) ↗

Install and configure the NPS extension

This section provides instructions for configuring VPN to use MFA for client authentication with the VPN server.

ⓘ Note

The REQUIRE_USER_MATCH registry key is case sensitive. All values must be set in UPPER CASE format.

After you install and configure the NPS extension, this server requires all RADIUS-based client authentication to use MFA. If all your VPN users are not enrolled in Microsoft Entra multifactor authentication, you can do either of the following:

- Set up another RADIUS server to authenticate users who are not configured to use MFA.
- Create a registry entry that allows challenged users to provide a second authentication factor if they are enrolled in Microsoft Entra multifactor authentication.

Create a new string value named *REQUIRE_USER_MATCH* in *HKLM\SOFTWARE\Microsoft\AzureMfa*, and set the value to *TRUE* or *FALSE*.

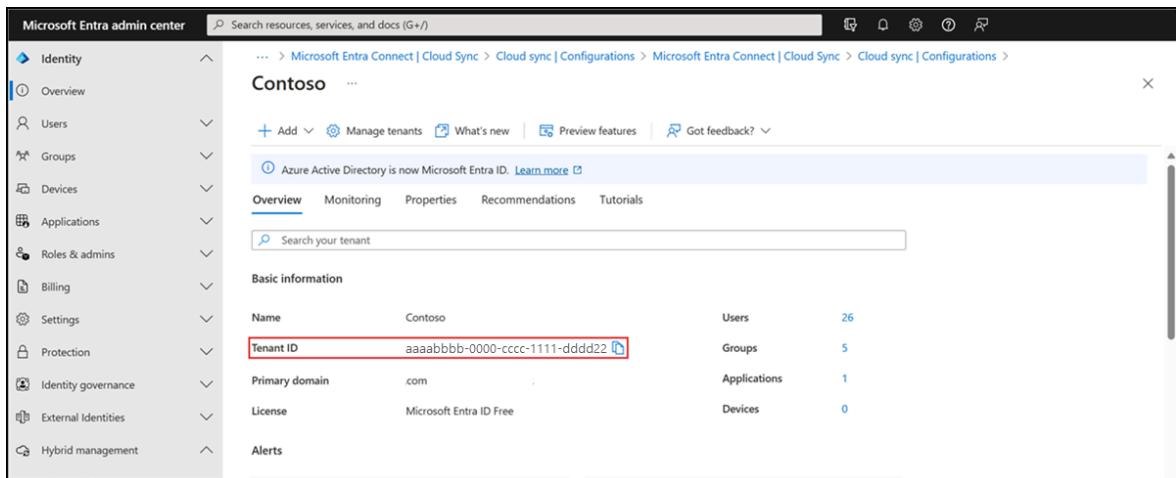
Name	Type	Data
ab (Default)	REG_SZ	(value not set)
ab AZURE_MFA_HOSTN...	REG_SZ	adnotifications.windowsazure.com
ab AZURE_MFA_TARGET...	REG_SZ	StrongAuthenticationService.svc/Connector
ab CLIENT_CERT_IDENI...	REG_SZ	CN=9ba26dc2-6843-4f09-8f62-cfa66648902c, OU=...
ab CLIENT_ID	REG_SZ	981f26a1-7f43-403b-a875-f8b09b8cd720
ab HELP_DOC_URL	REG_SZ	Forthcoming_Feature
ab STS_URL	REG_SZ	https://login.windows.net/
ab TENANT_ID	REG_SZ	9ba26dc2-6843-4f09-8f62-cfa66648902c
ab VERPOSE_LOC	REG_SZ	FALSE
ab REQUIRE_USER_MATCH	REG_SZ	TRUE

If the value is set to *TRUE* or is blank, all authentication requests are subject to an MFA challenge. If the value is set to *FALSE*, MFA challenges are issued only to users who are enrolled in Microsoft Entra multifactor authentication. Use the *FALSE* setting only in testing or in production environments during an onboarding period.

Obtain the directory tenant ID

As part of the configuration of the NPS extension, you must supply administrator credentials and the ID of your Microsoft Entra tenant. To get the tenant ID, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#).
2. Browse to **Identity > Settings**.



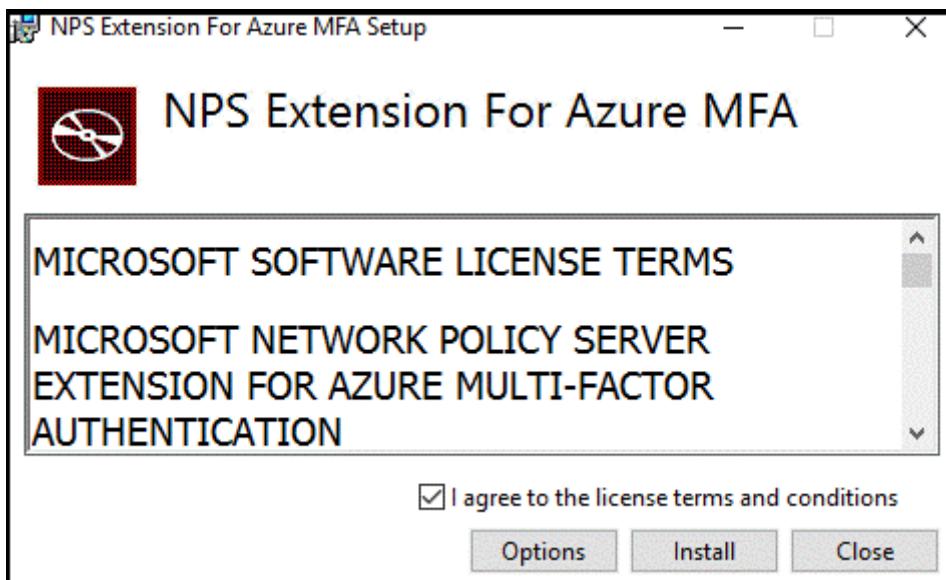
The screenshot shows the Microsoft Entra admin center interface. On the left, there's a navigation sidebar under the 'Identity' section. The main area displays basic information for the tenant 'Contoso'. A red box highlights the 'Tenant ID' field, which contains the value 'aaaaabbb-0000-cccc-1111-dddd22'. Other visible details include the primary domain 'com', license 'Microsoft Entra ID Free', and various user and group counts.

Name	Value	Users	Groups	Applications	Devices
Name	Contoso	26	5	1	0
Tenant ID	aaaaabbb-0000-cccc-1111-dddd22				
Primary domain	com				
License	Microsoft Entra ID Free				

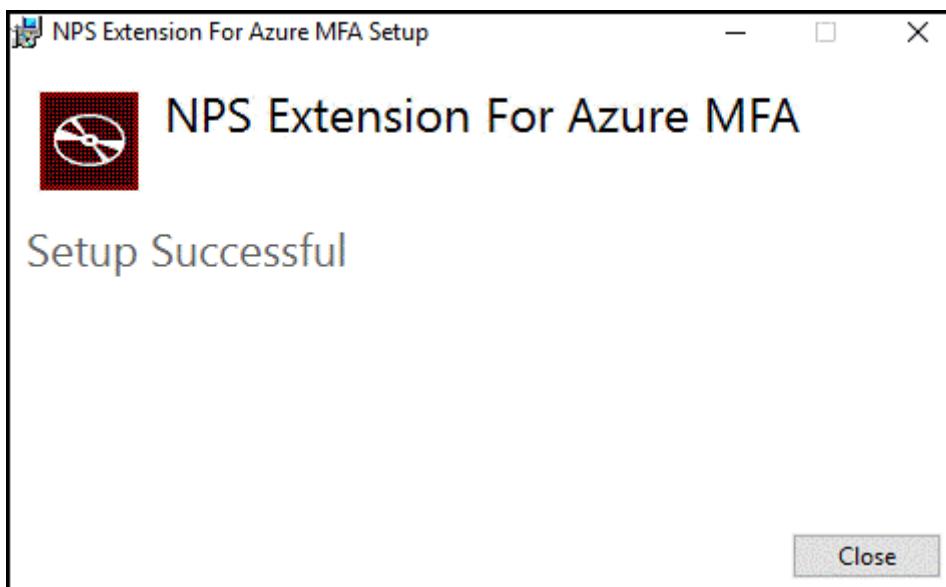
Install the NPS extension

The NPS extension must be installed on a server that has the Network Policy and Access Services role installed and that functions as the RADIUS server in your design. Do *not* install the NPS extension on your VPN server.

1. Download the NPS extension from [Microsoft Download Center](#).
2. Copy the setup executable file (`NpsExtnForAzureMfaInstaller.exe`) to the NPS server.
3. On the NPS server, double-click `NpsExtnForAzureMfaInstaller.exe` and, if you're prompted, select **Run**.
4. In the **NPS Extension For Microsoft Entra multifactor authentication Setup** window, review the software license terms, select the **I agree to the license terms and conditions** check box, and then select **Install**.



5. In the NPS Extension For Microsoft Entra multifactor authentication Setup window, select Close.



Configure certificates for use with the NPS extension by using a Graph PowerShell script

To ensure secure communications and assurance, configure certificates for use by the NPS extension. The NPS components include a Graph PowerShell script that configures a self-signed certificate for use with NPS.

The script performs the following actions:

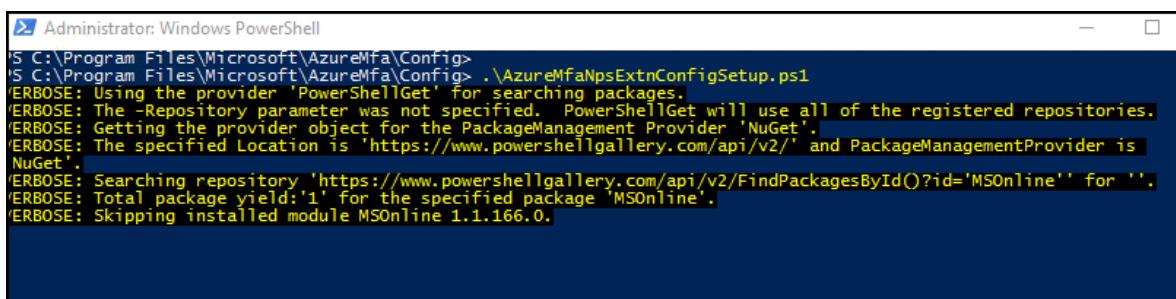
- Creates a self-signed certificate.
- Associates the public key of the certificate to the service principal on Microsoft Entra ID.
- Stores the certificate in the local machine store.
- Grants the network user access to the certificate's private key.

- Restarts the NPS service.

If you want to use your own certificates, you must associate the public key of your certificate with the service principal on Microsoft Entra ID, and so on.

To use the script, provide the extension with your Microsoft Entra administrative credentials and the Microsoft Entra tenant ID that you copied earlier. The account must be in the same Microsoft Entra tenant as you wish to enable the extension for. Run the script on each NPS server where you install the NPS extension.

1. Run Graph PowerShell as an administrator.
2. At the PowerShell command prompt, enter `cd "c:\Program Files\Microsoft\AzureMfa\Config"`, and then select Enter.
3. At the next command prompt, enter `.\\AzureMfaNpsExtnConfigSetup.ps1`, and then select Enter. The script checks to see whether Graph PowerShell is installed. If it isn't installed, the script installs Graph PowerShell for you.



```
'S C:\Program Files\Microsoft\AzureMfa\Config>
'S C:\Program Files\Microsoft\AzureMfa\Config> .\AzureMfaNpsExtnConfigSetup.ps1
'ERBOSE: Using the provider 'PowerShellGet' for searching packages.
'ERBOSE: The -Repository parameter was not specified. PowerShellGet will use all of the registered repositories.
'ERBOSE: Getting the provider object for the PackageManagement Provider 'NuGet'.
'ERBOSE: The specified Location is 'https://www.powershellgallery.com/api/v2/' and PackageManagementProvider is
'NuGet'.
'ERBOSE: Searching repository 'https://www.powershellgallery.com/api/v2/FindPackagesById()?id='MSOnline'' for ''.
'ERBOSE: Total package yield:1' for the specified package 'MSOnline'.
'ERBOSE: Skipping installed module MSOnline 1.1.166.0.'
```

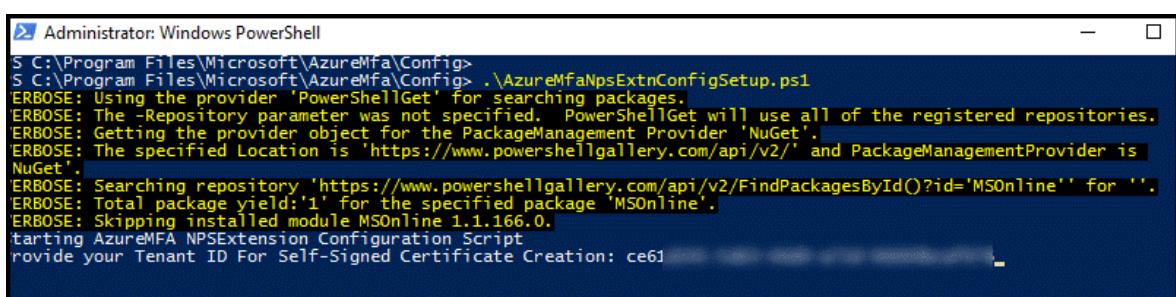
If you get a security error due to TLS, enable TLS 1.2 using the

```
[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12
```

command from your PowerShell prompt.

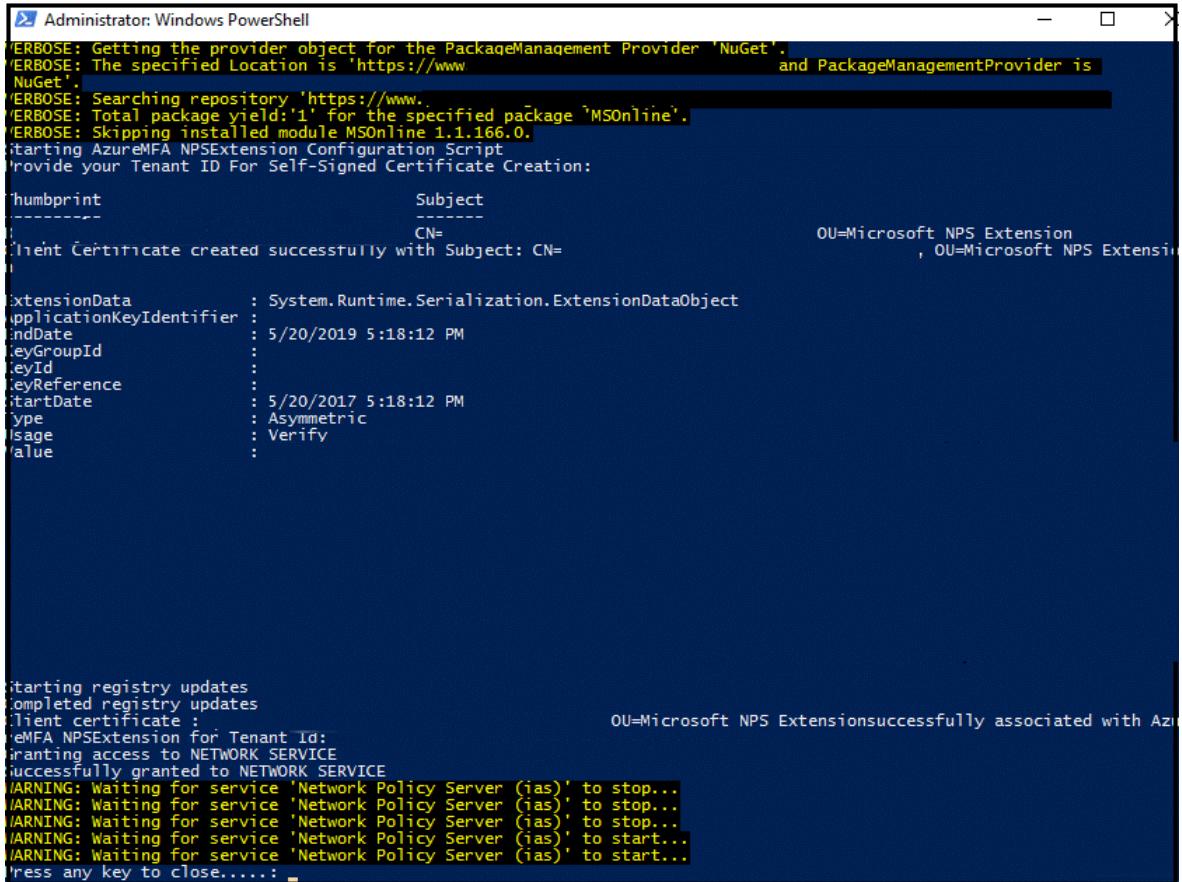
After the script verifies the installation of the PowerShell module, it displays the Graph PowerShell module sign-in window.

4. Enter your Microsoft Entra administrator credentials and password, and then select **Sign in**.
5. At the command prompt, paste the tenant ID that you copied earlier, and then select Enter.



```
'S C:\Program Files\Microsoft\AzureMfa\Config>
'S C:\Program Files\Microsoft\AzureMfa\Config> .\AzureMfaNpsExtnConfigSetup.ps1
'ERBOSE: Using the provider 'PowerShellGet' for searching packages.
'ERBOSE: The -Repository parameter was not specified. PowerShellGet will use all of the registered repositories.
'ERBOSE: Getting the provider object for the PackageManagement Provider 'NuGet'.
'ERBOSE: The specified Location is 'https://www.powershellgallery.com/api/v2/' and PackageManagementProvider is
'NuGet'.
'ERBOSE: Searching repository 'https://www.powershellgallery.com/api/v2/FindPackagesById()?id='MSOnline'' for ''.
'ERBOSE: Total package yield:1' for the specified package 'MSOnline'.
'ERBOSE: Skipping installed module MSOnline 1.1.166.0.
Starting AzureMFA NPSExtension Configuration Script
Provide your Tenant ID For Self-Signed Certificate Creation: ce61'
```

The script creates a self-signed certificate and performs other configuration changes. The output is like that in the following image:



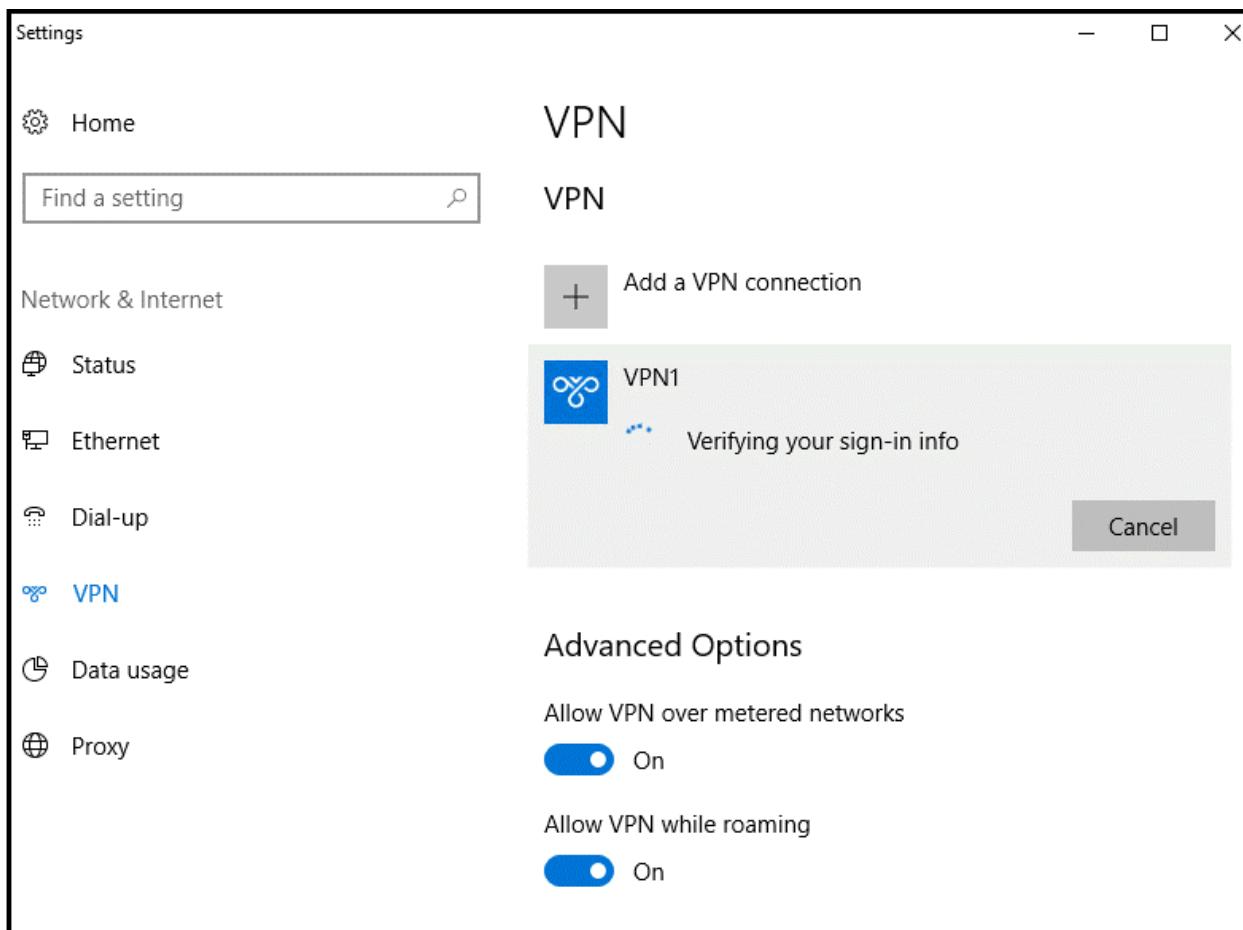
```
Administrator: Windows PowerShell
[ERBOSE: Getting the provider object for the PackageManagement Provider 'NuGet'.
[ERBOSE: The specified Location is 'https://www.NuGet.org/api/v2' and PackageManagementProvider is 'NuGet'.
[ERBOSE: Searching repository 'https://www.NuGet.org/api/v2'.
[ERBOSE: Total package yield: 1 for the specified package 'MSOnline'.
[ERBOSE: Skipping installed module MSOnline 1.1.166.0.
Starting AzureMFA NPSExtension Configuration Script
Provide your Tenant ID For Self-Signed Certificate Creation:
thumbprint Subject
----- CN= OU=Microsoft NPS Extension
Client Certificate created successfully with Subject: CN= , OU=Microsoft NPS Extension
ExtensionData : System.Runtime.Serialization.ExtensionDataObject
ApplicationKeyIdentifier : 
EndDate : 5/20/2019 5:18:12 PM
KeyGroupId :
KeyId :
KeyReference :
StartDate : 5/20/2017 5:18:12 PM
Type : Asymmetric
Usage : Verify
Value :

Starting registry updates
Completed registry updates
Client certificate : successfully associated with AzureMFA NPSExtension for Tenant Id: granting access to NETWORK SERVICE
Successfully granted to NETWORK SERVICE
[WARNING: Waiting for service 'Network Policy Server (ias)' to stop...
[WARNING: Waiting for service 'Network Policy Server (ias)' to stop...
[WARNING: Waiting for service 'Network Policy Server (ias)' to stop...
[WARNING: Waiting for service 'Network Policy Server (ias)' to start...
[WARNING: Waiting for service 'Network Policy Server (ias)' to start...
Press any key to close.....:
```

6. Reboot the server.

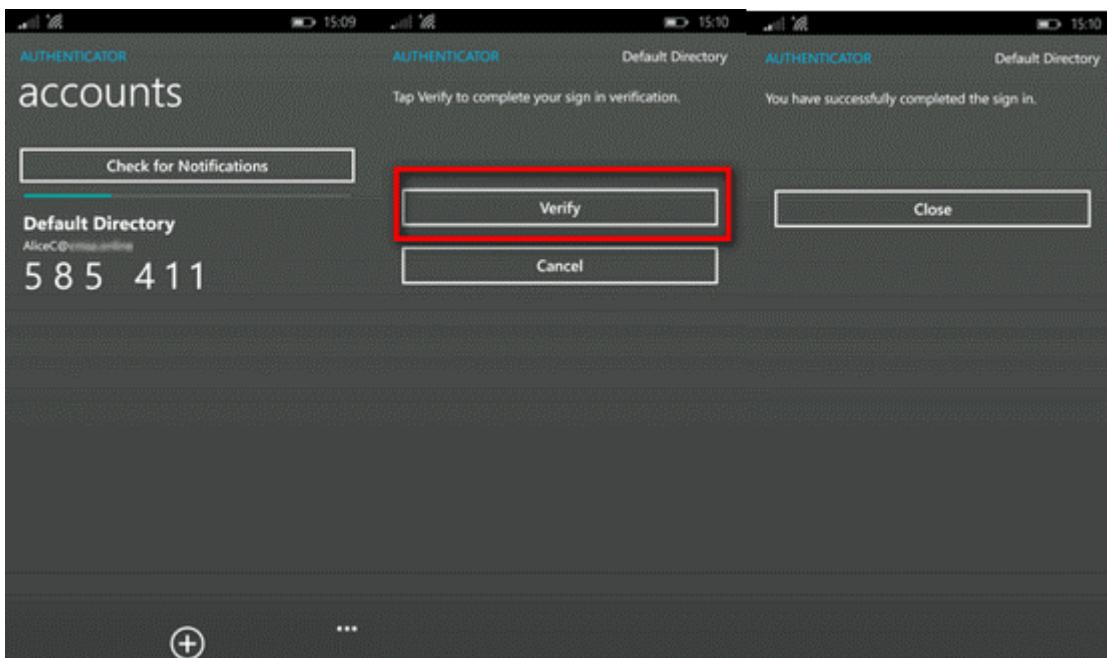
Verify the configuration

To verify the configuration, you must establish a new VPN connection with the VPN server. After you've successfully entered your credentials for primary authentication, the VPN connection waits for the secondary authentication to succeed before the connection is established, as shown in the following section.



If you successfully authenticate with the secondary verification method that you previously configured in Microsoft Entra multifactor authentication, you're connected to the resource. However, if the secondary authentication is unsuccessful, you're denied access to the resource.

In the following example, the Microsoft Authenticator app on a Windows Phone provides the secondary authentication:



After you've successfully authenticated by using the secondary method, you're granted access to the virtual port on the VPN server. Because you were required to use a secondary authentication method by using a mobile app on a trusted device, the sign-in process is more secure than if it were using only a username and password combination.

View Event Viewer logs for successful sign-in events

To view successful sign-in events in the Windows Event Viewer, you can view the Security log or the Network Policy and Access Services custom view, as shown in the following image:

The screenshot shows the Windows Event Viewer interface. At the top, a title bar reads "Network Policy and Access Services" and "Number of events: 9". Below this, a table lists four events, all categorized under "Information" level and "Microsoft Windows security auditing" source. The details for each event are as follows:

Level	Date and Time	Source
Information	6/12/2017 3:15:46 PM	Microsoft Windows security auditi...
Information	6/12/2017 2:56:04 PM	Microsoft Windows security auditi...
Information	6/12/2017 2:56:02 PM	Microsoft Windows security auditi...
Information	6/12/2017 2:54:02 PM	Microsoft Windows security auditi...

Event 6272, Microsoft Windows security auditing, is selected. The "Details" tab is active, showing the following information:

Network Policy Server granted access to a user.

User:

Security ID:	CONTOSO\AliceC
Account Name:	alicec
Account Domain:	CONTOSO
Fully Qualified Account Name:	CONTOSO\alicec

Client Machine:

Security ID:	NULL SID
Account Name:	-
Fully Qualified Account Name:	-
Called Station Identifier:	192.168.102.253
Calling Station Identifier:	192.168.102.2

NAS:

NAS IPv4 Address:	192.168.101.253
NAS IPv6 Address:	-
NAS Identifier:	VPN1
NAS Port-Type:	Virtual
NAS Port:	129

RADIUS Client:

Client Friendly Name:	VPN1
Client IP Address:	192.168.101.253

Authentication Details:

Connection Request Policy Name:	Virtual Private Network (VPN) Connections
Network Policy Name:	Virtual Private Network (VPN) Connections
Authentication Provider:	Windows
Authentication Server:	DC1.contoso.com
Authentication Type:	Extension
EAP Type:	-
Account Session Identifier:	37

Logging Results: Accounting information was written to the local log file.

On the server where you installed the NPS extension for Microsoft Entra multifactor authentication, you can find Event Viewer application logs that are specific to the extension at *Application and Services Logs\Microsoft\AzureMfa*.

AuthZOptCh Number of events: 3		
Level	Date and Time	Source
Information	6/12/2017 3:15:46 PM	AuthZ
Information	6/12/2017 2:56:04 PM	AuthZ
Information	6/12/2017 2:54:00 PM	AuthZ

Event 2, AuthZ

General Details

NPS Extension for Azure MFA: CID: a0a0a0a0-bbbb-cccc-dddd-e1e1e1e1e1e1: Access Accepted for user 0000021812FA1D10 with Azure MFA response Success message 000000298BFFCC28 session b1b1b1b1-cccc-dddd-eeee-f2f2f2f2f2f2

Log Name:	Microsoft-AzureMfa-AuthZ/AuthZOptCh		
Source:	AuthZ	Logged:	6/12/2017 3:15:46 PM
Event ID:	2	Task Category:	None
Level:	Information	Keywords:	
User:	NETWORK SERVICE	Computer:	DC1.contoso.com
OpCode:	Info		
More Information:	Event Log Online Help		

Troubleshooting guide

If the configuration isn't working as expected, begin troubleshooting by verifying that the user is configured to use MFA. Have the user sign in to the [Microsoft Entra admin center](#). If the user is prompted for secondary authentication and can successfully authenticate, you can eliminate an incorrect configuration of MFA as an issue.

If MFA is working for the user, review the relevant Event Viewer logs. The logs include the security event, Gateway operational, and Microsoft Entra multifactor authentication logs that are discussed in the previous section.

An example of a security log that displays a failed sign-in event (event ID 6273) is shown here:

```

TimeCreated : 6/12/2017 2:28:45 PM
ProviderName : Microsoft-Windows-Security-Auditing
Id : 6273
Message : Network Policy Server denied access to a user.

        Contact the Network Policy Server administrator for more information.

        User:
        Security ID: aaaaaaaaa-0b0b-1c1c-2d2d-333333333333
        Account Name: a
        Account Domain: CONTOSO
        Fully Qualified Account Name: CONTOSO\a

        Client Machine:
        Security ID: bbbbbbbb-0c0c-1d1d-3e4d-444444444444
        Account Name: -
        Fully Qualified Account Name: -
        Called Station Identifier: -
        Calling Station Identifier: -

        NAS:
        NAS IPv4 Address: -
        NAS IPv6 Address: -
        NAS Identifier: VPN1
        NAS Port-Type: Virtual
        NAS Port: -

        RADIUS Client:
        Client Friendly Name: VPN1
        Client IP Address: -

        Authentication Details:
        Connection Request Policy Name: Virtual Private Network (VPN) Connections
        Network Policy Name: Virtual Private Network (VPN) Connections
        Authentication Provider: Windows
        Authentication Server: DC1.
        Authentication Type: EAP
        EAP Type: -
        Account Session Identifier: 32
        Logging Results: Accounting information was written to the local log file.
        Reason Code: 66
        Reason: The user attempted to use an authentication method that is not enabled o
n the matching
network policy.

```

A related event from the Microsoft Entra multifactor authentication log is shown here:

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs categorized by source, including Event Viewer (Local), Server Roles, Windows Logs, Applications and Services Logs, and Microsoft logs for various services like AppV, AzureMfa, and AuthZ. The right pane shows a list of events for the 'AuthZOptCh' log, which has 74 events. One specific event is selected, showing its details in a modal window.

Event Properties - Event 2, AuthZ

General

NPS Extension for Azure MFA: Access Rejected for user with Azure MFA response PhoneAppDenied and message

Details

Log Name:	Microsoft-
Source:	AuthZ
Event ID:	2
Level:	Information
User:	NETWORK SERVICE
OpCode:	Info
More Information:	Event Log Online Help

Buttons: Copy, Close

To do advanced troubleshooting, consult the NPS database format log files where the NPS service is installed. The log files are created in the %SystemRoot%\System32\Logs folder as comma-delimited text files. For a description of the log files, see [Interpret NPS Database Format Log Files](#).

The entries in these log files are difficult to interpret unless you export them to a spreadsheet or a database. You can find many Internet Authentication Service (IAS) parsing tools online to assist you in interpreting the log files. The output of one such downloadable [shareware application](#) is shown here:

The screenshot shows the IAS Log Viewer interface. At the top, there's a menu bar with File, Edit, View, Reports, Tools, and Help. Below the menu is a toolbar with various icons. The main area has tabs for Records, Connects, Alerts, and Scheduled Tasks. The Records tab is selected, displaying a table of log entries. The table columns include Start DateTime, User Name, Stop DateTime, Duration, User IP, Output Octets, Input Octets, Connect Request, and Connect Re... (with a tooltip 'The request was discarded by a third-party ext...'). A tooltip for 'Input Octets' also appears above the column header. The table contains several rows of log data. To the right of the table, a detailed view pane is open, showing properties for a selected log entry. This pane includes fields like Called Station Id, Client IP Address, Connect Request, Connect Result, Duration, Record Count, Server Name, Session Time, Start Date/Time, and Terminate Cause. Some fields have their own tool tips. The detailed view pane has scroll bars.

To do additional troubleshooting, you can use a protocol analyzer such as Wireshark or [Microsoft Message Analyzer](#). The following image from Wireshark shows the RADIUS messages between the VPN server and the NPS.

*Ethernet 2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

radius

No.	Time	Source	Destination	Protocol	Length	Info
77	3.142697	192.168.101.253	192.168.101.1	RADIUS	428	Access-Request(1) (id=10, l=386)
201	18.115217	192.168.101.1	192.168.101.253	RADIUS	317	Access-Accept(2) (id=10, l=275)

```
> Frame 77: 428 bytes on wire (3424 bits), 428 bytes captured (3424 bits) on interface 0
> Ethernet II, Src: Microsoft_39:16:0d (00:15:5d:39:16:0d), Dst: Microsoft_39:16:0b (00:15:5d:39:16:0b)
> Internet Protocol Version 4, Src: 192.168.101.253, Dst: 192.168.101.1
> User Datagram Protocol, Src Port: 61328, Dst Port: 1812
> RADIUS Protocol

0000  00 15 5d 39 16 0b 00 15  5d 39 16 0d 08 00 45 00  ..]9..... ]9....E.
0010  01 9e 39 0d 00 00 80 11  b3 f2 c0 a8 65 fd c0 a8  ..9..... ....e...
0020  65 01 ef 90 07 14 01 8a  b7 d7 01 0a 01 82 1f 43  e..... .....C
0030  71 1c 47 26 4c 72 57 52  01 ae 9d f9 8b 50 2c 03  q.G&LrWR .....P,.
0040  38 20 06 56 50 4e 31 04  06 c0 a8 65 fd 06 06 00  8 .VPN1. ....e....
0050  00 00 02 07 06 00 00 00  01 05 06 00 00 00 81 3d  ..... ....=.
0060  06 00 00 00 05 40 06 00  00 00 01 41 06 00 00 00  .....@.. ...A....
0070  01 1e 11 31 39 32 2e 31  36 38 2e 31 30 32 2e 32  ...192.1 68.102.2
0080  35 33 43 11 31 39 32 2e  31 36 38 2e 31 30 32 2e  53C.192. 168.102.
0090  32 35 33 1f 0f 31 39 32  2e 31 36 38 2e 31 30 32  253..192. 168.102
00a0  2e 32 42 0f 31 39 32 2e  31 36 38 2e 31 30 32 2e  .2B.192. 168.102.
00b0  32 01 08 61 6c 69 63 65  63 1a 0c 00 00 01 37 2f  2..alice c.....7/
00c0  06 00 00 00 02 1a 0c 00  00 01 37 09 06 00 00 01  ..... .7.....
00d0  37 1a 12 00 00 01 37 12  0c 4d 53 52 41 53 56 35  7.....7. .MSRASV5
00e0  2e 32 30 1a 2e 00 00 01  37 38 28 7b 34 45 30 39  .20..... 78({4E09
00f0  42 30 39 45 2d 41 43 42  42 2d 34 45 31 35 2d 42  B09E-ACB B-4E15-B
```

For more information, see [Integrate your existing NPS infrastructure with Microsoft Entra multifactor authentication](#).

Next steps

[Get Microsoft Entra multifactor authentication](#)

[Remote Desktop Gateway and Azure Multi-Factor Authentication Server using RADIUS](#)

[Integrate your on-premises directories with Microsoft Entra ID](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) ↗

Plan and deploy on-premises Microsoft Entra Password Protection

Article • 03/04/2025

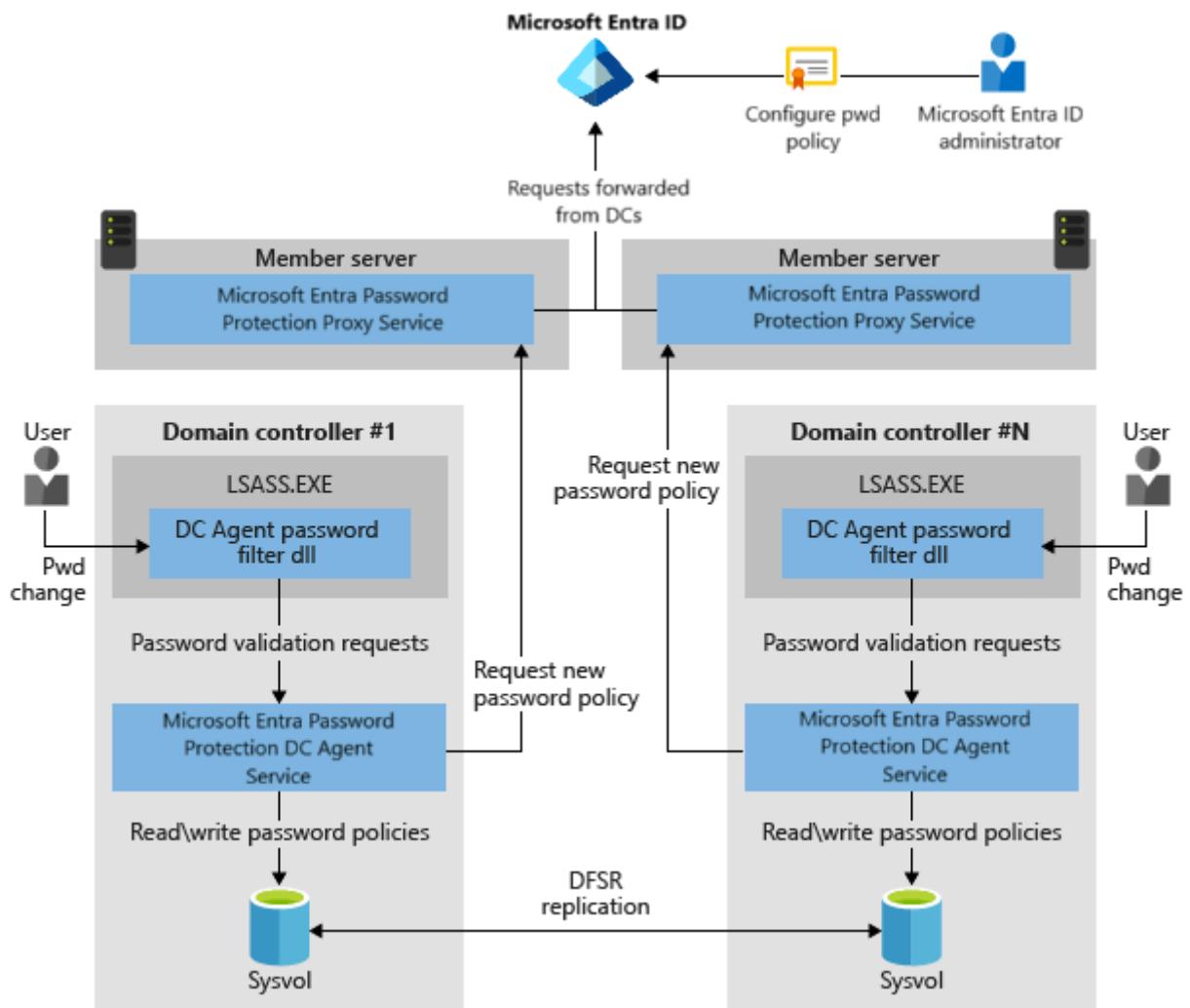
Users often create passwords that use common local words such as a school, sports team, or famous person. These passwords are easy to guess, and weak against dictionary-based attacks. To enforce strong passwords in your organization, Microsoft Entra Password Protection provides a global and custom banned password list. A password change request fails if there's a match in this banned password list.

To protect your on-premises Active Directory Domain Services (AD DS) environment, you can install and configure Microsoft Entra Password Protection to work with your on-prem DC. This article shows you how to install and register the Microsoft Entra Password Protection proxy service and Microsoft Entra Password Protection DC agent in your on-premises environment.

For more information on how Microsoft Entra Password Protection works in an on-premises environment, see [How to enforce Microsoft Entra Password Protection for Windows Server Active Directory](#).

Deployment strategy

The following diagram shows how the basic components of Microsoft Entra Password Protection work together in an on-premises Active Directory environment:



It's a good idea to review how the software works before you deploy it. For more information, see [Conceptual overview of Microsoft Entra Password Protection](#).

We recommend that you start deployments in *audit* mode. Audit mode is the default initial setting, where passwords can continue to be set. Passwords that would be blocked are recorded in the event log. After you deploy the proxy servers and DC agents in audit mode, monitor the impact that the password policy will have on users when the policy is enforced.

During the audit stage, many organizations find that the following situations apply:

- They need to improve existing operational processes to use more secure passwords.
- Users often use unsecure passwords.
- They need to inform users about the upcoming change in security enforcement, possible impact on them, and how to choose more secure passwords.

It's also possible for stronger password validation to affect your existing Active Directory domain controller deployment automation. We recommend that at least one DC promotion and one DC demotion happen during the audit period evaluation to help uncover such issues. For more information, see the following articles:

- Ntdsutil.exe is unable to set a weak Directory Services Repair Mode password
- Domain controller replica promotion fails because of a weak Directory Services Repair Mode password
- Domain controller demotion fails due to a weak local Administrator password

After the feature has been running in audit mode for a reasonable period, you can switch the configuration from *Audit* to *Enforce* to require more secure passwords. Extra monitoring during this time is a good idea.

It is important to note that Microsoft Entra Password Protection can only validate passwords during password change or set operations. Passwords that were accepted and stored in Active Directory prior to the deployment of Microsoft Entra Password Protection will never be validated and will continue working as-is. Over time, all users and accounts will eventually start using Microsoft Entra Password Protection-validated passwords as their existing passwords expire normally. Accounts configured with "password never expires" are exempt from this.

Multiple forest considerations

There are no additional requirements to deploy Microsoft Entra Password Protection across multiple forests.

Each forest is independently configured, as described in the following section to [deploy on-prem Microsoft Entra Password Protection](#). Each Microsoft Entra Password Protection proxy can only support domain controllers from the forest that it's joined to.

The Microsoft Entra Password Protection software in any forest is unaware of password protection software that's deployed in other forests, regardless of Active Directory trust configurations.

Read-only domain controller considerations

Password change or set events aren't processed and persisted on read-only domain controllers (RODCs). Instead, they're forwarded to writable domain controllers. You don't have to install the Microsoft Entra Password Protection DC agent software on RODCs.

Further, it's not supported to run the Microsoft Entra Password Protection proxy service on a read-only domain controller.

High availability considerations

The main concern for password protection is the availability of Microsoft Entra Password Protection proxy servers when the DCs in a forest try to download new policies or other data from Azure. Each Microsoft Entra Password Protection DC agent uses a simple round-robin-style algorithm when deciding which proxy server to call. The agent skips proxy servers that aren't responding.

For most fully connected Active Directory deployments that have healthy replication of both directory and sysvol folder state, two Microsoft Entra Password Protection proxy servers is enough to ensure availability. This configuration results in timely download of new policies and other data. You can deploy additional Microsoft Entra Password Protection proxy servers if desired.

The design of the Microsoft Entra Password Protection DC agent software mitigates the usual problems that are associated with high availability. The Microsoft Entra Password Protection DC agent maintains a local cache of the most recently downloaded password policy. Even if all registered proxy servers become unavailable, the Microsoft Entra Password Protection DC agents continue to enforce their cached password policy.

A reasonable update frequency for password policies in a large deployment is usually days, not hours or less. So, brief outages of the proxy servers don't significantly impact Microsoft Entra Password Protection.

Deployment requirements

For information on licensing, see [Microsoft Entra Password Protection licensing requirements](#).

The following core requirements apply:

- All machines, including domain controllers, that have Microsoft Entra Password Protection components installed must have the Universal C Runtime installed.
 - You can get the runtime by making sure you have all updates from Windows Update. Or you can get it in an OS-specific update package. For more information, see [Update for Universal C Runtime in Windows](#).
- You need an account that has Active Directory domain administrator privileges in the forest root domain to register the Windows Server Active Directory forest with Microsoft Entra ID.
- The Key Distribution Service must be enabled on all domain controllers in the domain that run Windows Server 2012 and later versions. By default, this service is enabled via manual trigger start.

- Network connectivity must exist between at least one domain controller in each domain and at least one server that hosts the proxy service for Microsoft Entra Password Protection. This connectivity must allow the domain controller to access RPC endpoint mapper port 135 and the RPC server port on the proxy service.
 - By default, the RPC server port is a dynamic RPC port from the range (49152 - 65535), but it can be configured to [use a static port](#).
- All machines where the Microsoft Entra Password Protection Proxy service will be installed must have network access to the following endpoints:

[\[+\] Expand table](#)

Endpoint	Purpose
https://login.microsoftonline.com	Authentication requests
https://enterpriseregistration.windows.net	Microsoft Entra Password Protection functionality
https://autoupdate.msappproxy.net	Microsoft Entra Password Protection auto-upgrade functionality

Note

Some endpoints, such as the CRL endpoint, are not addressed in this article. For a list of all supported endpoints, see [Microsoft 365 URLs and IP address ranges](#). In addition, other endpoints are required for Microsoft Entra admin center authentication. For more information, see [Microsoft Entra admin center URLs for proxy bypass](#).

Microsoft Entra Password Protection DC agent

The following requirements apply to the Microsoft Entra Password Protection DC agent:

- Machines where the Microsoft Entra Password Protection DC agent software will be installed must run Windows Server 2012 R2 or later, including Windows Server Core editions.
 - The Active Directory domain or forest can be any supported functional level.
- All machines where the Microsoft Entra Password Protection DC agent will be installed must have .NET 4.7.2 installed.
 - If .NET 4.7.2 is not already installed, download and run the installer found at [The .NET Framework 4.7.2 offline installer for Windows](#).

- Any Active Directory domain that runs the Microsoft Entra Password Protection DC agent service must use Distributed File System Replication (DFSR) for sysvol replication.
 - If your domain isn't already using DFSR, you must migrate before installing Microsoft Entra Password Protection. For more information, see [SYSVOL Replication Migration Guide: FRS to DFS Replication](#)

Warning

The Microsoft Entra Password Protection DC agent software will currently install on domain controllers in domains that are still using FRS (the predecessor technology to DFSR) for sysvol replication, but the software will NOT work properly in this environment.

Additional negative side-effects include individual files failing to replicate, and sysvol restore procedures appearing to succeed but silently failing to replicate all files.

Migrate your domain to use DFSR as soon as possible, both for DFSR's inherent benefits and to unblock the deployment of Microsoft Entra Password Protection. Future versions of the software will be automatically disabled when running in a domain that's still using FRS.

Microsoft Entra Password Protection proxy service

The following requirements apply to the Microsoft Entra Password Protection proxy service:

- All machines where the Microsoft Entra Password Protection proxy service will be installed must run Windows Server 2012 R2 or later, including Windows Server Core editions.

Note

The Microsoft Entra Password Protection proxy service deployment is a mandatory requirement for deploying Microsoft Entra Password Protection even though the domain controller may have outbound direct internet connectivity.

- All machines where the Microsoft Entra Password Protection proxy service will be installed must have .NET 4.7.2 installed.
 - If .NET 4.7.2 is not already installed, download and run the installer found at [The .NET Framework 4.7.2 offline installer for Windows](#).
- All machines that host the Microsoft Entra Password Protection proxy service must be configured to grant domain controllers the ability to log on to the proxy service. This ability is controlled via the "Access this computer from the network" privilege assignment.
- All machines that host the Microsoft Entra Password Protection proxy service must be configured to allow outbound TLS 1.2 HTTP traffic.
- A [Global Administrator](#) is required to register the Microsoft Entra Password Protection proxy service for the first time in a given tenant. Subsequent proxy and forest registrations with Microsoft Entra ID may use an account with at least the [Security Administrator](#) role.
- Network access must be enabled for the set of ports and URLs specified in the [application proxy environment setup procedures](#). This is in addition to the two endpoints described above.

Microsoft Entra Connect Agent Updater prerequisites

The Microsoft Entra Connect Agent Updater service is installed side by side with the Microsoft Entra Password Protection Proxy service. Additional configuration is required in order for the Microsoft Entra Connect Agent Updater service to be able to function:

- If your environment uses an HTTP proxy server, follow the guidelines specified in [Work with existing on-premises proxy servers](#).
- The Microsoft Entra Connect Agent Updater service also requires the TLS 1.2 steps specified in [TLS requirements](#).

Warning

Microsoft Entra Password Protection proxy and Microsoft Entra application proxy install different versions of the Microsoft Entra Connect Agent Updater service, which is why the instructions refer to Application Proxy content. These different versions are incompatible when installed side by side and doing so will prevent the Agent Updater service from contacting Azure for software updates, so you should never install Microsoft Entra Password Protection Proxy and Application Proxy on the same machine.

Download required software

There are two required installers for an on-premises Microsoft Entra Password Protection deployment:

- Microsoft Entra Password Protection DC agent
(*AzureADPasswordProtectionDCAgentSetup.msi*)
- Microsoft Entra Password Protection proxy
(*AzureADPasswordProtectionProxySetup.exe*)

Download both installers from the [Microsoft Download Center](#).

Install and configure the proxy service

The Microsoft Entra Password Protection proxy service is typically on a member server in your on-premises AD DS environment. Once installed, the Microsoft Entra Password Protection proxy service communicates with Microsoft Entra ID to maintain a copy of the global and customer banned password lists for your Microsoft Entra tenant.

In the next section, you install the Microsoft Entra Password Protection DC agents on domain controllers in your on-premises AD DS environment. These DC agents communicate with the proxy service to get the latest banned password lists for use when processing password change events within the domain.

Choose one or more servers to host the Microsoft Entra Password Protection proxy service. The following considerations apply for the server(s):

- Each such service can only provide password policies for a single forest. The host machine must be joined to any domain in that forest.
- You can install the proxy service in either root or child domains, or a combination of those.
- You need network connectivity between at least one DC in each domain of the forest and one password protection proxy server.
- You can run the Microsoft Entra Password Protection proxy service on a domain controller for testing, but that domain controller then requires internet connectivity. This connectivity can be a security concern. We recommend this configuration for testing only.
- We recommend at least two Microsoft Entra Password Protection proxy servers per forest for redundancy, as noted in the previous section on [high availability considerations](#).
- It's not supported to run the Microsoft Entra Password Protection proxy service on a read-only domain controller.

- If necessary, you can remove the proxy service by using **Add or remove programs**.
No manual cleanup of the state that the proxy service maintains is needed.

To install the Microsoft Entra Password Protection proxy service, complete the following steps:

1. To install the Microsoft Entra Password Protection proxy service, run the `AzureADPasswordProtectionProxySetup.exe` software installer.

The software installation doesn't require a reboot and may be automated using standard MSI procedures, as in the following example:

Console

```
AzureADPasswordProtectionProxySetup.exe /quiet
```

① Note

The Windows Firewall service must be running before you install the `AzureADPasswordProtectionProxySetup.exe` package to avoid an installation error.

If Windows Firewall is configured to not run, the workaround is to temporarily enable and run the Firewall service during the installation. The proxy software has no specific dependency on Windows Firewall after installation.

If you're using a third-party firewall, it must still be configured to satisfy the deployment requirements. These include allowing inbound access to port 135 and the proxy RPC server port. For more information, see the previous section on [deployment requirements](#).

2. The Microsoft Entra Password Protection proxy software includes a new PowerShell module, `AzureADPasswordProtection`. The following steps run various cmdlets from this PowerShell module.

To use this module, open a PowerShell window as an administrator and import the new module as follows:

PowerShell

```
Import-Module AzureADPasswordProtection
```

Warning

The 64 bit version of PowerShell must be used. Certain cmdlets may not work with PowerShell (x86).

3. To check that the Microsoft Entra Password Protection proxy service is running, use the following PowerShell command:

```
PowerShell
```

```
Get-Service AzureADPasswordProtectionProxy | fl
```

The result should show a **Status** of *Running*.

4. The proxy service is running on the machine, but doesn't have credentials to communicate with Microsoft Entra ID. Register the Microsoft Entra Password Protection proxy server with Microsoft Entra ID using the `Register-AzureADPasswordProtectionProxy` cmdlet.

This cmdlet requires *Global Administrator* credentials the first time any proxy is registered for a given tenant. Subsequent proxy registrations in that tenant, whether for the same or different proxies, may use *Security Administrator* credentials.

After this command succeeds once, additional invocations will also succeed but are unnecessary.

The `Register-AzureADPasswordProtectionProxy` cmdlet supports the following three authentication modes. The first two modes support Microsoft Entra multifactor authentication but the third mode doesn't.

Tip

There might be a noticeable delay before completion the first time that this cmdlet is run for a specific Azure tenant. Unless a failure is reported, don't worry about this delay.

- Interactive authentication mode:

```
PowerShell
```

```
Register-AzureADPasswordProtectionProxy -AccountUpn  
'yourglobaladmin@yourtenant.onmicrosoft.com'
```

 **Note**

This mode doesn't work on Server Core operating systems. Instead, use one of the following authentication modes. Also, this mode might fail if Internet Explorer Enhanced Security Configuration is enabled. The workaround is to disable that Configuration, register the proxy, and then re-enable it.

- Device-code authentication mode:

PowerShell

```
Register-AzureADPasswordProtectionProxy -AccountUpn  
'yourglobaladmin@yourtenant.onmicrosoft.com' -  
AuthenticateUsingDeviceCode
```

When prompted, following the link to open a web browser and enter the authentication code.

- Silent (password-based) authentication mode:

PowerShell

```
$globalAdminCredentials = Get-Credential  
Register-AzureADPasswordProtectionProxy -AzureCredential  
$globalAdminCredentials
```

 **Note**

This mode fails if Microsoft Entra multifactor authentication is required for your account. In that case, use one of the previous two authentication modes, or instead use a different account that does not require MFA.

You may also see MFA required if Azure Device Registration (which is used under the covers by Microsoft Entra Password Protection) has been configured to globally require MFA. To workaround this requirement you may use a different account that supports MFA with one of the previous

two authentication modes, or you may also temporarily relax the Azure Device Registration MFA requirement.

To make this change, select **Identity** in the [Microsoft Entra admin center](#), then select **Devices > Device Settings**. Set **Require multifactor authentication to join devices** to **No**. Be sure to reconfigure this setting back to **Yes** once registration is complete.

We recommend that MFA requirements be bypassed for test purposes only.

You don't currently have to specify the `-ForestCredential` parameter, which is reserved for future functionality.

Registration of the Microsoft Entra Password Protection proxy service is necessary only once in the lifetime of the service. After that, the Microsoft Entra Password Protection proxy service will automatically perform any other necessary maintenance.

5. To make sure that the changes have taken effect, run `Test-AzureADPasswordProtectionProxyHealth -TestAll`. For help resolving errors, see [Troubleshoot: On-premises Microsoft Entra Password Protection](#).
6. Now register the on-premises Active Directory forest with the necessary credentials to communicate with Azure by using the `Register-AzureADPasswordProtectionForest` PowerShell cmdlet.

 **Note**

If multiple Microsoft Entra Password Protection proxy servers are installed in your environment, it doesn't matter which proxy server you use to register the forest.

The cmdlet requires either *Global Administrator* or *Security Administrator* credentials for your Azure tenant. It also requires on-premises Active Directory Enterprise Administrator privileges. You must also run this cmdlet using an account with local administrator privileges. The Azure account that is used to register the forest may be different from the on-premises Active Directory account.

This step is run once per forest.

The `Register-AzureADPasswordProtectionForest` cmdlet supports the following three authentication modes. The first two modes support Microsoft Entra multifactor authentication but the third mode doesn't.

 **Tip**

There might be a noticeable delay before completion the first time that this cmdlet is run for a specific Azure tenant. Unless a failure is reported, don't worry about this delay.

- Interactive authentication mode:

PowerShell

```
Register-AzureADPasswordProtectionForest -AccountUpn  
'yourglobaladmin@yourtenant.onmicrosoft.com'
```

 **Note**

This mode won't work on Server Core operating systems. Instead use one of the following two authentication modes. Also, this mode might fail if Internet Explorer Enhanced Security Configuration is enabled. The workaround is to disable that Configuration, register the forest, and then re-enable it.

- Device-code authentication mode:

PowerShell

```
Register-AzureADPasswordProtectionForest -AccountUpn  
'yourglobaladmin@yourtenant.onmicrosoft.com' -  
AuthenticateUsingDeviceCode
```

When prompted, following the link to open a web browser and enter the authentication code.

- Silent (password-based) authentication mode:

PowerShell

```
$globalAdminCredentials = Get-Credential  
Register-AzureADPasswordProtectionForest -AzureCredential
```

```
$globalAdminCredentials
```

① Note

This mode fails if Microsoft Entra multifactor authentication is required for your account. In that case, use one of the previous two authentication modes, or instead use a different account that does not require MFA.

You may also see MFA required if Azure Device Registration (which is used under the covers by Microsoft Entra Password Protection) has been configured to globally require MFA. To workaround this requirement you may use a different account that supports MFA with one of the previous two authentication modes, or you may also temporarily relax the Azure Device Registration MFA requirement.

To make this change, select **Identity** in the [Microsoft Entra admin center](#), then select **Devices > Device Settings**. Set **Require multifactor authentication to join devices** to **No**. Be sure to reconfigure this setting back to **Yes** once registration is complete.

We recommend that MFA requirements be bypassed for test purposes only.

These examples only succeed if the currently signed-in user is also an Active Directory domain administrator for the root domain. If this isn't the case, you can supply alternative domain credentials via the *-ForestCredential* parameter.

Registration of the Active Directory forest is necessary only once in the lifetime of the forest. After that, the Microsoft Entra Password Protection DC agents in the forest automatically perform any other necessary maintenance. After `Register-AzureADPasswordProtectionForest` runs successfully for a forest, additional invocations of the cmdlet succeed, but are unnecessary.

For `Register-AzureADPasswordProtectionForest` to succeed, at least one DC running Windows Server 2012 or later must be available in the Microsoft Entra Password Protection proxy server's domain. The Microsoft Entra Password Protection DC agent software doesn't have to be installed on any domain controllers prior to this step.

7. To make sure that the changes have taken effect, run `Test-AzureADPasswordProtectionProxyHealth -TestAll`. For help resolving errors, see

Configure the proxy service to communicate through an HTTP proxy

If your environment requires the use of a specific HTTP proxy to communicate with Azure, use the following steps to configure the Microsoft Entra Password Protection service.

Create a *AzureADPasswordProtectionProxy.exe.config* file in the `%ProgramFiles%\Azure AD Password Protection Proxy\Service` folder. Include the following content:

XML

```
<configuration>
  <system.net>
    <defaultProxy enabled="true">
      <proxy bypassonlocal="true"
            proxyaddress="http://yourhttpproxy.com:8080" />
    </defaultProxy>
  </system.net>
</configuration>
```

If your HTTP proxy requires authentication, add the *useDefaultCredentials* tag:

XML

```
<configuration>
  <system.net>
    <defaultProxy enabled="true" useDefaultCredentials="true">
      <proxy bypassonlocal="true"
            proxyaddress="http://yourhttpproxy.com:8080" />
    </defaultProxy>
  </system.net>
</configuration>
```

In both cases, replace `http://yourhttpproxy.com:8080` with the address and port of your specific HTTP proxy server.

If your HTTP proxy is configured to use an authorization policy, you must grant access to the Active Directory computer account of the machine that hosts the proxy service for password protection.

We recommend that you stop and restart the Microsoft Entra Password Protection proxy service after you create or update the *AzureADPasswordProtectionProxy.exe.config* file.

The proxy service doesn't support the use of specific credentials for connecting to an HTTP proxy.

Configure the proxy service to listen on a specific port

The Microsoft Entra Password Protection DC agent software uses RPC over TCP to communicate with the proxy service. By default, the Microsoft Entra Password Protection proxy service listens on any available dynamic RPC endpoint. You can configure the service to listen on a specific TCP port, if necessary due to networking topology or firewall requirements in your environment. When you configure a static port, you must open port 135 and the static port of your choice.

To configure the service to run under a static port, use the `Set-AzureADPasswordProtectionProxyConfiguration` cmdlet as follows:

PowerShell

```
Set-AzureADPasswordProtectionProxyConfiguration -StaticPort <portnumber>
```

⚠️ Warning

You must stop and restart the Microsoft Entra Password Protection proxy service for these changes to take effect.

To configure the service to run under a dynamic port, use the same procedure but set *StaticPort* back to zero:

PowerShell

```
Set-AzureADPasswordProtectionProxyConfiguration -StaticPort 0
```

⚠️ Warning

You must stop and restart the Microsoft Entra Password Protection proxy service for these changes to take effect.

The Microsoft Entra Password Protection proxy service requires a manual restart after any change in port configuration. You don't have to restart the Microsoft Entra Password Protection DC agent service on domain controllers after you make these configuration changes.

To query for the current configuration of the service, use the `Get-AzureADPasswordProtectionProxyConfiguration` cmdlet as shown in the following example

PowerShell

```
Get-AzureADPasswordProtectionProxyConfiguration | fl
```

The following example output shows that the Microsoft Entra Password Protection proxy service is using a dynamic port:

Output

```
ServiceName : AzureADPasswordProtectionProxy
DisplayName : Azure AD password protection Proxy
StaticPort  : 0
```

Install the DC agent service

To install the Microsoft Entra Password Protection DC agent service, run the `AzureADPasswordProtectionDCAgentSetup.msi` package.

You can automate the software installation by using standard MSI procedures, as shown in the following example:

Console

```
msiexec.exe /i AzureADPasswordProtectionDCAgentSetup.msi /quiet /qn
/norestart
```

The `/norestart` flag can be omitted if you prefer to have the installer automatically reboot the machine.

The software installation, or uninstallation, requires a restart. This requirement is because password filter DLLs are only loaded or unloaded by a restart.

The installation of on-prem Microsoft Entra Password Protection is complete after the DC agent software is installed on a domain controller, and that computer is rebooted. No other configuration is required or possible. Password change events against the on-prem DCs use the configured banned password lists from Microsoft Entra ID.

To enable on-prem Microsoft Entra Password Protection or configure custom banned passwords, see [Enable on-premises Microsoft Entra Password Protection](#).

💡 Tip

You can install the Microsoft Entra Password Protection DC agent on a machine that's not yet a domain controller. In this case, the service starts and runs but remain inactive until the machine is promoted to be a domain controller.

Upgrading the proxy service

The Microsoft Entra Password Protection proxy service supports automatic upgrade. Automatic upgrade uses the Microsoft Entra Connect Agent Updater service, which is installed side by side with the proxy service. Automatic upgrade is on by default, and may be enabled or disabled using the `Set-`

`AzureADPasswordProtectionProxyConfiguration` cmdlet.

The current setting can be queried using the `Get-`

`AzureADPasswordProtectionProxyConfiguration` cmdlet. We recommend that the automatic upgrade setting always is enabled.

The `Get-AzureADPasswordProtectionProxy` cmdlet may be used to query the software version of all currently installed Microsoft Entra Password Protection proxy servers in a forest.

ⓘ Note

The proxy service will only automatically upgrade to a newer version when critical security patches are needed.

Manual upgrade process

A manual upgrade is accomplished by running the latest version of the `AzureADPasswordProtectionProxySetup.exe` software installer. The latest version of the software is available on the [Microsoft Download Center](#).

It's not required to uninstall the current version of the Microsoft Entra Password Protection proxy service - the installer performs an in-place upgrade. No reboot should be required when upgrading the proxy service. The software upgrade may be automated using standard MSI procedures, such as

`AzureADPasswordProtectionProxySetup.exe /quiet`.

Upgrading the DC agent

When a newer version of the Microsoft Entra Password Protection DC agent software is available, the upgrade is accomplished by running the latest version of the `AzureADPasswordProtectionDCAgentSetup.msi` software package. The latest version of the software is available on the [Microsoft Download Center](#).

It's not required to uninstall the current version of the DC agent software - the installer performs an in-place upgrade. A reboot is always required when upgrading the DC agent software - this requirement is caused by core Windows behavior.

The software upgrade may be automated using standard MSI procedures, such as

```
msiexec.exe /i AzureADPasswordProtectionDCAgentSetup.msi /quiet /qn /norestart.
```

You may omit the `/norestart` flag if you prefer to have the installer automatically reboot the machine.

The `Get-AzureADPasswordProtectionDCAgent` cmdlet may be used to query the software version of all currently installed Microsoft Entra Password Protection DC agents in a forest.

Next steps

Now that you've installed the services that you need for Microsoft Entra Password Protection on your on-premises servers, [enable on-prem Microsoft Entra Password Protection](#) to complete your deployment.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Enable on-premises Microsoft Entra Password Protection

Article • 03/04/2025

Users often create passwords that use common local words such as a school, sports team, or famous person. These passwords are easy to guess, and weak against dictionary-based attacks. To enforce strong passwords in your organization, Microsoft Entra Password Protection provides a global and custom banned password list. A password change request fails if there's a match in these banned password list.

To protect your on-premises Active Directory Domain Services (AD DS) environment, you can install and configure Microsoft Entra Password Protection to work with your on-premises DC. This article shows you how to enable Microsoft Entra Password Protection for your on-premises environment.

For more information on how Microsoft Entra Password Protection works in an on-premises environment, see [How to enforce Microsoft Entra Password Protection for Windows Server Active Directory](#).

Before you begin

This article shows you how to enable Microsoft Entra Password Protection for your on-premises environment. Before you complete this article, [install and register the Microsoft Entra Password Protection proxy service and DC agents](#) in your on-premises AD DS environment.

Enable on-premises password protection

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Administrator](#).
2. Browse to **Protection > Authentication methods > Password protection**.
3. Set the option for **Enable password protection on Windows Server Active Directory** to **Yes**.

When this setting is set to *No*, all deployed Microsoft Entra Password Protection DC agents go into a quiescent mode where all passwords are accepted as-is. No validation activities are performed, and audit events aren't generated.

4. It's recommended to initially set the **Mode** to *Audit*. After you're comfortable with the feature and the impact on users in your organization, you can switch the **Mode** to *Enforced*. For more information, see the following section on [modes of operation](#).

5. When ready, select **Save**.

The screenshot shows the Microsoft Entra ID Security interface for managing authentication methods. The left sidebar has sections for 'Manage' (Policies, Password protection, Registration campaign, Authentication strengths, Settings) and 'Monitoring' (Activity, User registration details, Registration and reset events, Bulk operation results). The main area is titled 'Authentication methods | Password protection'. Under 'Manage', 'Password protection' is selected and highlighted with a red box. It contains settings for 'Custom smart lockout' (Lockout threshold: 10, Lockout duration in seconds: 60), 'Custom banned passwords' (enforced custom list: Yes, list: contoso, fabrikam, tailwind, michigan, wolverine, harbaugh, howard), and 'Password protection for Windows Server Active Directory' (Enable password protection on Windows Server Active Directory: Yes, Mode: Audit). A 'Save' button is at the top right.

Modes of operation

When you enable on-premises Microsoft Entra Password Protection, you can use either *audit* mode or *enforce* mode. We recommend that initial deployment and testing always start out in audit mode. Entries in the event log should then be monitored to anticipate whether any existing operational processes would be disturbed once *Enforce* mode is enabled.

Audit mode

Audit mode is intended as a way to run the software in a "what if" mode. Each Microsoft Entra Password Protection DC agent service evaluates an incoming password according to the currently active policy.

If the current policy is configured to be in audit mode, "bad" passwords result in event log messages but are processed and updated. This behavior is the only difference between audit and enforce mode. All other operations run the same.

Enforced Mode

Enforced mode is intended as the final configuration. Like when in audit mode, each Microsoft Entra Password Protection DC agent service evaluates incoming passwords according to the currently active policy. When enforced mode is enabled though, a password that's considered insecure according to the policy is rejected.

When a password is rejected in enforced mode by the Microsoft Entra Password Protection DC agent, an end user sees a similar error like they would see if their password was rejected by traditional on-premises password complexity enforcement. For example, a user might see the following traditional error message at the Windows logon or change password screen:

"Unable to update the password. The value provided for the new password does not meet the length, complexity, or history requirements of the domain."

This message is only one example of several possible outcomes. The specific error message can vary depending on the actual software or scenario that is attempting to set an insecure password.

Affected end users may need to work with their IT staff to understand the new requirements and to choose secure passwords.

 **Note**

Microsoft Entra Password Protection has no control over the specific error message displayed by the client machine when a weak password is rejected.

Next steps

To customize the banned password list for your organization, see [Configure the Microsoft Entra Password Protection custom banned password list](#).

To monitor on-premises events, see [Monitoring on-premises Microsoft Entra Password Protection](#).

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Monitor and review logs for on-premises Microsoft Entra Password Protection environments

Article • 03/04/2025

After the deployment of Microsoft Entra Password Protection, monitoring and reporting are essential tasks. This article goes into detail to help you understand various monitoring techniques, including where each service logs information and how to report on the use of Microsoft Entra Password Protection.

Monitoring and reporting are done either by event log messages or by running PowerShell cmdlets. The DC agent and proxy services both log event log messages. All PowerShell cmdlets described below are only available on the proxy server (see the AzureADPasswordProtection PowerShell module). The DC agent software does not install a PowerShell module.

DC agent event logging

On each domain controller, the DC agent service software writes the results of each individual password validation operation (and other status) to a local event log:

```
\Applications and Services Logs\Microsoft\AzureADPasswordProtection\DCAgent\Admin  
\Applications and Services  
Logs\Microsoft\AzureADPasswordProtection\DCAgent\Operational  
\Applications and Services Logs\Microsoft\AzureADPasswordProtection\DCAgent\Trace
```

The DC agent Admin log is the primary source of information for how the software is behaving.

Note that the Trace log is off by default.

Events logged by the various DC agent components fall within the following ranges:

[+] Expand table

Component	Event ID range
DC Agent password filter dll	10000-19999

Component	Event ID range
DC agent service hosting process	20000-29999
DC agent service policy validation logic	30000-39999

DC agent Admin event log

Password validation outcome events

On each domain controller, the DC agent service software writes the results of each individual password validation to the DC agent admin event log.

For a successful password validation operation, there is generally one event logged from the DC agent password filter dll. For a failing password validation operation, there are generally two events logged, one from the DC agent service, and one from the DC Agent password filter dll.

Discrete events to capture these situations are logged, based around the following factors:

- Whether a given password is being set or changed.
- Whether validation of a given password passed or failed.
- Whether validation failed due to the Microsoft global policy, the organizational policy, or a combination.
- Whether audit only mode is currently on or off for the current password policy.

The key password-validation-related events are as follows:

[\[+\] Expand table](#)

Event	Password change	Password set
Pass	10014	10015
Fail (due to customer password policy)	10016, 30002	10017, 30003
Fail (due to Microsoft password policy)	10016, 30004	10017, 30005
Fail (due to combined Microsoft and customer password policies)	10016, 30026	10017, 30027

Event	Password change	Password set
Fail (due to user name)	10016, 30021	10017, 30022
Audit-only Pass (would have failed customer password policy)	10024, 30008	10025, 30007
Audit-only Pass (would have failed Microsoft password policy)	10024, 30010	10025, 30009
Audit-only Pass (would have failed combined Microsoft and customer password policies)	10024, 30028	10025, 30029
Audit-only Pass (would have failed due to user name)	10016, 30024	10017, 30023

The cases in the table above that refer to "combined policies" are referring to situations where a user's password was found to contain at least one token from both the Microsoft banned password list and the customer banned password list.

The cases in the table above that refer to "user name" are referring to situations where a user's password was found to contain either the user's account name and/or one of the user's friendly names. Either scenario will cause the user's password to be rejected when the policy is set to Enforce, or passed if the policy is in Audit mode.

When a pair of events is logged together, both events are explicitly associated by having the same CorrelationId.

Password validation summary reporting via PowerShell

The `Get-AzureADPasswordProtectionSummaryReport` cmdlet may be used to produce a summary view of password validation activity. An example output of this cmdlet is as follows:

PowerShell

```
Get-AzureADPasswordProtectionSummaryReport -DomainController bplrootdc2
DomainController : bplrootdc2
PasswordChangesValidated : 6677
PasswordSetsValidated : 9
PasswordChangesRejected : 10868
PasswordSetsRejected : 34
PasswordChangeAuditOnlyFailures : 213
PasswordSetAuditOnlyFailures : 3
```

PasswordChangeErrors	: 0
PasswordSetErrors	: 1

The scope of the cmdlet's reporting may be influenced using one of the `-Forest`, `-Domain`, or `-DomainController` parameters. Not specifying a parameter implies `-Forest`.

ⓘ Note

If you only install the DC agent on one DC, the `Get-AzureADPasswordProtectionSummaryReport` cmdlet will read events only from that DC. To get events from multiple DCs, you'll need the DC agent installed on each DC.

The `Get-AzureADPasswordProtectionSummaryReport` cmdlet works by querying the DC agent admin event log, and then counting the total number of events that correspond to each displayed outcome category. The following table contains the mappings between each outcome and its corresponding event ID:

[+] Expand table

Get-AzureADPasswordProtectionSummaryReport property	Corresponding event ID
>PasswordChangesValidated	10014
PasswordSetsValidated	10015
PasswordChangesRejected	10016
PasswordSetsRejected	10017
PasswordChangeAuditOnlyFailures	10024
PasswordSetAuditOnlyFailures	10025
PasswordChangeErrors	10012
PasswordSetErrors	10013

Note that the `Get-AzureADPasswordProtectionSummaryReport` cmdlet is shipped in PowerShell script form and if needed may be referenced directly at the following location:

```
%ProgramFiles%\WindowsPowerShell\Modules\AzureADPasswordProtection\Get-
AzureADPasswordProtectionSummaryReport.ps1
```

ⓘ Note

This cmdlet works by opening a PowerShell session to each domain controller. In order to succeed, PowerShell remote session support must be enabled on each domain controller, and the client must have sufficient privileges. For more information on PowerShell remote session requirements, run 'Get-Help about_Remote_Troubleshooting' in a PowerShell window.

Note

This cmdlet works by remotely querying each DC agent service's Admin event log. If the event logs contain large numbers of events, the cmdlet may take a long time to complete. In addition, bulk network queries of large data sets may impact domain controller performance. Therefore, this cmdlet should be used carefully in production environments.

Sample event log messages

Event ID 10014 (Successful password change)

text

The changed password for the specified user was validated as compliant with the current Azure password policy.

UserName: SomeUser
FullName: Some User

Event ID 10017 (Failed password change):

text

The reset password for the specified user was rejected because it did not comply with the current Azure password policy. Please see the correlated event log message for more details.

UserName: SomeUser
FullName: Some User

Event ID 30003 (Failed password change):

text

The reset password for the specified user was rejected because it matched at least one of the tokens present in the per-tenant banned password list of the current Azure password policy.

UserName: SomeUser
FullName: Some User

Event ID 10024 (Password accepted due to policy in audit only mode)

text

The changed password for the specified user would normally have been rejected because it did not comply with the current Azure password policy. The current Azure password policy is configured for audit-only mode so the password was accepted. Please see the correlated event log message for more details.

UserName: SomeUser
FullName: Some User

Event ID 30008 (Password accepted due to policy in audit only mode)

text

The changed password for the specified user would normally have been rejected because it matches at least one of the tokens present in the per-tenant banned password list of the current Azure password policy. The current Azure password policy is configured for audit-only mode so the password was accepted.

UserName: SomeUser
FullName: Some User

Event ID 30001 (Password accepted due to no policy available)

text

The password for the specified user was accepted because an Azure password policy is not available yet

UserName: SomeUser
FullName: Some User

This condition may be caused by one or more of the following reasons:%n

1. The forest has not yet been registered with Azure.

Resolution steps: an administrator must register the forest using the Register-AzureADPasswordProtectionForest cmdlet.

2. An Azure AD password protection Proxy is not yet available on at least one machine in the current forest.

Resolution steps: an administrator must install and register a proxy using the Register-AzureADPasswordProtectionProxy cmdlet.

3. This DC does not have network connectivity to any Azure AD password protection Proxy instances.

Resolution steps: ensure network connectivity exists to at least one Azure AD password protection Proxy instance.

4. This DC does not have connectivity to other domain controllers in the domain.

Resolution steps: ensure network connectivity exists to the domain.

Event ID 30006 (New policy being enforced)

text

The service is now enforcing the following Azure password policy.

Enabled: 1

AuditOnly: 1

Global policy date: 2018-05-15T00:00:00.000000000Z

Tenant policy date: 2018-06-10T20:15:24.432457600Z

Enforce tenant policy: 1

Event ID 30019 (Microsoft Entra Password Protection is disabled)

text

The most recently obtained Azure password policy was configured to be disabled. All passwords submitted for validation from this point on will automatically be considered compliant with no processing performed.

No further events will be logged until the policy is changed.%n

DC Agent Operational log

The DC agent service will also log operational-related events to the following log:

```
\Applications and Services  
Logs\Microsoft\AzureADPasswordProtection\DCAgent\Operational
```

DC Agent Trace log

The DC agent service can also log verbose debug-level trace events to the following log:

```
\Applications and Services Logs\Microsoft\AzureADPasswordProtection\DCAgent\Trace
```

Trace logging is disabled by default.

Warning

When enabled, the Trace log receives a high volume of events and may impact domain controller performance. Therefore, this enhanced log should only be enabled when a problem requires deeper investigation, and then only for a minimal amount of time.

DC Agent text logging

The DC agent service can be configured to write to a text log by setting the following registry value:

```
text  
  
HKLM\System\CurrentControlSet\Services\AzureADPasswordProtectionDCAgent\Parameters!EnableTextLogging = 1 (REG_DWORD value)
```

Text logging is disabled by default. A restart of the DC agent service is required for changes to this value to take effect. When enabled the DC agent service will write to a log file located under:

```
%ProgramFiles%\Azure AD Password Protection DC Agent\Logs
```

Tip

The text log receives the same debug-level entries that can be logged to the Trace log, but is generally in an easier format to review and analyze.

⚠️ Warning

When enabled, this log receives a high volume of events and may impact domain controller performance. Therefore, this enhanced log should only be enabled when a problem requires deeper investigation, and then only for a minimal amount of time.

DC agent performance monitoring

The DC agent service software installs a performance counter object named **Microsoft Entra Password Protection**. The following perf counters are currently available:

[+] Expand table

Perf counter name	Description
Passwords processed	This counter displays the total number of passwords processed (accepted or rejected) since last restart.
Passwords accepted	This counter displays the total number of passwords that were accepted since last restart.
Passwords rejected	This counter displays the total number of passwords that were rejected since last restart.
Password filter requests in progress	This counter displays the number of password filter requests currently in progress.
Peak password filter requests	This counter displays the peak number of concurrent password filter requests since the last restart.
Password filter request errors	This counter displays the total number of password filter requests that failed due to an error since last restart. Errors can occur when the Microsoft Entra Password Protection DC agent service is not running.
Password filter requests/sec	This counter displays the rate at which passwords are being processed.
Password filter request processing time	This counter displays the average time required to process a password filter request.

Perf counter name	Description
Peak password filter request processing time	This counter displays the peak password filter request processing time since the last restart.
Passwords accepted due to audit mode	This counter displays the total number of passwords that would normally have been rejected, but were accepted because the password policy was configured to be in audit-mode (since last restart).

DC Agent discovery

The `Get-AzureADPasswordProtectionDCAgent` cmdlet may be used to display basic information about the various DC agents running in a domain or forest. This information is retrieved from the `serviceConnectionPoint` object(s) registered by the running DC agent service(s).

An example output of this cmdlet is as follows:

```
PowerShell

Get-AzureADPasswordProtectionDCAgent
ServerFQDN      : bplChildDC2.bplchild.bplRootDomain.com
Domain          : bplchild.bplRootDomain.com
Forest          : bplRootDomain.com
PasswordPolicyDateUTC : 2/16/2018 8:35:01 AM
HeartbeatUTC    : 2/16/2018 8:35:02 AM
```

The various properties are updated by each DC agent service on an approximate hourly basis. The data is still subject to Active Directory replication latency.

The scope of the cmdlet's query may be influenced using either the `-Forest` or `-Domain` parameters.

If the `HeartbeatUTC` value gets stale, this may be a symptom that the Microsoft Entra Password Protection DC Agent on that domain controller is not running, or has been uninstalled, or the machine was demoted and is no longer a domain controller.

If the `PasswordPolicyDateUTC` value gets stale, this may be a symptom that the Microsoft Entra Password Protection DC Agent on that machine is not working properly.

DC agent newer version available

The DC agent service will log a 30034 warning event to the Operational log upon detecting that a newer version of the DC agent software is available, for example:

text

An update for Azure AD Password Protection DC Agent is available.

If autoupgrade is enabled, this message may be ignored.

If autoupgrade is disabled, refer to the following link for the latest version available:

<https://aka.ms/AzureADPasswordProtectionAgentSoftwareVersions>

Current version: 1.2.116.0

The event above does not specify the version of the newer software. You should go to the link in the event message for that information.

(!) Note

Despite the references to "autoupgrade" in the above event message, the DC agent software does not currently support this feature.

Proxy service event logging

The Proxy service emits a minimal set of events to the following event logs:

\Applications and Services

Logs\Microsoft\AzureADPasswordProtection\ProxyService\Admin

\Applications and Services

Logs\Microsoft\AzureADPasswordProtection\ProxyService\Operational

\Applications and Services

Logs\Microsoft\AzureADPasswordProtection\ProxyService\Trace

Note that the Trace log is off by default.

⚠ Warning

When enabled, the Trace log receives a high volume of events and this may impact performance of the proxy host. Therefore, this log should only be enabled when a

problem requires deeper investigation, and then only for a minimal amount of time.

Events are logged by the various Proxy components using the following ranges:

[+] Expand table

Component	Event ID range
Proxy service hosting process	10000-19999
Proxy service core business logic	20000-29999
PowerShell cmdlets	30000-39999

Proxy service text logging

The Proxy service can be configured to write to a text log by setting the following registry value:

HKLM\System\CurrentControlSet\Services\AzureADPasswordProtectionProxy\Parameters!EnableTextLogging = 1 (REG_DWORD value)

Text logging is disabled by default. A restart of the Proxy service is required for changes to this value to take effect. When enabled the Proxy service will write to a log file located under:

%ProgramFiles%\Azure AD Password Protection Proxy\Logs

Tip

The text log receives the same debug-level entries that can be logged to the Trace log, but is generally in an easier format to review and analyze.

Warning

When enabled, this log receives a high volume of events and may impact the machine's performance. Therefore, this enhanced log should only be enabled when a problem requires deeper investigation, and then only for a minimal amount of time.

PowerShell cmdlet logging

PowerShell cmdlets that result in a state change (for example, Register-AzureADPasswordProtectionProxy) will normally log an outcome event to the Operational log.

In addition, most of the Microsoft Entra Password Protection PowerShell cmdlets will write to a text log located under:

```
%ProgramFiles%\Azure AD Password Protection Proxy\Logs
```

If a cmdlet error occurs and the cause and\or solution is not readily apparent, these text logs may also be consulted.

Proxy discovery

The `Get-AzureADPasswordProtectionProxy` cmdlet may be used to display basic information about the various Microsoft Entra Password Protection Proxy services running in a domain or forest. This information is retrieved from the serviceConnectionPoint object(s) registered by the running Proxy service(s).

An example output of this cmdlet is as follows:

```
PowerShell

Get-AzureADPasswordProtectionProxy
ServerFQDN      : bplProxy.bplchild2.bplRootDomain.com
Domain          : bplchild2.bplRootDomain.com
Forest          : bplRootDomain.com
HeartbeatUTC    : 12/25/2018 6:35:02 AM
```

The various properties are updated by each Proxy service on an approximate hourly basis. The data is still subject to Active Directory replication latency.

The scope of the cmdlet's query may be influenced using either the –Forest or –Domain parameters.

If the HeartbeatUTC value gets stale, this may be a symptom that the Microsoft Entra Password Protection Proxy on that machine is not running or has been uninstalled.

Proxy agent newer version available

The Proxy service will log a 20002 warning event to the Operational log upon detecting that a newer version of the proxy software is available, for example:

```
text
```

An update for Azure AD Password Protection Proxy is available.

If autoupgrade is enabled, this message may be ignored.

If autoupgrade is disabled, refer to the following link for the latest version available:

<https://aka.ms/AzureADPasswordProtectionAgentSoftwareVersions>

Current version: 1.2.116.0

.

The event above does not specify the version of the newer software. You should go to the link in the event message for that information.

This event will be emitted even if the Proxy agent is configured with autoupgrade enabled.

Next steps

[Troubleshooting for Microsoft Entra Password Protection](#)

For more information on the global and custom banned password lists, see the article [Ban bad passwords](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Troubleshoot: On-premises Microsoft Entra Password Protection

Article • 03/04/2025

After the deployment of Microsoft Entra Password Protection, troubleshooting may be required. This article goes into detail to help you understand some common troubleshooting steps.

The DC agent can't locate a proxy in the directory

The main symptom of this problem is 30,017 events in the DC agent Admin event log.

The usual cause of this issue is that a proxy hasn't been registered. If a proxy is registered, there may be some delay due to AD replication latency until a particular DC agent is able to see that proxy.

The DC agent isn't able to communicate with a proxy

The main symptom of this problem is 30,018 events in the DC agent Admin event log.

This problem may have several possible causes:

- The DC agent is located in an isolated portion of the network that doesn't allow network connectivity to the registered proxy(s). This problem may be benign as long as other DC agents can communicate with the proxy(s) in order to download password policies from Azure. Once downloaded, those policies are obtained by the isolated DC via replication of the policy files in the sysvol share.
- The proxy host machine is blocking access to the RPC endpoint mapper endpoint (port 135)

The Microsoft Entra Password Protection Proxy installer automatically creates a Windows Firewall inbound rule that allows access to port 135. If this rule is later deleted or disabled, DC agents are unable to communicate with the Proxy service. If the builtin Windows Firewall is disabled in lieu of another firewall product, you must configure that firewall to allow access to port 135.

- The proxy host machine is blocking access to the RPC endpoint (dynamic or static) listened on by the Proxy service

The Microsoft Entra Password Protection Proxy installer automatically creates a Windows Firewall inbound rule that allows access to any inbound ports listened to by the Microsoft Entra Password Protection Proxy service. If this rule is later deleted or disabled, DC agents are unable to communicate with the Proxy service. If the builtin Windows Firewall is disabled in lieu of another firewall product, you must configure that firewall to allow access to any inbound ports listened to by the Microsoft Entra Password Protection Proxy service. This configuration may be made more specific if the Proxy service is configured to listen on a specific static RPC port (using the `Set-AzureADPasswordProtectionProxyConfiguration` cmdlet).

- The proxy host machine isn't configured to allow domain controllers the ability to sign-in to the machine. This behavior is controlled via the "Access this computer from the network" user privilege assignment. All domain controllers in all domains in the forest must be granted this privilege. This setting is often constrained as part of a larger network hardening effort.

Proxy service is unable to communicate with Azure

1. Ensure the proxy machine has connectivity to the endpoints listed in the [deployment requirements](#).
2. Ensure that the forest and all proxy servers are registered against the same Azure tenant.

You can check this requirement by running the `Get-AzureADPasswordProtectionProxy` and `Get-AzureADPasswordProtectionDCAgent` PowerShell cmdlets, then compare the `AzureTenant` property of each returned item. For correct operation, the reported tenant name must be the same across all DC agents and proxy servers.

If an Azure tenant registration mismatch condition does exist, this problem can be fixed by running the `Register-AzureADPasswordProtectionProxy` and/or `Register-AzureADPasswordProtectionForest` PowerShell cmdlets as needed, making sure to use credentials from the same Azure tenant for all registrations.

DC agent is unable to encrypt or decrypt password policy files

Microsoft Entra Password Protection has a critical dependency on the encryption and decryption functionality supplied by the Microsoft Key Distribution Service. Encryption or decryption failures can manifest with various symptoms and have several potential causes.

- Ensure that the KDS service is enabled and functional on all Windows Server 2012 and later domain controllers in a domain.

By default the KDS service's service start mode is configured as Manual (Trigger Start). This configuration means that the first time a client tries to use the service, it's started on-demand. This default service start mode is acceptable for Microsoft Entra Password Protection to work.

If the KDS service start mode is configured to Disabled, this configuration must be fixed before Microsoft Entra Password Protection can function properly.

A simple test for this issue is to manually start the KDS service, either via the Service Management MMC console, or using other management tools (for example, run "net start kdssvc" from a command prompt console). The KDS service is expected to start successfully and stay running.

The most common root cause for the KDS service being unable to start is that the Active Directory domain controller object is located outside of the default Domain Controllers OU. This configuration isn't supported by the KDS service and isn't a limitation imposed by Microsoft Entra Password Protection. The fix for this condition is to move the domain controller object to a location under the default Domain Controllers OU.

- Incompatible KDS encrypted buffer format change from Windows Server 2012 R2 to Windows Server 2016

A KDS security fix was introduced in Windows Server 2016 that modifies the format of KDS encrypted buffers. These buffers sometimes fail to decrypt on Windows Server 2012 and Windows Server 2012 R2. The reverse direction is okay. Buffers that are KDS-encrypted on Windows Server 2012 and Windows Server 2012 R2 always successfully decrypt on Windows Server 2016 and later. If the domain controllers in your Active Directory domains are running a mix of these operating systems, occasional Microsoft Entra Password Protection decryption failures may be reported. It isn't possible to accurately predict the timing or symptoms of these failures given the nature of the security fix. Also, given that it's non-deterministic

which Microsoft Entra Password Protection DC Agent on which domain controller encrypts data at a given time.

There's no workaround for this issue other than to not run a mix of these incompatible operating systems in your Active Directory domain(s). In other words, you should run only Windows Server 2012 and Windows Server 2012 R2 domain controllers, OR you should only run Windows Server 2016 and above domain controllers.

DC agent thinks the forest hasn't been registered

The symptom of this issue is 30,016 events getting logged in the DC Agent\Admin channel that says in part:

text

The forest hasn't been registered with Azure. Password policies can't be downloaded from Azure unless this is corrected.

There are two possible causes for this issue.

- The forest hasn't been registered. To resolve the problem, run the Register-AzureADPasswordProtectionForest command as described in the [deployment requirements](#).
- The forest is registered, but the DC agent is unable to decrypt the forest registration data. This case has the same root cause as issue #2 listed under [DC agent is unable to encrypt or decrypt password policy files](#). An easy way to confirm this theory is that you'll see this error only on DC agents running on Windows Server 2012 or Windows Server 2012R2 domain controllers, while DC agents running on Windows Server 2016 and later domain controllers are fine. The workaround is the same: upgrade all domain controllers to Windows Server 2016 or later.

Weak passwords are being accepted but should not be

This problem may have several causes.

- Your DC agent(s) are running a public preview software version that's expired. See [Public preview DC agent software has expired](#).

- Your DC agent(s) can't download a policy or is unable to decrypt existing policies. Check for possible causes in the prior articles.
- The password policy Enforce mode is still set to Audit. If this configuration is in effect, reconfigure it to Enforce using the Microsoft Entra Password Protection portal. For more information, see [Modes of operation](#).
- The password policy is disabled. If this configuration is in effect, reconfigure it to enabled using the Microsoft Entra Password Protection portal. For more information, see [Modes of operation](#).
- You haven't installed the DC agent software on all domain controllers in the domain. In this situation, it's difficult to ensure that remote Windows clients target a particular domain controller during a password change operation. If you think you successfully targeted a particular DC where the DC agent software is installed, you can verify by double-checking the DC agent admin event log: regardless of outcome, there is at least one event to document the outcome of the password validation. If there's no event present for the user whose password is changed, then the password change was likely processed by a different domain controller.

As an alternative test, try setting\changing passwords while logged in directly to a DC where the DC agent software is installed. This technique isn't recommended for production Active Directory domains.

While incremental deployment of the DC agent software is supported subject to these limitations, Microsoft strongly recommends that the DC agent software is installed on all domain controllers in a domain as soon as possible.

- The password validation algorithm may actually be working as expected. See [How are passwords evaluated](#).

Ntdsutil.exe fails to set a weak DSRM password

Active Directory always validates a new Directory Services Repair Mode password to make sure it meets the domain's password complexity requirements; this validation also calls into password filter dlls like Microsoft Entra Password Protection. If the new DSRM password is rejected, the following error message results:

text

```
C:\>ntdsutil.exe
ntdsutil: set dsrm password
Reset DSRM Administrator Password: reset password on server null
Please type password for DS Restore Mode Administrator Account: *****
```

```
Please confirm new password: *****
Setting password failed.
    WIN32 Error Code: 0xa91
    Error Message: Password doesn't meet the requirements of the filter
dll's
```

When Microsoft Entra Password Protection logs the password validation event log event(s) for an Active Directory DSRM password, it's expected that the event log messages won't include a user name. This behavior occurs because the DSRM account is a local account that isn't part of the actual Active Directory domain.

Domain controller replica promotion fails because of a weak DSRM password

During the DC promotion process, the new Directory Services Repair Mode password is submitted to an existing DC in the domain for validation. If the new DSRM password is rejected, the following error message results:

PowerShell

```
Install-ADDSDomainController : Verification of prerequisites for Domain
Controller promotion failed. The Directory Services Restore Mode password
doesn't meet a requirement of the password filter(s). Supply a suitable
password.
```

Just like in the previous issue, any Microsoft Entra Password Protection password validation outcome event will have empty user names for this scenario.

Domain controller demotion fails due to a weak local Administrator password

It's supported to demote a domain controller that is still running the DC agent software. Administrators should be aware however that the DC agent software continues to enforce the current password policy during the demotion procedure. The new local Administrator account password (specified as part of the demotion operation) is validated like any other password. Microsoft recommends that secure passwords be chosen for local Administrator accounts as part of a DC demotion procedure.

Once the demotion is successful, and the domain controller is rebooted and is again running as a normal member server, the DC agent software reverts to running in a passive mode. It may then be uninstalled at any time.

Booting into Directory Services Repair Mode

If the domain controller is booted into Directory Services Repair Mode, the DC agent password filter dll detects this condition and causes all password validation or enforcement activities to be disabled, regardless of the currently active policy configuration. The DC agent password filter dll logs a 10023 warning event to the Admin event log, for example:

text

The password filter dll is loaded but the machine appears to be a domain controller that is booted into Directory Services Repair Mode. All password change and set requests are automatically approved. No further messages are logged until after the next reboot.

Public preview DC agent software has expired

During the Microsoft Entra Password Protection public preview period, the DC agent software was hard-coded to stop processing password validation requests on the following dates:

- Version 1.2.65.0 stopped processing password validation requests on September 1 2019.
- Version 1.2.25.0 and prior stopped processing password validation requests on July 1 2019.

As the deadline approaches, all time-limited DC agent versions emit a 10021 event in the DC agent Admin event log at boot time that looks like this:

text

The password filter dll has successfully loaded and initialized.

The allowable trial period is nearing expiration. Once the trial period has expired, the password filter dll no longer processes passwords. Please contact Microsoft for a newer supported version of the software.

Expiration date: 9/01/2019 0:00:00 AM

This message won't be repeated until the next reboot.

Once the deadline has passed, all time-limited DC agent versions emit a 10022 event in the DC agent Admin event log at boot time that looks like this:

text

The password filter dll is loaded but the allowable trial period has expired. All password change and set requests are automatically approved. Please contact Microsoft for a newer supported version of the software.

No further messages are logged until after the next reboot.

Since the deadline is only checked on initial boot, you may not see these events until long after the calendar deadline has passed. Once the deadline is recognized, no negative effects on either the domain controller or the larger environment occur other than all passwords are automatically approved.

Important

Microsoft recommends that expired public preview DC agents be immediately upgraded to the latest version.

An easy way to discover DC agents in your environment that need to be upgrade is by running the `Get-AzureADPasswordProtectionDCAgent` cmdlet, for example:

PowerShell

```
PS C:\> Get-AzureADPasswordProtectionDCAgent

ServerFQDN      : bpl1.bpl.com
SoftwareVersion : 1.2.125.0
Domain          : bpl.com
Forest          : bpl.com
PasswordPolicyDateUTC : 8/1/2019 9:18:05 PM
HeartbeatUTC    : 8/1/2019 10:00:00 PM
AzureTenant     : bpltest.onmicrosoft.com
```

For this article, the `SoftwareVersion` field is obviously the key property to look at. You can also use PowerShell filtering to filter out DC agents that are already at or above the required baseline version, for example:

PowerShell

```
PS C:\> $LatestAzureADPasswordProtectionVersion = "1.2.125.0"
PS C:\> Get-AzureADPasswordProtectionDCAgent | Where-Object
{$_ .SoftwareVersion -lt $LatestAzureADPasswordProtectionVersion}
```

The Microsoft Entra Password Protection Proxy software isn't time-limited in any version. Microsoft still recommends that both DC and proxy agents be upgraded to the latest

versions as they're released. The `Get-AzureADPasswordProtectionProxy` cmdlet may be used to find Proxy agents that require upgrades, similar to the example above for DC agents.

Refer to [Upgrading the DC agent](#) and [Upgrading the Proxy service](#) for more details on specific upgrade procedures.

Emergency remediation

If a situation occurs where the DC agent service is causing problems, the DC agent service may be immediately shut down. The DC agent password filter dll still attempts to call the non-running service and logs warning events (10012, 10013), but all incoming passwords are accepted during that time. The DC agent service may then also be configured via the Windows Service Control Manager with a startup type of "Disabled" as needed.

Another remediation measure would be to set the Enable mode to No in the Microsoft Entra Password Protection portal. Once the updated policy is downloaded, each DC agent service shifts into a quiescent mode where all passwords are accepted as-is. For more information, see [Modes of operation](#).

Removal

If you decide to uninstall the Microsoft Entra password protection software and cleanup all related state from the domain(s) and forest, this task can be accomplished using the following steps:

 **Important**

It's important to perform these steps in order. If any instance of the Proxy service is left running, it periodically re-creates its serviceConnectionPoint object. If any instance of the DC agent service is left running, it periodically re-creates its serviceConnectionPoint object and the sysvol state.

1. Uninstall the Proxy software from all machines. This step does **not** require a reboot.
2. Uninstall the DC Agent software from all domain controllers. This step **requires** a reboot.
3. Manually remove all Proxy service connection points in each domain naming context. The location of these objects may be discovered with the following Active

Directory PowerShell command:

PowerShell

```
$scp = "serviceConnectionPoint"
$keywords = "{ebefb703-6113-413d-9167-9f8dd4d24468}*"*
Get-ADObject -SearchScope Subtree -Filter { objectClass -eq $scp -and
keywords -like $keywords }
```

Don't omit the asterisk ("*") at the end of the \$keywords variable value.

The resulting object(s) found via the `Get-ADObject` command can then be piped to `Remove-ADObject`, or deleted manually.

4. Manually remove all DC agent connection points in each domain naming context.

There may be one or more objects per domain controller in the forest, depending on how widely the software was deployed. The location of that object may be discovered with the following Active Directory PowerShell command:

PowerShell

```
$scp = "serviceConnectionPoint"
$keywords = "{2bac71e6-a293-4d5b-ba3b-50b995237946}*"*
Get-ADObject -SearchScope Subtree -Filter { objectClass -eq $scp -and
keywords -like $keywords }
```

The resulting object(s) found via the `Get-ADObject` command can then be piped to `Remove-ADObject`, or deleted manually.

Don't omit the asterisk ("*") at the end of the \$keywords variable value.

5. Manually remove the forest-level configuration state. The forest configuration state is maintained in a container in the Active Directory configuration naming context. It can be discovered and deleted as follows:

PowerShell

```
$passwordProtectionConfigContainer = "CN=Azure AD Password
Protection,CN=Services," + (Get-ADRootDSE).configurationNamingContext
Remove-ADObject -Recursive $passwordProtectionConfigContainer
```

6. Manually remove all sysvol related state by manually deleting the following folder and all of its contents:

```
\\\sysvol\\AzureADPasswordProtection
```

If necessary, this path may also be accessed locally on a given domain controller; the default location would be something like the following path:

```
%windir%\sysvol\domain\Policies\AzureADPasswordProtection
```

This path is different if the sysvol share is configured in a non-default location.

Health testing with PowerShell cmdlets

The AzureADPasswordProtection PowerShell module includes two health-related cmdlets that perform basic verification that the software is installed and working. It's a good idea to run these cmdlets after setting up a new deployment, periodically thereafter, and when a problem is being investigated.

Each individual health test returns a basic Passed or Failed result, plus an optional message on failure. In cases where the cause of a failure isn't clear, look for error event log messages that may explain the failure. Enabling text-log messages may also be useful. For more details, see [Monitor Microsoft Entra Password Protection](#).

Proxy health testing

The `Test-AzureADPasswordProtectionProxyHealth` cmdlet supports two health tests that can be run individually. A third mode allows for the running of all tests that don't require any parameter input.

Proxy registration verification

This test verifies that the Proxy agent is properly registered with Azure and is able to authenticate to Azure. A successful run looks like this:

```
PowerShell

PS C:\> Test-AzureADPasswordProtectionProxyHealth -VerifyProxyRegistration

DiagnosticName      Result AdditionalInfo
-----            -----
VerifyProxyRegistration  Passed
```

If an error is detected, the test returns a Failed result and an optional error message. Here's an example of one possible failure:

```
PowerShell
```

```
PS C:\> Test-AzureADPasswordProtectionProxyHealth -VerifyProxyRegistration
```

DiagnosticName	Result	AdditionalInfo
VerifyProxyRegistration	Failed	No proxy certificates were found - please run the Register-AzureADPasswordProtectionProxy cmdlet to register the proxy.

Proxy verification of end-to-end Azure connectivity

This test is a superset of the -VerifyProxyRegistration test. It requires that the Proxy agent is properly registered with Azure, is able to authenticate to Azure, and finally adds a check that a message can be successfully sent to Azure thus verifying full end-to-end communication is working.

A successful run looks like this:

PowerShell

```
PS C:\> Test-AzureADPasswordProtectionProxyHealth -VerifyAzureConnectivity
```

DiagnosticName	Result	AdditionalInfo
VerifyAzureConnectivity	Passed	

Proxy verification of all tests

This mode allows for the bulk running of all tests supported by the cmdlet that don't require parameter input. A successful run looks like this:

PowerShell

```
PS C:\> Test-AzureADPasswordProtectionProxyHealth -TestAll
```

DiagnosticName	Result	AdditionalInfo
VerifyTLSConfiguration	Passed	
VerifyProxyRegistration	Passed	
VerifyAzureConnectivity	Passed	

DC Agent health testing

The `Test-AzureADPasswordProtectionDCAgentHealth` cmdlet supports several health tests that can be run individually. A third mode allows for the running of all tests that

don't require any parameter input.

Basic DC agent health tests

The following tests can all be run individually and don't accept parameters. A brief description of each test is listed in the following table.

[\[+\] Expand table](#)

DC agent health test	Description
-VerifyPasswordFilterDll	Verifies that the password filter dll is currently loaded and is able to call the DC agent service
-VerifyForestRegistration	Verifies that the forest is currently registered
-VerifyEncryptionDecryption	Verifies that basic encryption and decryption is working using the Microsoft KDS service
-VerifyDomainIsUsingDFSR	Verifies that the current domain is using DFSR for sysvol replication
-VerifyAzureConnectivity	Verifies end-to-end communication with Azure is working using any available proxy

What follows is an example of the -VerifyPasswordFilterDll test passing, and other successful tests look similar:

PowerShell

```
PS C:\> Test-AzureADPasswordProtectionDCAgentHealth -VerifyPasswordFilterDll

DiagnosticName      Result AdditionalInfo
-----            -----
VerifyPasswordFilterDll Passed
```

DC agent verification of all tests

This mode allows for the bulk running of all tests supported by the cmdlet that don't require parameter input. A successful run looks like this:

PowerShell

```
PS C:\> Test-AzureADPasswordProtectionDCAgentHealth -TestAll

DiagnosticName      Result AdditionalInfo
-----            -----
VerifyPasswordFilterDll Passed
```

```
VerifyForestRegistration Passed
VerifyEncryptionDecryption Passed
VerifyDomainIsUsingDFSR Passed
VerifyAzureConnectivity Passed
```

Connectivity testing using specific proxy servers

Many troubleshooting situations involve investigating network connectivity between DC agents and proxies. There are two health tests available to focus on such issues specifically. These tests require that a particular proxy server be specified.

Verifying connectivity between a DC agent and a specific proxy

This test validates connectivity over the first communication leg from the DC agent to the proxy. It verifies that the proxy receives the call, however no communication with Azure is involved. A successful run looks like this:

PowerShell

```
PS C:\> Test-AzureADPasswordProtectionDCAgentHealth -VerifyProxyConnectivity
bp12.bpl.com

DiagnosticName      Result AdditionalInfo
-----            -----
VerifyProxyConnectivity  Passed
```

Here's an example failure condition where the proxy service running on the target server is stopped:

PowerShell

```
PS C:\> Test-AzureADPasswordProtectionDCAgentHealth -VerifyProxyConnectivity
bp12.bpl.com

DiagnosticName      Result AdditionalInfo
-----            -----
VerifyProxyConnectivity  Failed The RPC endpoint mapper on the specified
proxy returned no results; please check that the proxy service is running on
that server.
```

Verifying connectivity between a DC agent and Azure (using a specific proxy)

This test validates full end-to-end connectivity between a DC agent and Azure using a specific proxy. A successful run looks like this:

```
PowerShell

PS C:\> Test-AzureADPasswordProtectionDCAgentHealth -  
VerifyAzureConnectivityViaSpecificProxy bp12.bpl.com

DiagnosticName          Result AdditionalInfo  
-----  
VerifyAzureConnectivityViaSpecificProxy Passed
```

Next steps

[Frequently asked questions for Microsoft Entra Password Protection](#)

For more information on the global and custom banned password lists, see the article [Ban bad passwords](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Microsoft Entra Password Protection on-premises frequently asked questions

FAQ

This section provides answers to many commonly asked questions about Microsoft Entra Password Protection.

General questions

What guidance should users be given on how to select a secure password?

Microsoft's current guidance on this topic can be found at the following link:

[Microsoft Password Guidance ↗](#)

Is on-premises Microsoft Entra Password Protection supported in non-public clouds?

On-premises Microsoft Entra Password Protection is supported in both Azure Global and Azure Government clouds.

The Microsoft Entra admin center allows modification of the on-premises-specific "Password protection for Windows Server Active Directory" configuration in non-supported clouds; such changes persist but never take effect. Registration of on-premises proxy agents or forests is unsupported in non-supported clouds, and any such registration attempts always fail.

How can I apply Microsoft Entra Password Protection benefits to a subset of my on-premises users?

Not supported. Once deployed and enabled, Microsoft Entra Password Protection applies equally to all users.

What is the difference between a password change and a password set (or reset)?

A password change is when a user chooses a new password after proving they have knowledge of the old password. For example, a password change is what happens when a user logs into Windows and is then prompted to choose a new password.

A password set (sometimes called a password reset) is when an administrator replaces the password on an account with a new password, for example by using the Active Directory Users and Computers management tool. This operation requires a high level of privilege (usually Domain Admin), and the person performing the operation usually doesn't have knowledge of the old password. Help-desk scenarios often perform password sets, for instance when assisting a user who forgot their password. You'll also see password set events when a brand new user account is created for the first time with a password.

The password validation policy behaves the same regardless of whether a password change or set is being done. The Microsoft Entra Password Protection DC Agent service does log different events to inform you whether a password change or set operation was done. See [Microsoft Entra Password Protection monitoring and logging](#).

Does Microsoft Entra Password Protection validate existing passwords after being installed?

No - Microsoft Entra Password Protection can only enforce password policy on cleartext passwords during a password change or set operation. Once Active Directory accepts a password, only authentication-protocol-specific hashes of that password are persisted. The clear-text password is never persisted, therefore Microsoft Entra Password Protection can't validate existing passwords.

After initial deployment of Microsoft Entra Password Protection, all users and accounts will eventually start using a Microsoft Entra Password Protection-validated password as their existing passwords expire normally over time. If desired, you can accelerate this process by a one-time manual expiration of user account passwords.

Accounts configured with "password never expires" aren't forced to change their password unless manual expiration is done.

Why are duplicated password rejection events logged when attempting to set a weak password using the Active Directory Users and Computers management snap-in?

The Active Directory Users and Computers management snap-in first attempts to set the new password using the Kerberos protocol. Upon failure, the snap-in makes a second attempt to set the password using a legacy (SAM RPC) protocol. The specific protocols used aren't important. If the new password is considered weak by Microsoft Entra Password Protection, this snap-in behavior produces two sets of password reset rejection events being logged.

Why are Microsoft Entra Password Protection password validation events being logged with an empty user name?

Active Directory supports the ability to test a password to see if it passes the domain's current password complexity requirements, for example using the [NetValidatePasswordPolicy](#) api. When a password is validated in this way, the testing also includes validation by password-filter-dll based products such as Microsoft Entra Password Protection, but the user names passed to a given password filter dll are empty. In this scenario, Microsoft Entra Password Protection still validates the password using the currently in-effect password policy and issues an event log message to capture the outcome. However, the event log message will have empty user name fields.

I have hybrid users who attempt to change their password in Microsoft Entra ID and receive the response "We've seen that password too many times before. Choose something harder to guess." In this case, why don't I see a validation attempt on-premises?

When a hybrid user changes their password in Microsoft Entra ID, whether through Microsoft Entra SSPR, MyAccount, or another Microsoft Entra password change mechanism, their password is evaluated against the global and custom banned password lists in the cloud. When the password reaches Active Directory through password-writeback, it's already validated in Microsoft Entra ID.

Password resets and changes initiated in Microsoft Entra ID that fail validation for hybrid users can be found in the Microsoft Entra audit logs. See [Troubleshoot self-service password reset in Microsoft Entra ID](#).

Is it supported to install Microsoft Entra Password Protection side by side with other password-filter-based products?

Yes. Support for multiple registered password filter dlls is a core Windows feature and not specific to Microsoft Entra Password Protection. All registered password filter dlls must agree before a password is accepted.

How can I deploy and configure Microsoft Entra Password Protection in my Active Directory environment without using Azure?

Not supported. Microsoft Entra Password Protection is an Azure feature that supports being extended into an on-premises Active Directory environment.

How can I modify the contents of the policy at the Active Directory level?

Not supported. The policy can only be administered using the Microsoft Entra admin center. Also see the previous question.

Why is DFSR required for sysvol replication?

FRS (the predecessor technology to DFSR) has many known problems and is entirely unsupported in newer versions of Windows Server Active Directory. Zero testing of Microsoft Entra Password Protection is done on FRS-configured domains.

For more information, please see the following articles:

[The Case for Migrating sysvol replication to DFSR](#)

[The End is Nigh for FRS ↗](#)

If your domain isn't already using DFSR, you MUST migrate it to use DFSR before installing Microsoft Entra Password Protection. For more information, see the following link:

[SYSVOL Replication Migration Guide: FRS to DFS Replication](#)

Warning

The Microsoft Entra Password Protection DC Agent software currently installs on domain controllers in domains that are still using FRS for sysvol replication, but the software does NOT work properly in this environment. Negative side-effects include individual files failing to replicate, and sysvol restore procedures appearing to succeed but silently failing to replicate all files. You should migrate your domain

to use DFSR as soon as possible, both for DFSR's inherent benefits and also to unblock the deployment of Microsoft Entra Password Protection. Future versions of the software are automatically disabled when running in a domain that is still using FRS.

How much disk space does the feature require on the domain sysvol share?

The precise space usage varies since it depends on factors such as the number and length of the banned tokens in the Microsoft global banned list and the per-tenant custom list, plus encryption overhead. The contents of these lists are likely to grow in the future. With that in mind, a reasonable expectation is that the feature requires at least five (5) megabytes of space on the domain sysvol share.

Why is a reboot required to install or upgrade the DC agent software?

This requirement is caused by core Windows behavior.

Is there any way to configure a DC agent to use a specific proxy server?

No. Since the proxy server is stateless, it's not important which specific proxy server is used.

Is it okay to deploy the Microsoft Entra Password Protection Proxy service side by side with other services such as Microsoft Entra Connect?

Yes. The Microsoft Entra Password Protection Proxy service and Microsoft Entra Connect should never conflict directly with each other.

Unfortunately, the Microsoft Entra Password Protection Proxy software installs a version of the Microsoft Entra Connect Agent Updater service that is incompatible with the version installed by the [Microsoft Entra application proxy](#) software. This incompatibility may result in the Agent Updater service being unable to contact Azure for software updates. It isn't recommended to install Microsoft Entra Password Protection Proxy and Microsoft Entra application proxy on the same machine.

In what order should the DC agents and proxies be installed and registered?

Any ordering of Proxy agent installation, DC agent installation, forest registration, and Proxy registration is supported.

Should I be concerned about the performance hit on my domain controllers from deploying this feature?

The Microsoft Entra Password Protection DC Agent service shouldn't significantly impact domain controller performance in an existing healthy Active Directory deployment.

For most Active Directory deployments, password change operations are a small proportion of the overall workload on any given domain controller. As an example, imagine an Active Directory domain with 10,000 user accounts and a MaxPasswordAge policy set to 30 days. On average, this domain sees $10000/30 = \sim 333$ password change operations each day, which is a minor number of operations for even a single domain controller. Consider a potential worst case scenario: suppose those ~ 333 password changes on a single DC were done over a single hour. For example, this scenario may occur when many employees all come to work on a Monday morning. Even in that case, we're still looking at $\sim 333/60$ minutes = six password changes per minute, which again isn't a significant load.

However if your current domain controllers are already running at performance-limited levels (for example, maxed out with respect to CPU, disk space, disk I/O, and so on), it's advisable to add more domain controllers or expand available disk space, before deploying this feature. Refer to the previous question about sysvol disk space usage above.

I want to test Microsoft Entra Password Protection on just a few DCs in my domain. Is it possible to force user password changes to use those specific DCs?

No. The Windows client OS controls which domain controller is used when a user changes their password. The domain controller is selected based on factors such as Active Directory site and subnet assignments, environment-specific network configuration, and so on. Microsoft Entra Password Protection doesn't control these factors and can't influence which domain controller is selected to change a user's password.

One way to partially reach this goal would be to deploy Microsoft Entra Password Protection on all of the domain controllers in a given Active Directory site. This approach provides reasonable coverage for the Windows clients assigned to that site, and for the users that are logging into those clients and changing their passwords.

If I install the Microsoft Entra Password Protection DC Agent service on just the Primary Domain Controller (PDC), will all other domain controllers in the domain also be protected?

No. When a user's password is changed on a given non-PDC domain controller, the clear-text password is never sent to the PDC (this idea is a common mis-perception). Once a new password is accepted on a given DC, that DC uses that password to create the various authentication-protocol-specific hashes of that password and then persists those hashes in the directory. The clear-text password isn't persisted. The updated hashes are then replicated to the PDC. User passwords may in some cases be changed directly on the PDC, again depending on various factors such as network topology and Active Directory site design. (See the previous question.)

In summary, deployment of the Microsoft Entra Password Protection DC Agent service on the PDC is required to reach 100% security coverage of the feature across the domain. Deploying the feature on the PDC only doesn't provide Microsoft Entra Password Protection security benefits for any other DCs in the domain.

Why is custom smart lockout not working even after the agents are installed in my on-premises Active Directory environment?

Custom smart lockout is only supported in Microsoft Entra ID. Changes to the custom smart lockout settings in the Microsoft Entra admin center have no effect on the on-premises Active Directory environment, even with the agents installed.

Is a System Center Operations Manager management pack available for Microsoft Entra Password Protection?

No.

Why is Microsoft Entra ID still rejecting weak passwords even though I've configured the policy to be in Audit

mode?

Audit mode is only supported in the on-premises Active Directory environment. Microsoft Entra ID is implicitly always in "enforce" mode when it evaluates passwords.

My users see the traditional Windows error message when a password is rejected by Microsoft Entra Password Protection. Is it possible to customize this error message so that users know what really happened?

No. The error message seen by users when a password is rejected by a domain controller is controlled by the client machine, not by the domain controller. This behavior happens whether a password is rejected by the default Active Directory password policies or by a password-filter-based solution such as Microsoft Entra Password Protection.

Password testing procedures

You may want to do some basic testing of various passwords in order to validate proper operation of the software and to gain a better understanding of the [password evaluation algorithm](#). This section outlines a method for such testing that is designed to produce repeatable results.

Why is it necessary to follow such steps? There are several factors that make it difficult to perform controlled, repeatable testing of passwords in the on-premises Active Directory environment:

- The password policy is configured and persisted in Azure, and copies of the policy are synced periodically by the on-premises DC agent(s) using a polling mechanism. The latency inherent in this polling cycle may cause confusion. For example, if you configure the policy in Azure but forget to sync it to the DC agent, then your tests may not yield the expected results. The polling interval is currently hardcoded to be once per hour, but waiting an hour between policy changes is non-ideal for an interactive testing scenario.
- Once a new password policy is synced down to a domain controller, more latency occurs while it replicates to other domain controllers. These delays can cause unexpected results if you test a password change against a domain controller that hasn't received the latest version of the policy.
- Testing password changes via a user interface makes it difficult to have confidence in your results. For example, it's easy to mis-type an invalid password into a user

- interface, especially since most password user interfaces hide user input (for example, such as the Windows Ctrl-Alt-Delete -> Change password UI).
- It isn't possible to strictly control which domain controller is used when testing password changes from domain-joined clients. The Windows client OS selects a domain controller based on factors such as Active Directory site and subnet assignments, environment-specific network configuration, and so on.

In order to avoid these problems, the following steps are based on command-line testing of password resets while logged into a domain controller.

 **Warning**

Use these procedures only in a test environment. While the DC agent service is stopped, all incoming password changes and resets are accepted without validation. This also helps avoid the increased risks of logging into a domain controller.

The following steps assume you've installed the DC agent on at least one domain controller, have installed at least one proxy, and have registered both the proxy and the forest.

1. Log on to a domain controller using Domain Admin credentials or other credentials that have sufficient privileges to create test user accounts and reset passwords. Ensure that the domain controller has the DC agent software installed and has been rebooted.
2. Open up Event Viewer and navigate to the [DC Agent Admin event log](#).
3. Open an elevated command prompt window.
4. Create a test account for doing password testing

There are many ways to create a user account, but a command-line option is offered here as a way to make it easy during repetitive testing cycles:

text

```
net.exe user <testuseraccountname> /add <password>
```

For discussion purposes below, assume that we have created a test account named "ContosoUser", for example:

text

```
net.exe user ContosoUser /add <password>
```

5. Sign in to the Microsoft Entra admin center [↗](#) as at least an **Authentication Administrator**.
6. Browse to Protection > Authentication methods > Password protection.
7. Modify the Microsoft Entra Password Protection policy as needed for the testing you want to perform. For example, you may decide to configure either Enforced or Audit Mode, or you may decide to modify the list of banned terms in your custom banned passwords list.
8. Synchronize the new policy by stopping and restarting the DC agent service.

This step can be accomplished in various ways. One way would be to use the Service Management administrative console, by right-clicking on the Microsoft Entra Password Protection DC Agent service and choosing "Restart". Another way may be performed from the command prompt window like so:

text

```
net stop AzureADPasswordProtectionDCAgent && net start  
AzureADPasswordProtectionDCAgent
```

9. Check the Event Viewer to verify that a new policy has been downloaded.

Each time the DC agent service is stopped and started, you should see two 30006 events issued in close succession. The first 30006 event will reflect the policy that was cached on disk in the sysvol share. The second 30006 event (if present) should have an updated Tenant policy date, and if so will reflect the policy that was downloaded from Azure. The Tenant policy date value is currently coded to display the approximate timestamp that the policy was downloaded from Azure.

If the second 30006 event doesn't appear, you should troubleshoot the problem before continuing.

The 30006 events will look similar to this example:

text

The service is now enforcing the following Azure password policy.

```
Enabled: 1  
AuditOnly: 0  
Global policy date: 2018-05-15T00:00:00.000000000Z
```

```
Tenant policy date: 2018-06-10T20:15:24.432457600Z  
Enforce tenant policy: 1
```

For example, changing between Enforced and Audit mode will result in the AuditOnly flag being modified (the policy listed with AuditOnly=0 is in Enforced mode). Changes to the custom banned password list aren't directly reflected in the 30006 event above and aren't logged anywhere else for security reasons. Successfully downloading the policy from Azure after this change will also include the modified custom banned password list.

10. Run a test by trying to reset a new password on the test user account.

This step can be done from the command prompt window like so:

```
text  
  
net.exe user ContosoUser <password>
```

After running the command, you can get more information about the outcome of the command by looking in the event viewer. Password validation outcome events are documented in the [DC Agent Admin event log](#) topic; you'll use such events to validate the outcome of your test in addition to the interactive output from the net.exe commands.

Let's try an example: attempting to set a password that is banned by the Microsoft global list (note that list is [not documented](#) but we can test here against a known banned term). This example assumes that you have configured the policy to be in Enforced mode, and have added zero terms to the custom banned password list.

```
text  
  
net.exe user ContosoUser Password  
The password doesn't meet the password policy requirements. Check the  
minimum password length, password complexity, and password history  
requirements.  
  
More help is available by typing NET HELPMSG 2245.
```

Per the documentation, because our test was a password reset operation you should see a 10017 and a 30005 event for the ContosoUser user.

The 10017 event should look like this example:

```
text
```

The reset password for the specified user was rejected because it didn't comply with the current Azure password policy. For more information, please see the correlated event log message.

UserName: ContosoUser
FullName:

The 30005 event should look like this example:

text

The reset password for the specified user was rejected because it matched at least one of the tokens present in the Microsoft global banned password list of the current Azure password policy.

UserName: ContosoUser
FullName:

That was fun - let's try another example! Now, we'll attempt to set a password that is banned by the custom banned list while the policy is in Audit mode. This example assumes that you've completed the following steps: configured the policy to be in Audit mode, added the term "lachrymose" to the custom banned password list, and synchronized the resultant new policy to the domain controller by cycling the DC agent service as previously described.

Ok, set a variation of the banned password:

text

```
net.exe user ContosoUser LaChRymoSE!1  
The command completed successfully.
```

Remember, this time it succeeded because the policy is in Audit mode. You should see a 10025 and a 30007 event for the ContosoUser user.

The 10025 event should look like this example:

text

The reset password for the specified user would normally have been rejected because it didn't comply with the current Azure password policy. The current Azure password policy is configured for audit-only mode so the password was accepted. Please see the correlated event log message for more details.

```
UserName: ContosoUser  
FullName:
```

The 30007 event should look like this example:

text

The reset password for the specified user would normally be rejected because it matches at least one of the tokens present in the per-tenant banned password list of the current Azure password policy. The current Azure password policy is configured for audit-only mode so the password was accepted.

```
UserName: ContosoUser  
FullName:
```

11. Continue testing various passwords of your choice and checking the results in the event viewer using the procedures outlined in the previous steps. If you need to change the policy in the Microsoft Entra admin center, don't forget to synchronize the new policy down to the DC agent as described earlier.

We've covered procedures that enable you to do controlled testing of Microsoft Entra Password Protection's password validation behavior. Resetting user passwords from the command line directly on a domain controller may seem an odd means of doing such testing, but as described previously it's designed to produce repeatable results. As you're testing various passwords, keep the [password evaluation algorithm](#) in mind as it may help to explain results that you didn't expect.

Warning

When all testing is completed don't forget to delete any user accounts created for testing purposes!

Additional content

The following links aren't part of the core Microsoft Entra Password Protection documentation but may be a useful source of additional information on the feature.

[Microsoft Entra Password Protection is now generally available!](#) ↗

[Email Phishing Protection Guide – Part 15: Implement the Microsoft Entra Password Protection Service \(for On-Premises too!\)](#) ↗

[Microsoft Entra Password Protection and Smart Lockout are now in Public Preview!](#) ↗

Microsoft Premier\Unified support training available

If you want to learn more about Microsoft Entra Password Protection and how to deploy it, you can use a Microsoft proactive service. This service is available to customers with a Premier or Unified support contract. The service is called Microsoft Entra ID: Password Protection. Contact your Customer Success Account Manager for more information.

Next steps

If you have an on-premises Microsoft Entra Password Protection question that isn't answered here, submit a Feedback item below - thank you!

[Deploy Microsoft Entra password protection](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Microsoft Entra Password Protection agent version history

Article • 03/04/2025

To download the most recent version, see [Microsoft Entra Password Protection for Windows Server Active Directory](#).

1.2.177.1

Release date: March 28, 2022

- Fixed software version being incorrect

1.2.177.0

Release date: March 14, 2022

- Minor bugfixes
- Fixed issue with Microsoft Entra Connect Agent Updater not being updated

1.2.176.0

Release date: June 4, 2021

- Minor bugfixes to issues which prevented the proxy and DC agents from running successfully in certain environments.

1.2.172.0

Release date: February 22, 2021

It has been almost two years since the GA versions of the on-premises Microsoft Entra Password Protection agents were released. A new update is now available - see change descriptions below. Thank you to everyone who has given us feedback on the product.

- The DC agent and Proxy agent software both now require .NET 4.7.2 to be installed.
 - If .NET 4.7.2 is not already installed, download and run the installer found at [The .NET Framework 4.7.2 offline installer for Windows](#).

- The AzureADPasswordProtection PowerShell module is now also installed by the DC agent software.
- Two new health-related PowerShell cmdlets have been added: Test-AzureADPasswordProtectionDCAgent and Test-AzureADPasswordProtectionProxy.
- The AzureADPasswordProtection DC Agent password filter dll will now load and run on machines where lsass.exe is configured to run in PPL mode.
- Bug fix to password algorithm that allowed banned passwords fewer than five characters in length to be incorrectly accepted.
 - This bug is only applicable if your on-premises AD minimum password length policy was configured to allow fewer than five character passwords in the first place.
- Other minor bug fixes.

The new installers will automatically upgrade older versions of the software. If you have installed both the DC agent and the Proxy software on a single machine (only recommended for test environments), you must upgrade both at the same time.

It is supported to run older and newer versions of the DC agent and proxy software within a domain or forest, although we recommend upgrading all agents to the latest version as a best practice. Any ordering of agent upgrades is supported - new DC agents can communicate through older Proxy agents, and older DC agents can communicate through newer Proxy agents.

1.2.125.0

Release date: March 2, 2019

- Fix minor typo errors in event log messages
- Update EULA agreement to final General Availability version

 **Note**

Build 1.2.125.0 is the General Availability build. Thank you again to everyone who has provided feedback on the product!

1.2.116.0

Release date: March 3, 2019

- The Get-AzureADPasswordProtectionProxy and Get-AzureADPasswordProtectionDCAgent cmdlets now report software version and the

current Azure tenant with the following limitations:

- Software version and Azure tenant data are only available for DC agents and proxies running version 1.2.116.0 or later.
- Azure tenant data may not be reported until a re-registration (or renewal) of the proxy or forest has occurred.
- The Proxy service now requires that .NET 4.7 is installed.
 - If .NET 4.7 is not already installed, download and run the installer found at [The .NET Framework 4.7 offline installer for Windows](#).
 - On Server Core systems, it may be necessary to pass the /q flag to the .NET 4.7 installer to get it to succeed.
- The Proxy service now supports automatic upgrade. Automatic upgrade uses the Microsoft Entra Connect Agent Updater service, which is installed side by side with the Proxy service. Automatic upgrade is on by default.
- Automatic upgrade can be enabled or disabled using the Set-AzureADPasswordProtectionProxyConfiguration cmdlet. The current setting can be queried using the Get-AzureADPasswordProtectionProxyConfiguration cmdlet.
- The service binary for the DC agent service has been renamed to AzureADPasswordProtectionDCAgent.exe.
- The service binary for the Proxy service has been renamed to AzureADPasswordProtectionProxy.exe. Firewall rules may need to be modified accordingly if a third-party firewall is in-use.
 - NOTE: if an http proxy config file was being used in a previous Proxy install, it will need to be renamed (from *proxyservice.exe.config* to *AzureADPasswordProtectionProxy.exe.config*) after this upgrade.
- All time-limited functionality checks have been removed from the DC agent.
- Minor bugs fixes and logging improvements.

1.2.65.0

Release date: February 1, 2019

Changes:

- DC agent and proxy service are now supported on Server Core. Minimum OS requirements are unchanged from before: Windows Server 2012 for DC agents, and Windows Server 2012 R2 for proxies.
- The Register-AzureADPasswordProtectionProxy and Register-AzureADPasswordProtectionForest cmdlets now support device-code-based Azure authentication modes.

- The Get-AzureADPasswordProtectionDCAgent cmdlet will ignore mangled and/or invalid service connection points. This change fixes the bug where domain controllers would sometimes show up multiple times in the output.
- The Get-AzureADPasswordProtectionSummaryReport cmdlet will ignore mangled and/or invalid service connection points. This change fixes the bug where domain controllers would sometimes show up multiple times in the output.
- The Proxy PowerShell module is now registered from %ProgramFiles%\WindowsPowerShell\Modules. The machine's PSModulePath environment variable is no longer modified.
- A new Get-AzureADPasswordProtectionProxy cmdlet has been added to aid in discovering registered proxies in a forest or domain.
- The DC agent uses a new folder in the sysvol share for replicating password policies and other files.

Old folder location:

```
\\"<domain>\sysvol\<domain fqdn>\Policies\{4A9AB66B-4365-4C2A-996C-  
58ED9927332D}
```

New folder location:

```
\\"<domain>\sysvol\<domain fqdn>\AzureADPasswordProtection
```

(This change was made to avoid false-positive "orphaned GPO" warnings.)

Note

No migration or sharing of data will be done between the old folder and the new folder. Older DC agent versions will continue to use the old location until upgraded to this version or later. Once all DC agents are running version 1.2.65.0 or later, the old sysvol folder may be manually deleted.

- The DC agent and proxy service will now detect and delete mangled copies of their respective service connection points.
- Each DC agent will periodically delete mangled and stale service connection points in its domain, for both DC agent and proxy service connection points. Both DC agent and proxy service connection points are considered stale if its heartbeat timestamp is older than seven days.

- The DC agent will now renew the forest certificate as needed.
- The Proxy service will now renew the proxy certificate as needed.
- Updates to password validation algorithm: the global banned password list and customer-specific banned password list (if configured) are combined prior to password validations. A given password may now be rejected (fail or audit-only) if it contains tokens from both the global and customer-specific list. The event log documentation has been updated to reflect this; see [Monitor Microsoft Entra Password Protection](#).
- Performance and robustness fixes
- Improved logging

Warning

Time-limited functionality: the DC agent service in this release (1.2.65.0) will stop processing password validation requests as of September 1st 2019. DC agent services in prior releases (see list below) will stop processing as of July 1st 2019. The DC agent service in all versions will log 10021 events to the Admin event log in the two months leading up these deadlines. All time-limit restrictions will be removed in the upcoming GA release. The Proxy agent service is not time-limited in any version but should still be upgraded to the latest version in order to take advantage of all subsequent bug fixes and other improvements.

1.2.25.0

Release date: November 1, 2018

Fixes:

- DC agent and proxy service should no longer fail due to certificate trust failures.
- DC agent and proxy service have fixes for FIPS-compliant machines.
- Proxy service will now work properly in a TLS 1.2-only networking environment.
- Minor performance and robustness fixes
- Improved logging

Changes:

- The minimum required OS level for the Proxy service is now Windows Server 2012 R2. The minimum required OS level for the DC agent service remains at Windows Server 2012.

- The Proxy service now requires .NET version 4.6.2.
- The password validation algorithm uses an expanded character normalization table. This change may result in passwords being rejected that were accepted in prior versions.

1.2.10.0

Release date: August 17, 2018

Fixes:

- Register-AzureADPasswordProtectionProxy and Register-AzureADPasswordProtectionForest now support multi-factor authentication
- Register-AzureADPasswordProtectionProxy requires a WS2012 or later domain controller in the domain to avoid encryption errors.
- DC agent service is more reliable about requesting a new password policy from Azure on startup.
- DC agent service will request a new password policy from Azure every hour if necessary, but will now do so on a randomly selected start time.
- DC agent service will no longer cause an indefinite delay in new DC advertisement when installed on a server prior to its promotion as a replica.
- DC agent service will now honor the "Enable password protection on Windows Server Active Directory" configuration setting
- Both DC agent and proxy installers will now support in-place upgrade when upgrading to future versions.

Warning

In-place upgrade from version 1.1.10.3 is not supported and will result in an installation error. To upgrade to version 1.2.10 or later, you must first completely uninstall the DC agent and proxy service software, then install the new version from scratch. Re-registration of the Microsoft Entra password protection Proxy service is required. It is not required to re-register the forest.

Note

In-place upgrades of the DC agent software will require a reboot.

- DC agent and proxy service now support running on a server configured to only use FIPS-compliant algorithms.

- Minor performance and robustness fixes
- Improved logging

1.1.10.3

Release date: June 15, 2018

Initial public preview release

Next steps

[Deploy Microsoft Entra Password Protection](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Protect user accounts from attacks with Microsoft Entra smart lockout

Article • 04/29/2025

Smart lockout helps lock out bad actors that try to guess your users' passwords or use brute-force methods to get in. Smart lockout can recognize sign-ins that come from valid users and treat them differently than ones of attackers and other unknown sources. Attackers get locked out, while your users continue to access their accounts and be productive.

How smart lockout works

By default, smart lockout locks an account from sign-in after:

- 10 failed attempts in Azure Public and Microsoft Azure operated by 21Vianet tenants
- 3 failed attempts for Azure US Government tenants

The account locks again after each subsequent failed sign-in attempt. The lockout period is one minute at first, and longer in subsequent attempts. To minimize the ways an attacker could work around this behavior, we don't disclose the rate at which the lockout period increases after unsuccessful sign-in attempts.

Smart lockout tracks the last three bad password hashes to avoid incrementing the lockout counter for the same password. If someone enters the same bad password multiple times, this behavior doesn't cause the account to lock out.

Note

Hash tracking functionality isn't available for customers with pass-through authentication enabled as authentication happens on-premises not in the cloud.

Federated deployments that use Active Directory Federation Services (AD FS) 2016 and AD FS 2019 can enable similar benefits by using [AD FS Extranet Lockout and Extranet Smart Lockout](#). It's recommended to move to [managed authentication](#).

Smart lockout is always on, for all Microsoft Entra customers, with these default settings that offer the right mix of security and usability. Customization of the smart lockout settings, with values specific to your organization, requires Microsoft Entra ID P1 or higher licenses for your users.

Using smart lockout doesn't guarantee that a genuine user is never locked out. When smart lockout locks a user account, we try our best to not lock out the genuine user. The lockout

service attempts to ensure that bad actors can't gain access to a genuine user account. The following considerations apply:

- Lockout state across Microsoft Entra data centers is synchronized. However, the total number of failed sign-in attempts allowed before an account is locked out will have slight variance from the configured lockout threshold. Once an account is locked out, it's locked out everywhere across all Microsoft Entra data centers.
- Smart Lockout uses familiar location versus unfamiliar location to differentiate between a bad actor and the genuine user. Both unfamiliar and familiar locations have separate lockout counters.

To refrain the system from locking out a user signing in from an unfamiliar location, they must use the correct password to avoid being locked out and have a low number of previous lockout attempts from unfamiliar locations. If the user is locked out from an unfamiliar location, the user should consider SSPR to reset the lockout counter.

- After an account lockout, the user can initiate self-service password reset (SSPR) to sign in again. If the user chooses **I forgot my password** during SSPR, the duration of the lockout is reset to 0 seconds. If the user chooses **I know my password** during SSPR, the lockout timer continues, and the duration of the lockout isn't reset. To reset the duration and sign in again, the user needs to change their password.

Smart lockout can be integrated with hybrid deployments that use password hash sync or pass-through authentication to protect on-premises Active Directory Domain Services (AD DS) accounts from being locked out by attackers. By setting smart lockout policies in Microsoft Entra ID appropriately, attacks can be filtered out before they reach on-premises AD DS.

When using [pass-through authentication](#), the following considerations apply:

- The Microsoft Entra lockout threshold must be **less** than the AD DS account lockout threshold. Set the values so that the AD DS account lockout threshold is at least two or three times greater than the Microsoft Entra lockout threshold.
- The Microsoft Entra lockout duration must be **longer** than the AD DS account lockout duration. The Microsoft Entra duration is set in seconds, while the AD DS duration is set in minutes.

Tip

This configuration ensures Microsoft Entra smart lockout stops your on-premises AD DS accounts from being locked out by brute force attacks, like [password spray attacks](#) on your Microsoft Entra accounts.

For example, if you want your Microsoft Entra smart lockout duration to be higher than AD DS, then Microsoft Entra ID would be 120 seconds (2 minutes) while your on-premises AD is set to 1 minute (60 seconds). If you want your Microsoft Entra lockout threshold to be 10, then you want your on-premises AD DS lockout threshold to be 5.

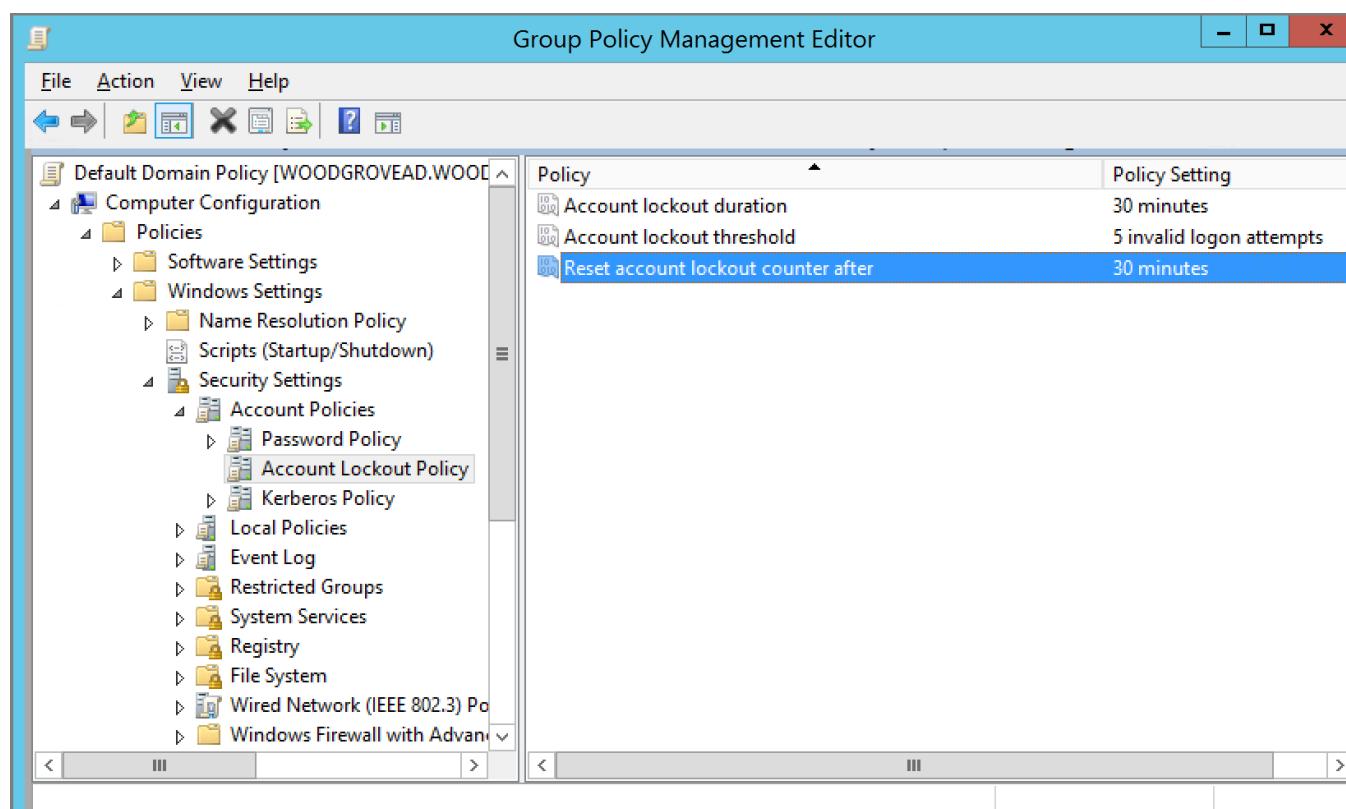
Important

An administrator can unlock the users' cloud account if they have been locked out by the Smart Lockout capability, without the need of waiting for the lockout duration to expire. For more information, see [Reset a user's password using Microsoft Entra ID](#).

Verify on-premises account lockout policy

To verify your on-premises AD DS account lockout policy, complete the following steps from a domain-joined system with administrator privileges:

1. Open the Group Policy Management tool.
2. Edit the group policy that includes your organization's account lockout policy, such as, the **Default Domain Policy**.
3. Browse to **Computer Configuration > Policies > Windows Settings > Security Settings > Account Policies > Account Lockout Policy**.
4. Verify your **Account lockout threshold** and **Reset account lockout counter after** values.



Manage Microsoft Entra smart lockout values

Based on your organizational requirements, you can customize the Microsoft Entra smart lockout values. Customization of the smart lockout settings, with values specific to your organization, requires Microsoft Entra ID P1 or higher licenses for your users. Customization of the smart lockout settings isn't available for Microsoft Azure operated by 21Vianet tenants.

To check or modify the smart lockout values for your organization, complete the following steps:

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Entra ID > Authentication methods > Password protection**.
3. Set the **Lockout threshold**, based on how many failed sign-ins are allowed on an account before its first lockout.
The default is 10 for Azure Public tenants and 3 for Azure US Government tenants.
4. Set the **Lockout duration in seconds**, to the length in seconds of each lockout.
The default is 60 seconds (one minute).

 **Note**

If the first sign-in after a lockout period has expired also fails, the account locks out again. If an account locks repeatedly, the lockout duration increases.

The screenshot shows the Microsoft Entra admin center interface. The left sidebar has a tree view with 'Identity' expanded, showing 'Overview', 'Users', 'Groups', 'Devices', 'Applications', 'Roles & admins', 'Billing', and 'Settings'. 'Protection' is selected and highlighted with a red box. Under 'Protection', 'Authentication methods' is also highlighted with a red box. Other options include 'Identity Protection', 'Conditional Access', 'Security Center', 'Identity Secure Score', 'Multifactor authentication', 'Password reset', 'Custom security attributes', and 'Risky activities'. The main content area is titled 'Authentication methods | Password protection' for 'Contoso - Microsoft Entra ID Security'. It includes a search bar, save/discard buttons, and a 'Got feedback?' link. A 'Manage' section has 'Policies' (selected and highlighted with a red box) and 'Password protection' (also highlighted with a red box). Other policy categories like 'Registration campaign', 'Authentication strengths', and 'Settings' are listed. A 'Monitoring' section includes 'Activity', 'User registration details', 'Registration and reset events', and 'Bulk operation results'. Under 'Password protection', settings for 'Custom smart lockout' (lockout threshold set to 10, duration to 60 seconds), 'Custom banned passwords' (list containing contoso, fabrikam, tailwind, michigan, wolverine, harbaugh, howard), and 'Password protection for Windows Server Active Directory' (set to 'Yes') are shown. Mode is set to 'Enforced'.

Testing Smart lockout

When the smart lockout threshold is triggered, you'll get the following message while the account is locked:

Your account is temporarily locked to prevent unauthorized use. Try again later, and if you still have trouble, contact your admin.

When you test smart lockout, your sign-in requests might be handled by different datacenters due to the geo-distributed and load-balanced nature of the Microsoft Entra authentication service.

Smart lockout tracks the last three bad password hashes to avoid incrementing the lockout counter for the same password. If someone enters the same bad password multiple times, this behavior doesn't cause the account to lock out.

Default protections

In addition to Smart lockout, Microsoft Entra ID also protects against attacks by analyzing signals including IP traffic and identifying anomalous behavior. Microsoft Entra ID blocks these malicious sign-ins by default and returns [AADSTS50053 - IdsLocked error code](#), regardless of the password validity.

Next steps

- To customize the experience further, you can [configure custom banned passwords for Microsoft Entra password protection](#).
- To help users reset or change their password from a web browser, you can [configure Microsoft Entra self-service password reset](#).

Track and investigate identity activities with linkable identifiers in Microsoft Entra (preview)

Article • 03/31/2025

Microsoft includes certain identifiers in all tokens that can be used to link activities from one root authentication. Linkable identifiers are currently in preview and exposed in customer-facing logs. Linkable identifiers help threat hunters and analysts investigate and remediate identity-related attacks. They significantly improve how security analysts and professionals can track, investigate, and remediate identity-related attacks across sessions and tokens, providing you with a more secure and transparent ecosystem.

There are two types of linkable identifiers:

- One is based on session ID (SID). It helps link all authentication artifacts issued from a single root authentication with the same identifier, which can be used to link or connect tokens in a single chain together. For example, a SID-based linkable identifier can track all the activities done by all the access tokens issued from a long-lived token, like a [refresh token \(RT\)](#) or session cookies.
- Another tracks activities done by a specific token access, like an [access token \(AT\)](#) or [ID token](#).

To help link all authentication artifacts issued from a single root authentication, the SID claim is created and included in primary refresh tokens (PRT), refresh token, or session cookie each time a user performs an interactive authentication for an account. The same SID value is added to each access token issued from a refresh token or in session cookie. It can be used to link all authentication artifacts, and can further filter for a specific user or device within a session.

SID-based scenarios include:

- Start with a session ID from Microsoft Entra sign in logs, and join with workload logs like Exchange Online audit logs or Microsoft Graph activity logs to identify all the activities done by all of the access tokens with the same session ID.
- Filter results further by UserId or Deviceld, or with a token issued within a time frame of a specific session.
- Determine how many sessions are alive for a given user (UserId) or a given device (Deviceld).

In addition, Microsoft Entra has another important linkable security claim called unique token identifier (UTI) that is a unique GUID present in all Microsoft Entra tokens. It serves to uniquely identify a token or request.

For token investigation, UTI gives finer granularity when you want to track down a particular suspicious token. A UTI is unique for every AT and SID and helps you investigate all of the tokens within a specific session. For more information about these claims, see [Access token](#) or [ID token](#).

UTI-based scenarios include:

- Start with a UTI (which points to a specific access token) from Microsoft Entra sign in logs and join with workload logs like Exchange Online audit logs or Microsoft Graph activity logs to identify all the activities done on behalf of the access token (UTI).

Linkable identifier claims

[+] [Expand table](#)

Claim	Format	Description
oid	String, a GUID	The immutable identifier for the requestor, which is the verified identity of the user or service principal. This ID uniquely identifies the requestor across applications.
tid	String, a GUID	Represents the tenant that the user is signing in to.
sid	String, a GUID	Represents a unique identifier for an entire session and is generated when a user does interactive authentication. This ID helps link all authentication artifacts issued from a single root authentication.
deviceid	String, a GUID	Represents a unique identifier for the device from which a user is interacting with an application.
uti	String	Represents the token identifier claim. This ID is a unique, per-token identifier that is case-sensitive.
iat	int, a Unix timestamp	Specifies when the authentication for this token occurred.

As of now, linkable identifiers are logged into Microsoft Entra sign in logs, Exchange Online Audit logs and Microsoft Graph Activity logs.

Linkable identifiers in Microsoft Entra sign in logs

A sign-in logs entry has the following linkable identifier claims.

[+] Expand table

Claim	Sign in log attribute name
oid	User ID
tid	Resource Tenant ID
sid	Session ID
deviceid	Device ID
uti	Unique Token Identifier
iat	Date

To view the sign-in logs from the Microsoft Entra admin center:

1. Sign in to the [Microsoft Entra admin center](#) as at least a [Reports Reader](#).
2. Browse to **Identity > Monitoring & health > Sign-in logs**.
3. Filter by time, or by specific user to look at the specific log entries.
4. Click any sign-in log entry.
5. **Basic Info** shows the User ID, Resource Tenant ID, Session ID, Unique Token Identifier, and Date. **Devices** shows the Device ID for registered and domain-joined devices.

Activity Details: Sign-ins

X

Basic info	Location	Device info	Authentication Details	Conditional Access	Report-only	...
Date (UTC)	3/5/2025, 11:28:36 PM					
Request ID	22320e4b-43ab-4134-b06b-f2074e8e0700					
Correlation ID	146b8c42-78d8-4bcf-b15e-9d0dd1908633					
Authentication requirement	Multifactor authentication					
Status	Success					
Continuous access evaluation	No					
Additional Details	MFA requirement satisfied by claim in the token					
Follow these steps:						
Troubleshoot Event			Launch the Sign-in Diagnostic.			
			1. Review the diagnosis and act on suggested fixes.			
User	Swarnika Iwalewa					
Username						
User ID						
Sign-in identifier						
Session ID	002e91f9-6e73-22ee-a998-a10327b89942					
App owner tenant ID						
Resource owner tenant ID						
User type	Member					
Cross tenant access type	None					

Activity Details: Sign-ins

X

Basic info	Location	Device info	Authentication Details	Conditional Access	Report-only	...
Device ID						
Browser	Edge 18.26100					
Operating System	Windows10					
Compliant	No					
Managed	No					
Join Type						

You should start with Microsoft Entra sign-in logs User ID attribute and manually search on the workload audit logs to track all the activities using a specific access token. Similarly, the Session ID attribute can be used to manually search on the workload audit logs to track all the activities.

Microsoft Exchange Online logs

Exchange Online audit logs help you access critical audit log event data to gain insight, and further investigate user activities. Exchange Online carries forward the linkable identifiers from Microsoft Entra tokens and logs all of the linkable identifiers in the Exchange Online audit logs.

For scenarios like mailbox update, items moved or deleted, you can start with linkable identifiers from Microsoft Entra sign in logs and search Microsoft Purview Audit (Standard) and Audit (Premium) to track all user actions on any items in a mailbox. For more information about how to search Exchange Online audit logs, see [Search the audit log | Microsoft Learn](#).

[+] Expand table

Claim	Exchange Online audit log attribute name
oid	TokenObjectId
tid	TokenTenantId
sid	SessionID / AADSessionId within App Access Context object
deviceid	DeviceId (Available only for registered/domain joined device)
uti	UniqueId within App Access Context object
iat	IssuedAtTime within App Access Context object

View Exchange Online logs using Microsoft Purview portal

1. Go to [Microsoft Purview portal](#).
2. Search for logs with a specific timeframe and record types starting with Exchange.

The screenshot shows the Microsoft Purview Audit search interface. The left sidebar has links for Home, Solutions (Audit, Search, Policies), Learn, Settings, and eDiscovery. The main area has a 'Search' section with fields for 'Date and time range (UTC)', 'Keyword Search', and 'Admin Units'. To the right, there's a search bar with 'Exchange' typed in, a list of checked filter options (ExchangeAdmin, ExchangeAggregatedOperation, ExchangeItem, ExchangeItemAggregated, ExchangeItemGroup, ExchangeSearch), and sections for 'Users', 'File, folder, or site', and 'Workloads'.

3. You can further filter for a specific user, or a UTI value from Microsoft Entra sign-in logs. You can filter all the activity logs within a session with `SessionId`.

4. The results show all the linkable identifiers.

The screenshot shows the Microsoft Entra Audit search interface. The left pane displays a table of audit logs with columns: Date (UTC), IP Address, User, Record Type, and Activity. The right pane shows detailed information for a selected log entry:

- Details**:
 - Date (UTC): 2025-01-07T18:39:06
 - IP Address: [redacted]
- Users**: User@contoso-onmicrosoft.com
- Activity**: Set-Mailbox
- Item**: [redacted]
- Details**: Admin Units
- AppAccessContext**:

```
{ "AADSessionId": "00000000-0000-0000-0000-000000000000", "IssuedAtTime": "2025-01-07T18:32:27Z", "UniqueId": "00000000-0000-0000-0000-000000000000" }
```
- CreationTime**: 2025-01-07T18:39:06



Details

fb78d390-0c51-40cd-8e17-fdbfab77341b

AppPoolName

MSExchangeAdminApiNetCore

ClientAppId

CorrelationID

ExternalAccess

false

OrganizationName

.onmicrosoft.com

OriginatingServer

AM8PR07MB8263 (15.20.8314.018)

Parameters

```
[  
  {  
    "Name": "Identity",  
    "Value": ".onmicrosoft.com"  
  },  
  {  
    "Name": "MaxSendSize",  
    "Value": "99 MB (103,809,024 bytes)"  
  }  
]
```

RequestId

62c74247-ad9f-48e2-44bc-25668796e2a5

SessionId

dab6990b-d64a-43c1-aead-1c12b25262e7

TokenObjectId

85c27ae3-8ca7-415d-bbcb-71aba361edaa

TokenTenantId

Close

5. Export the audit log and investigate for a specific `SessionId` or `UniqueTokenId` for all the activities for Exchange Online.

View Exchange Online logs using PowerShell commandlets

1. Run PowerShell as an administrator.
2. If the ExchangeOnlineManagement module isn't installed, run:

```
PowerShell  
  
Install-Module -Name ExchangeOnlineManagement
```

3. Connect to Exchange Online:

```
PowerShell  
  
Connect-ExchangeOnline -UserPrincipalName <user@4jkvzv.onmicrosoft.com>
```

4. Run some mailbox commands:

```
PowerShell  
  
Set-Mailbox user@4jkvzv.onmicrosoft.com -MaxSendSize 97MB
```

```
PowerShell  
  
Set-Mailbox user@4jkvzv.onmicrosoft.com -MaxSendSize 98MB
```

```
PowerShell  
  
Set-Mailbox user@4jkvzv.onmicrosoft.com -MaxSendSize 99MB
```

5. Search the unified audit log:

```
PowerShell  
  
Search-UnifiedAuditLog -StartDate 01/06/2025 -EndDate 01/08/2025 -  
RecordType ExchangeItem, ExchangeAdmin, ExchangeAggregatedOperation,  
ExchangeItemAggregated, ExchangeItemGroup, ExchangeSearch
```

6. The results have all of the linkable identifiers.

 Note

The linkable identifiers aren't available in the Exchange Online audit logs on some aggregated log entries, or logs generated from background processes.

For more information, see [Exchange Online PowerShell](#).

Microsoft Graph activity logs

Microsoft Graph activity logs are an audit trail of all HTTP requests that the Microsoft Graph service received and processed for a tenant. The logs are stored in Log Analytics for analysis.

If you send Microsoft Graph activity logs to a Log Analytics workspace, you can query the logs using Kusto Query Language (KQL). For scenarios involving Microsoft Graph activity, you can start with linkable identifiers from Microsoft Entra sign in logs, and check against Microsoft Graph activity logs to track all user actions on any items in a mailbox. For more information about how to search Microsoft Graph activity logs, see [Microsoft Graph Activity Logs](#).

[] [Expand table](#)

Claim	Attribute name in the Exchange Online audit log
oid	UserId
tid	TenantId
sid	SessionId
deviceid	DeviceId (available only for registered and domain-joined devices)
uti	SignInActivityId
iat	TokenIssuedAt

Join sign-in logs and Microsoft Graph activity logs using KQL

You can use KQL to join Microsoft Entra sign-in logs and Microsoft Graph Activity logs. You can filter logs by `uti` attribute to analyze all the activities by a specific access token. Or you can filter logs by `sid` claim to analyze all activities of all access tokens from a refresh token obtained from a root interactive authentication. The log can be filtered further by using other attributes like `UserId`, `DeviceId`, and so on.

kql

```
MicrosoftGraphActivityLogs
| where TimeGenerated > ago(4d) and UserId == '4624cd8c-6c94-4593-b0d8-a4983d797ccb'
| join kind=leftouter (union
SigninLogs,
AADNonInteractiveUserSignInLogs,
AADServicePrincipalSignInLogs,
AADManagedIdentitySignInLogs,
ADFSSignInLogs
| where TimeGenerated > ago(4d))
on $left.SignInActivityId == $right.UniqueTokenIdentifier
```

The screenshot shows the Microsoft Azure Log Analytics interface. On the left, there's a navigation sidebar with categories like Home, Licenses, Cross-tenant synchronization, Microsoft Entra Connect, Custom domain names, Mobility (MDM and WIP), Password reset, Company branding, User settings, Properties, Security, Monitoring, Sign-in logs, Audit logs, Provisioning logs, Health, Log Analytics (which is selected and highlighted in grey), Diagnostic settings, Workbooks, Usage & insights, and Bulk operation results (Preview). Below the main content area, there's a footer bar with '1s 476ms' and 'Display time (UTC+00:00) ~'. The main content area has tabs for 'Results' and 'Chart'. The 'Results' tab is active, showing a table of log data with columns: TimeGenerated [UTC], IPAddress, Location, RequestId, OperationId, ClientRequestId, ApiVersion, and RequestMethod. The table contains 17 rows of data, including fields like UserAgent (Outlook/34959949 CFNetwork/3826.400.120 Darwin/24.3.0), RequestUri (https://graph.microsoft.com/v1.0/organization?\$select=displayName), DurationMs (672607), ResponseSizeBytes (127), SignInActivityId (qwqRdcBjQEW7TOeisjwCAA), SessionId (127), DeviceId (null), TokenIssuedAt [UTC] (2025-03-04T00:55:51Z), AppId (27922004-5251-4030-b22d-91ecd9a37e4), UserId (null), Scopes (email Files.ReadWrite.All FileStorageContainer.Selected Mail.Read Mail.Read.Shared openid People.Read People.Read.All Presence.Read All profile Sites.ReadWrite.All User.Read User.ReadBasic User.ReadWrite User.ReadWrite.All), ClientAuthMethod (0), Wids (f2e9ff92c-3afb-46b9-b7cf-a126ee74c451 b79fb4d-3ef9-4689-8143-76b194e85509), ATContent (eyJ0eXAiOiJKV1QiLCJhbGciOiJSImlmQ3R1JX0HDE5fsl3dzNlb1NhUThUzN1Ytncp3RE9ZyJnVUk1NVpIN2MilCyaC16jEuQW84RDuhGIVOhdWOGtXUxXSQpVtVn3dNQFBQUBFQUBfD0FBK), ATContentH ({"typ": "JWT", "nonce": "d7GRB_KLGr_ilww3e..."}, and ATContentP ("JWT~00000001-0000-0000-0000-000000000000"). At the top of the main content area, there are buttons for 'New Query', 'Run', 'Save', 'Share', 'New alert rule', 'Export', 'Pin to', and 'Format query'. There's also a 'Try the new Log An...' link, a 'Feedback' button, and a 'Queries hub' link. The top right corner shows the user's name (swiv93@woodgrove.ms) and the tenant name (WOODGROVE (WOODGROVE.MS)).

For more information about queries in Log Analytics Workspace, see [Analyze Microsoft Entra activity logs with Log Analytics](#).

Scenario walkthrough

Let's walk through an example where a user logs into office.com. Then the user accesses Microsoft Graph, and executes some commands. Finally, the user access Exchange Online to use Outlook email, and do some mail operations.

1. Find the interactive login log line in the sign in logs, and capture the SessionId:

Sign-in logs		...	X				
		Download	Export Data Settings	Troubleshoot	Refresh	Columns	Got feedback?
Date : Last 24 hours	Show dates as : UTC	User contains Swarnika Iwalewa	X	Add filters			
User sign-ins (interactive)	User sign-ins (non-interactive)	Service principal sign-ins	Managed identity sign-ins				
Date (UTC)	Request ID	User	Application	Unique token identifier	Session ID		
3/20/2025, 7:54:38 PM	eecdada89-d87c-4398-8d7d-cd68e5f70200	Swarnika Iwalewa	OfficeHome	idrN7nzYmEONfc1o5fcCAA	0030dbe9-4c0f-98d3-d5c4-6dc3ea451adf		

2. Add a filter by `SessionId`. You can get the `SessionId` for interactive or noninteractive sign-ins.

Interactive sign-ins:

Date (UTC)	Request ID	User	Application	Unique token identifier	Session ID
3/20/2025, 8:13:47 PM	45aebfd7-a934-4855-a664-9b7e72b20300	Swarnika Iwalewa	Microsoft Account Controls V2	17-uRTSpVUimZJt-crIDAA	0030dbe9-4c0f-98d3-d5c4-6dce3a451adf
3/20/2025, 8:13:45 PM	a3de31b1-0a3a-49b1-903d-b02d0c750000	Swarnika Iwalewa	Azure AD Identity Governance - Entitlement ...	sTHeozaKUjmQPbAtDHUAAA	0030dbe9-4c0f-98d3-d5c4-6dce3a451adf
3/20/2025, 8:13:12 PM	f1a78c00-23ca-4b69-b449-04fc0260100	Swarnika Iwalewa	Microsoft Account Controls V2	AlynBcpaJuU05Q7zwCYBAAA	0030dbe9-4c0f-98d3-d5c4-6dce3a451adf
3/20/2025, 8:13:09 PM	b2ea03dc-177d-4435-8965-3782bdf90000	Swarnika Iwalewa	Azure AD Identity Governance - Entitlement ...	3APgsn0XNUSjZTeCvfaAAA	0030dbe9-4c0f-98d3-d5c4-6dce3a451adf
3/20/2025, 7:55:10 PM	547c9d4-6acc-4abe-ab0c-5a4646450300	Swarnika Iwalewa	OfficeHome	x18VmqqkqPfkbUDAA	0030dbe9-4c0f-98d3-d5c4-6dce3a451adf
3/20/2025, 7:54:40 PM	cccc5763-6e2d-478b-8f1f-c59a29c80000	Swarnika Iwalewa	OfficeHome	Y1MzC1uiDePH8WaKcgAAA	0030dbe9-4c0f-98d3-d5c4-6dce3a451adf
3/20/2025, 7:54:38 PM	eeccda89-d87c-4398-8d7d-cd68e5f70200	Swarnika Iwalewa	OfficeHome	idN7nxYmEONfc1o5fCAA	0030dbe9-4c0f-98d3-d5c4-6dce3a451adf

Noninteractive sign-ins:

Date (UTC)	Request ID	Username	Application
3/20/2025, 12:00:00	Aggregate	swiw93@woodgrove...	Microsoft Account Cor...
3/20/2025, 8:1	9feffffad-6073-437e-82	swiw93@woodgrove...	Microsoft Account Cor...
3/20/2025, 8:1	d2ce7522-7be9-4a76-4	swiw93@woodgrove...	Microsoft Account Cor...
> 3/20/2025, 12:00:00	Aggregate	swiw93@woodgrove...	Azure AD Identity Gov...
> 3/20/2025, 12:00:00	Aggregate	swiw93@woodgrove...	Azure AD Identity Gov...
> 3/20/2025, 12:00:00	Aggregate	swiw93@woodgrove...	Office 365 Search Serv...
> 3/20/2025, 12:00:00	2ec20d0-833e-4d79-5	swiw93@woodgrove...	OfficeHome
> 3/20/2025, 12:00:00	1cc7184f-f265-4522-ac	swiw93@woodgrove...	OfficeHome
> 3/20/2025, 12:00:00	5d3999a7-f54b-4f7e-b	swiw93@woodgrove...	Microsoft Office 365 Pr...
> 3/20/2025, 12:00:00	d8d2b362-b67e-4f96-t	swiw93@woodgrove...	OfficeHome

Activity Details: Sign-ins

CONTINUATION DOWNLOAD CELL GRID WORKSHEETS

Authentication requirement: Single-factor authentication
 Status: Success
 Continuous access evaluation: No
 Additional Details: MFA requirement satisfied by claim in the token

Follow these steps:
 Troubleshoot Event: Launch the Sign-in Diagnostic.
 1. Review the diagnosis and act on suggested fixes.

User: Swarnika Iwalewa
 Username: swiw93@woodgrove.ms
 User ID:
 Sign-in identifier:
 Session ID: 0030dbe9-4c0f-98d3-d5c4-6dce3a451adf
 App owner tenant ID:
 Resource owner tenant ID: False
 User type: Member
 Cross tenant access type: None
 Application: Microsoft Account Controls V2

3. To get all the activities on Microsoft Graph workload done by the user within this specific session, go to Log Analytics in Microsoft Entra admin center and run the query to join Microsoft Entra sign in logs and Microsoft Graph Activity logs. The following query filters by `UserId` and `SessionId`.

```

New Query 1+ ... +
Run Time range: Set in query Show: 30000 results
KQL mode

1 MicrosoftGraphActivityLogs
2 | where TimeGenerated > ago(2d) and UserId == 'aaaaaaaa-bbbb-cccc-1111-222222222222' and SessionId == '0030dbe9-4c0f-98d3-d5c4-6dce3a451ad'
3 | join kind=inner ComputerUnion
4 | SigninLogs
5 | AADNonInteractiveUserSignInLogs,
6 | ADServicePrincipalSignInLogs,
7 | AADManagedIdentitySignInLogs,
8 | ADFSSignInLogs
9 | where TimeGenerated > ago(2d)
on $left.SignInActivityId == $right.UniqueTokenIdentifier

```

Results Chart

TimeGenerated [UTC]	Userid	SignInActivityId	TokenIssuedAt [UTC]	SessionId	TenantId	DeviceId
3/20/2025, 8:13:48.227 PM	aaaaaaaa-bbbb-cccc-1111-222222222222	r_f_vn3Ngf1OCzp49LwDAA	3/20/2025, 8:08:47.000 PM	0030dbe9-4c0f-98d3-d5c4-6d...		
3/20/2025, 8:13:47.464 PM	aaaaaaaa-bbbb-cccc-1111-222222222222	MfA8QSNjNECzYucGP8XAA	3/20/2025, 8:08:46.000 PM	0030dbe9-4c0f-98d3-d5c4-6d...		
3/20/2025, 8:13:47.354 PM	aaaaaaaa-bbbb-cccc-1111-222222222222	MfA8QSNjNECzYucGP8XAA	3/20/2025, 8:08:46.000 PM	0030dbe9-4c0f-98d3-d5c4-6d...		
3/20/2025, 8:13:47.166 PM	aaaaaaaa-bbbb-cccc-1111-222222222222	c09sgxR0GUWFB3HbJlRAA	3/20/2025, 8:08:46.000 PM	0030dbe9-4c0f-98d3-d5c4-6d...		
3/20/2025, 8:13:47.017 PM	aaaaaaaa-bbbb-cccc-1111-222222222222	MfA8QSNjNECzYucGP8XAA	3/20/2025, 8:08:46.000 PM	0030dbe9-4c0f-98d3-d5c4-6d...		
3/20/2025, 8:13:43.757 PM	aaaaaaaa-bbbb-cccc-1111-222222222222	InxOuU7IukopukxNVO7ZAA	3/20/2025, 8:08:42.000 PM	0030dbe9-4c0f-98d3-d5c4-6d...		
3/20/2025, 8:13:41.762 PM	aaaaaaaa-bbbb-cccc-1111-222222222222	4ZHIVenNaEfGux8bIgEAA	3/20/2025, 8:08:40.000 PM	0030dbe9-4c0f-98d3-d5c4-6d...		
3/20/2025, 8:04:11.885 PM	aaaaaaaa-bbbb-cccc-1111-222222222222	LdAgCEdakkuouKv9USA	3/20/2025, 7:50:17.000 PM	0030dbe9-4c0f-98d3-d5c4-6d...		
3/20/2025, 7:55:18.158 PM	aaaaaaaa-bbbb-cccc-1111-222222222222	p5k3XUv1Rk-09MCz9eXAA	3/20/2025, 7:50:17.000 PM	0030dbe9-4c0f-98d3-d5c4-6d...		
3/20/2025, 7:55:17.672 PM	aaaaaaaa-bbbb-cccc-1111-222222222222					
3/20/2025, 7:55:16.767 PM	aaaaaaaa-bbbb-cccc-1111-222222222222	J4CAuCEJlkowOvK3-AUSA	3/20/2025, 7:50:17.000 PM	0030dbe9-4c0f-98d3-d5c4-6d...		

2s 791ms | Display time (UTC+00:00) | Query details | 1 - 11 of 11

Further filtering can be done on a `SignInActivityId` (uti claim) attribute to learn more about the access by specific request.

4. To get Exchange Online activities, open the Microsoft Purview portal and search by Users or Record Types.

New Search Audit retention policies

Searches completed 3 Active searches 1 Active unfiltered searches 1

Date and time range (UTC) • Start Mar 19 2025 00:00 End Mar 20 2025 00:00

Activities - friendly names Choose which activities to search for

Activities - operation names Enter operation values, separated by commas

Users Swarnika Iwalewa Add the users whose audit logs you want to see...

File, folder, or site Enter all or a part of the name of a file, website, or folder

Record Types ExchangeAdmin, ExchangeAggregatedOperation, ExchangeItem, Exchange...
ExchangeAdmin, ExchangeAggregatedOperation, ExchangeItem, ExchangeItemAggregated, ExchangeItemGroup, ExchangeSearch

Keyword Search Enter the keyword to search for

Admin Units Choose which Admin Units to search for

Search name Give the search a name

Search Clear all

Copy this search Delete Refresh

Search name	Job status	Progress (%)	Search time	Total results	Creation time (UTC) 07...	Search performed by
Mar 20 - Mar 20 swiw93	Completed	100%	3m, 21s	9	Mar 20, 2025 2:50 PM	swiw93@woodgrove.ms

4 items

5. Export the data.

Audit > Audit search

Search Query Information: Thu, 20 Mar 2025 00:00:00 GMT to Thu, 20 Mar 2025 23:30:00 GMT , swiw93@woodgrove.ms , ExchangeAdmin, ExchangeAggregatedOperation, ExchangeItem, ExchangeItemAggregated, ExchangeItemGroup, ExchangeSearch ,

Total Result Count: 9 items

Export 9 items Filter

Date (UTC)	IP Address	User	Record Type	Activity	Item	Admin Units	Details
Mar 20, 2025 8:53 PM	128.94.12.48:14840	swiw93@woodgrove.ms	ExchangeAdmin	Set-MailboxCalendarConf...	4624cc8c-6c94-4593-b0d8...		
Mar 20, 2025 8:18 PM	73.42.163.13	swiw93@woodgrove.ms	ExchangeItemAggregated	Accessed mailbox items		North America - Redmond ...	Mail Items Accessed
Mar 20, 2025 8:16 PM	2603:10b6:806:2a3:18	swiw93@woodgrove.ms	ExchangeItemAggregated	Accessed mailbox items			Mail Items Accessed
Mar 20, 2025 8:16 PM	128.94.12.48	swiw93@woodgrove.ms	ExchangeItemGroup	Moved messages to Delete...		North America - Redmond ...	
Mar 20, 2025 8:16 PM	128.94.12.48	swiw93@woodgrove.ms	ExchangeItemAggregated	Accessed mailbox items		North America - Redmond ...	Mail Items Accessed
Mar 20, 2025 8:16 PM	128.94.12.48	swiw93@woodgrove.ms	ExchangeItemAggregated	Accessed mailbox items		North America - Redmond ...	Mail Items Accessed
Mar 20, 2025 7:50 PM	128.94.12.48	swiw93@woodgrove.ms	ExchangeItemGroup	Moved messages to Delete...		North America - Redmond ...	
Mar 20, 2025 7:50 PM	2603:10b6:806:2a3:18	swiw93@woodgrove.ms	ExchangeItemAggregated	Accessed mailbox items			Mail Items Accessed
Mar 20, 2025 7:50 PM	128.94.12.48	swiw93@woodgrove.ms	ExchangeItemAggregated	Accessed mailbox items		North America - Redmond ...	Mail Items Accessed

6. The log entry has all of the linkable identifiers. You can search by `UniqueId` for each unique activity, and search by `AADSessionId` for all activities within the session.

The screenshot shows the Microsoft 365 Audit Log interface. The search query is set to 'Thu, 20 Mar 2025 00:00:00 GMT to Thu, 20 Mar 2025 23:30:00 GMT' and the search term is 'swiw93@woodgrove.ms'. The results table lists 9 items, showing details like Date (UTC), IP Address, User, Record Type, Activity, Item, and Admin Unit. The 'Activity' column shows actions such as 'Set-MailboxCalendarConfig...', 'Accessed mailbox items', and 'Moved messages to Delete...'. The 'Item' column contains GUIDs. The 'Admin Unit' column shows 'North America' for most entries. To the right of the table, a detailed view of one item is expanded, showing the following JSON object under 'AppAccessContext':

```
{ "AADSessionId": "0030dbe9-5c56-a711-3220-02b85a31a9eb", "IssuedAtTime": "1970-01-01T00:00:00", "UniqueId": "91UWjHsIE2qVtfwIUAAA" }
```

Other sections visible include 'Details', 'Date (UTC)', 'IP Address', 'Users', 'Activity', 'Item', 'Details', 'Admin Units', 'AppAccessContext', 'CreationTime', 'Id', 'Operation', 'OrganizationId', and 'RecordType'. A 'Close' button is at the bottom right of the expanded view.

Related content

[Microsoft Entra certificate-based authentication technical deep dive](#)

Feedback

Was this page helpful?

Yes

No

[Provide product feedback ↗](#)

Authentication Methods Activity

Article • 03/04/2025

The new authentication methods activity dashboard enables admins to monitor authentication method registration and usage across their organization. This reporting capability provides your organization with the means to understand what methods are being registered and how they're being used.

ⓘ Note

For information about viewing or deleting personal data, see [Azure Data Subject Requests for the GDPR](#). For more information about GDPR, see the [GDPR section of the Microsoft Trust Center](#) ↗ and the [GDPR section of the Service Trust portal](#) ↗.

Permissions and licenses

Built-in and custom roles with the following permissions can access the Authentication Methods Activity blade and APIs:

- Microsoft.directory/auditLogs/allProperties/read
- Microsoft.directory/signInReports/allProperties/read

The following roles have the required permissions:

- Reports Reader
- Security Reader
- Global Reader
- Application Administrator
- Cloud Application Administrator
- Security Operator
- Security Administrator
- Global Administrator

A Microsoft Entra ID P1 or P2 license is required to access Usage and insights. Microsoft Entra multifactor authentication and self-service password reset (SSPR) licensing information can be found on the [Microsoft Entra pricing site](#) ↗.

How it works

To access authentication method Usage and insights:

1. Sign in to the Microsoft Entra admin center [↗](#) as at least an **Authentication Policy Administrator**.
2. Browse to **Protection > Authentication Methods > Activity**.
3. There are two tabs in the report: **Registration** and **Usage**.

The screenshot shows the Microsoft Entra admin center interface. The left sidebar includes sections like Home, Favorites, Identity, Overview, Users, Groups, Devices, Applications, Enterprise applications, App registrations, Protection, Identity Protection, Conditional Access, and Authentication methods. The main content area is titled "Authentication methods | Activity" for the "Contoso - Microsoft Entra ID Security" tenant. A search bar is at the top. Below it, there are two tabs: "Registration" (which is highlighted with a red box) and "Usage". The "Registration" tab displays three boxes of information:

- "Users capable of Azure multifactor authentication": 0 of 34 total. A note says "100% of your organization isn't capable."
- "Users capable of passwordless authentication": 0 of 34 total. A note says "100% of your organization isn't capable."
- "Users capable of self-service password reset": 6 of 34 total. A note says "82% of your organization isn't enabled."

Registration details

You can access the **Registration** tab to show the number of users capable of multifactor authentication, passwordless authentication, and self-service password reset.

Click any of the following options to pre-filter a list of user registration details:

- **Users capable of Azure multifactor authentication** shows the breakdown of users who are both:
 - Registered for a strong authentication method
 - Enabled by policy to use that method for MFA

This number doesn't reflect users registered for MFA outside of Microsoft Entra ID.

- **Users capable of passwordless authentication** shows the breakdown of users who are registered to sign in without a password by using FIDO2, Windows Hello for Business, or passwordless Phone sign-in with the Microsoft Authenticator app.
- **Users capable of self-service password reset** shows the breakdown of users who can reset their passwords. Users can reset their password if they're both:
 - Registered for enough methods to satisfy their organization's policy for self-service password reset
 - Enabled to reset their password

Authentication methods | Activity

Contoso - Microsoft Entra ID Security

Search <> Registration Usage

Manage

- Policies
- Password protection
- Registration campaign
- Authentication strengths
- Settings

Monitoring

- Activity
- User registration details
- Registration and reset events
- Bulk operation results

Users capable of Azure multifactor authentication

0 of 34 total

✖ 100% of your organization isn't capable.

Users capable of passwordless authentication

0 of 34 total

✖ 100% of your organization isn't capable.

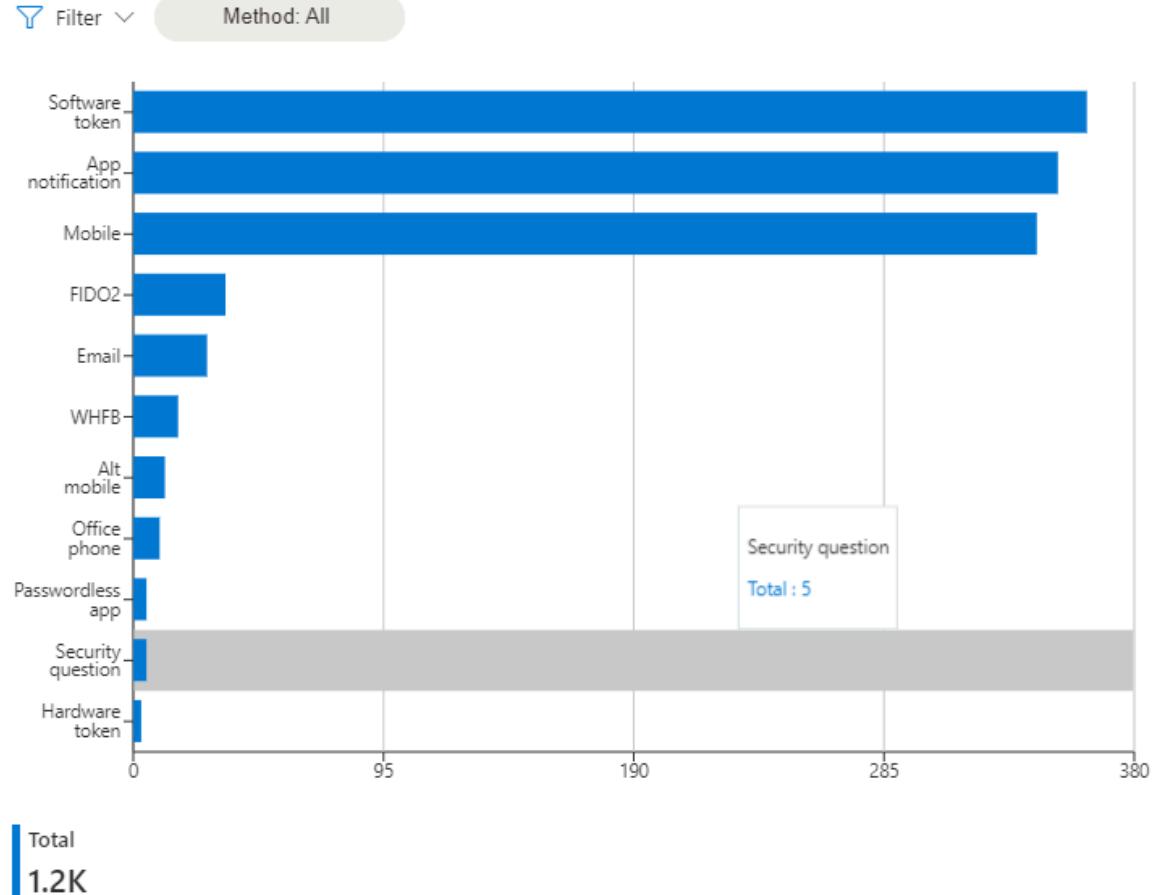
Users capable of self-service password reset

6 of 34 total

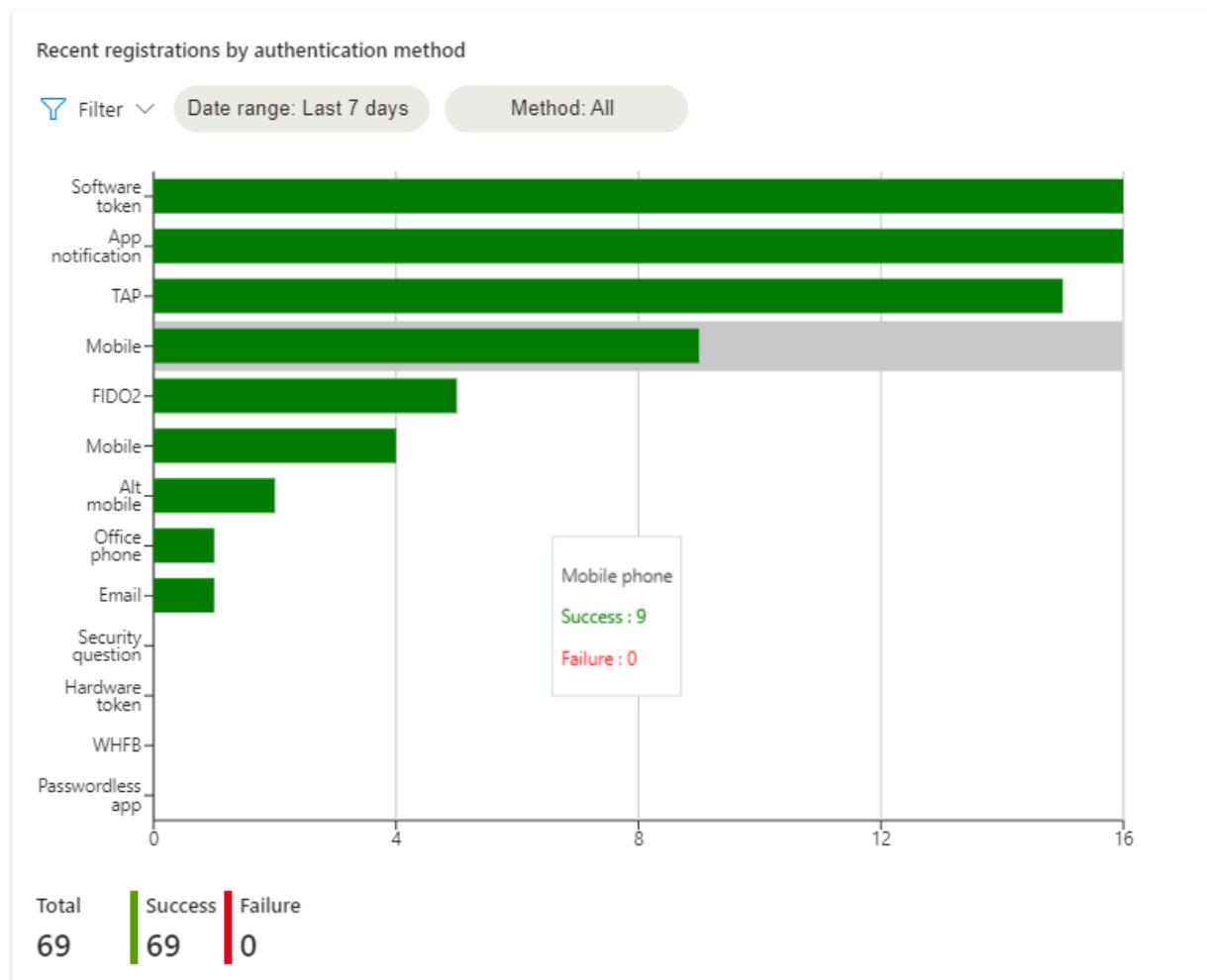
✖ 82% of your organization isn't enabled.

Users registered by authentication method shows how many users are registered for each authentication method. Click an authentication method to see who is registered for that method.

Users registered by authentication method



Recent registration by authentication method shows how many registrations succeeded and failed, sorted by authentication method. Click an authentication method to see recent registration events for that method.

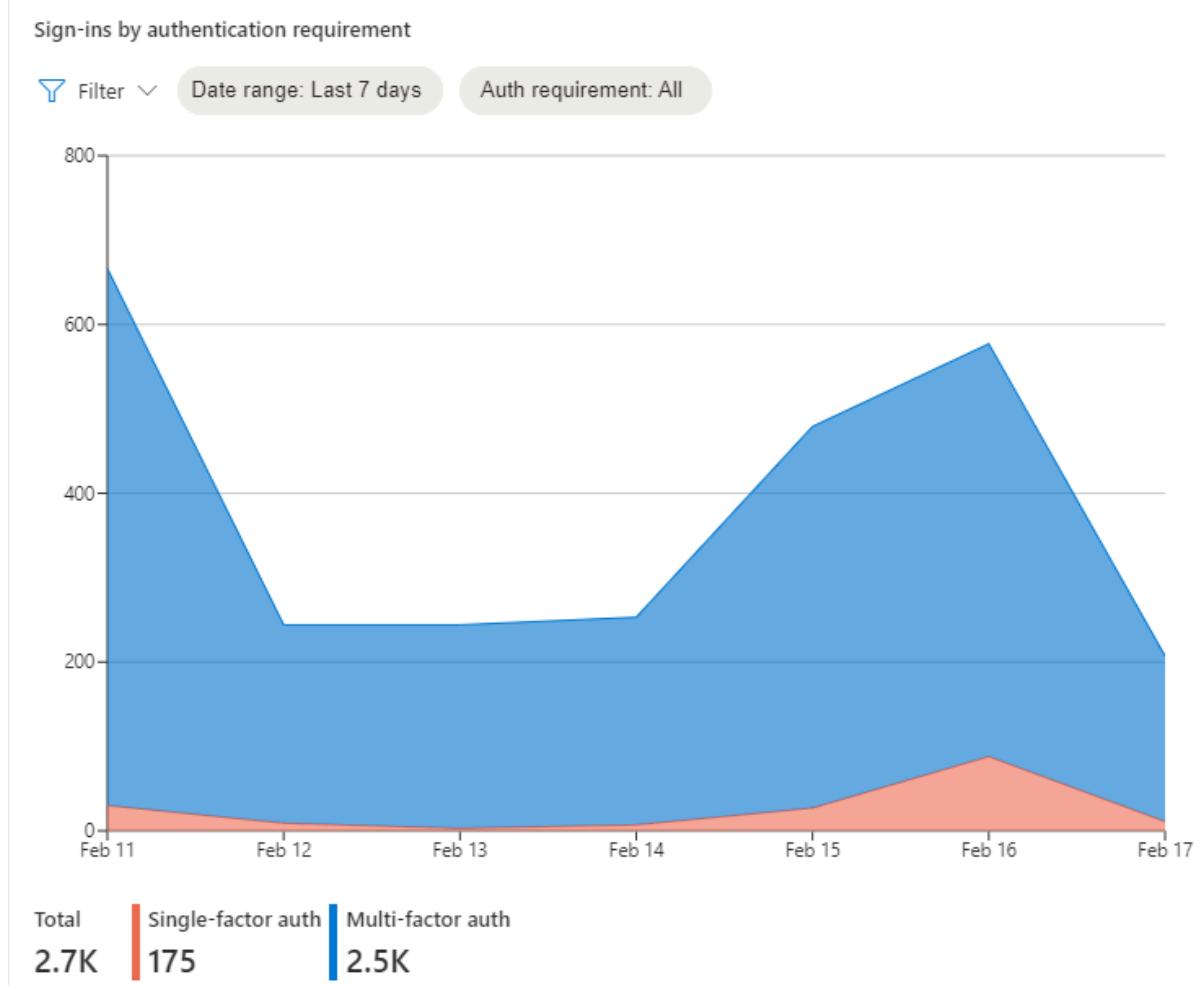


Usage details

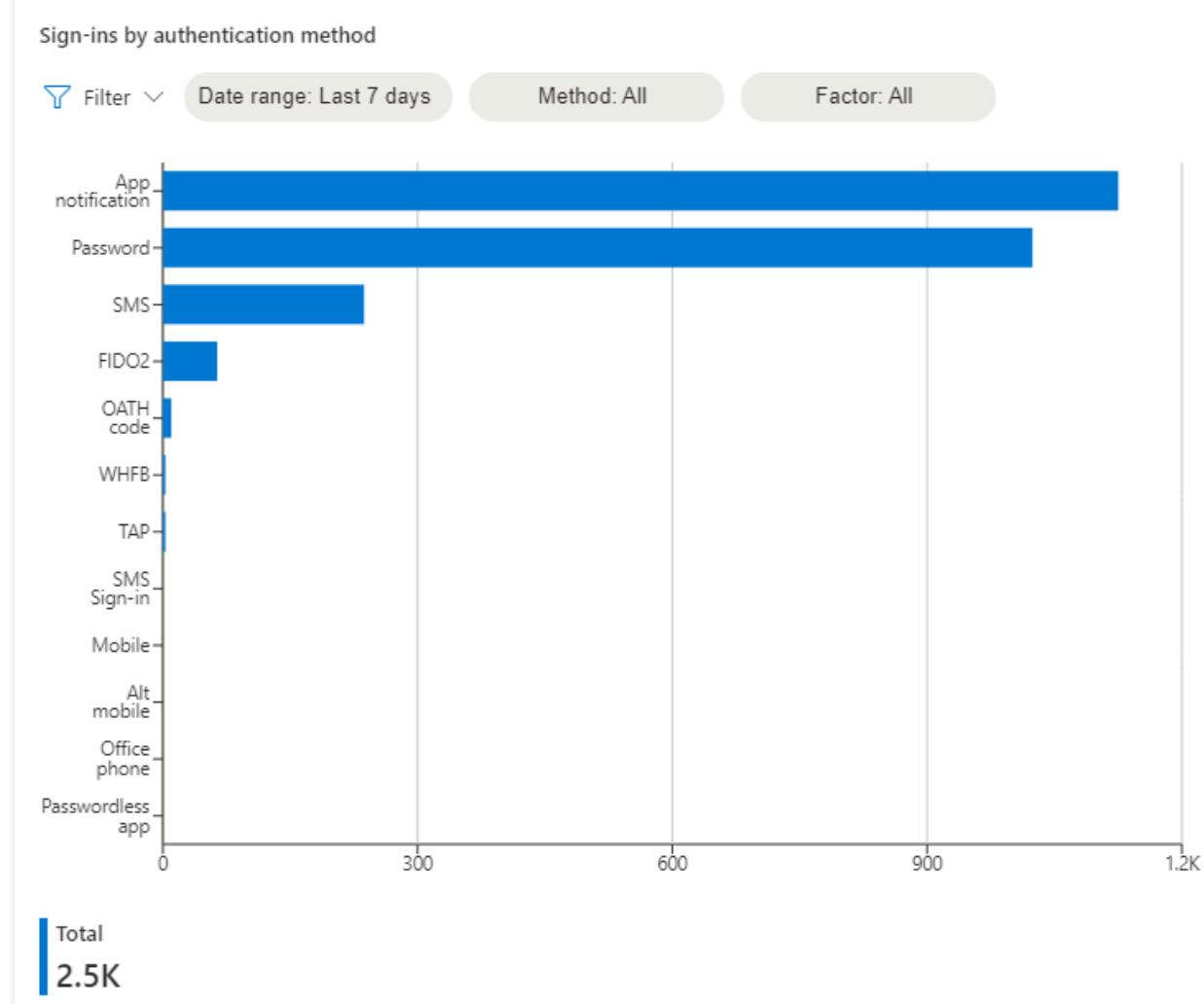
The **Usage** report shows which authentication methods are used to sign-in and reset passwords.



Sign-ins by authentication requirement shows the number of successful user interactive sign-ins that were required for single-factor versus multifactor authentication in Microsoft Entra ID. Sign-ins where MFA was enforced by a third-party MFA provider are not included.



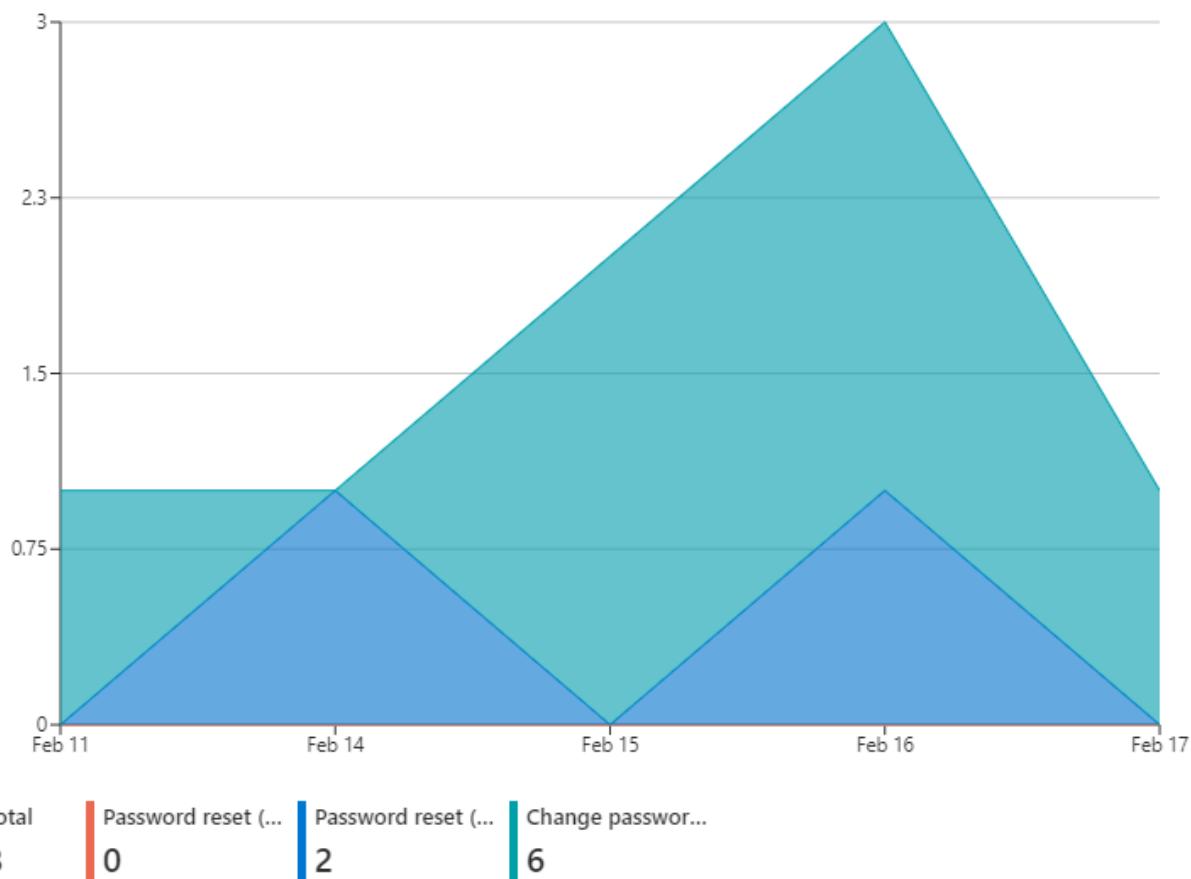
Sign-ins by authentication method shows the number of user interactive sign-ins (success and failure) by authentication method used. It doesn't include sign-ins where the authentication requirement was satisfied by a claim in the token.



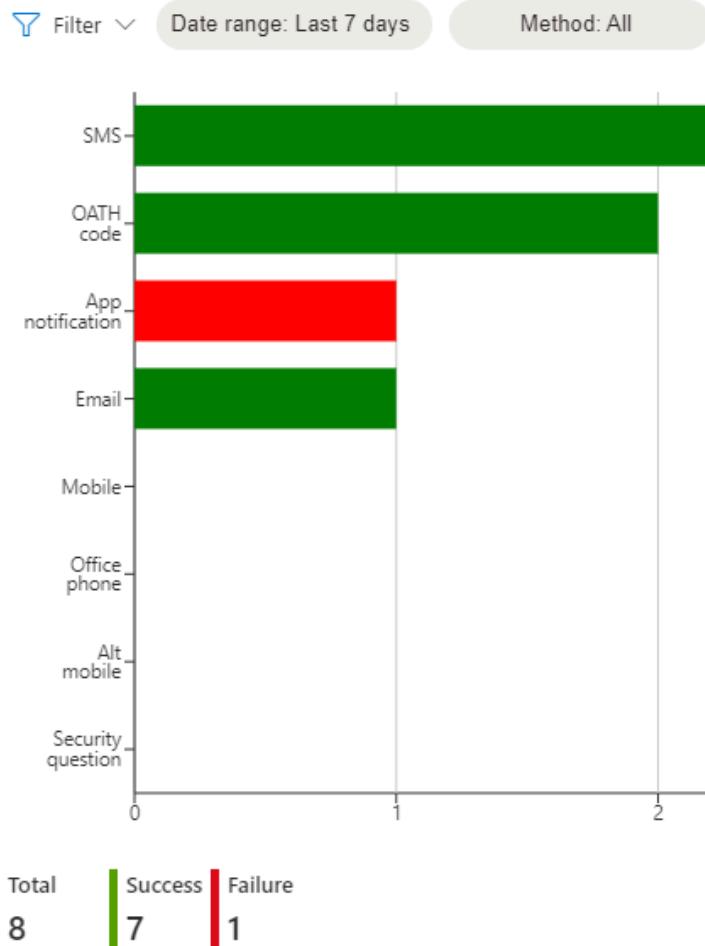
Number of password resets and account unlocks shows the number of successful password changes and password resets (self-service and by admin) over time.

Number of password changes and resets

Filter Date range: Last 7 days



Password resets by authentication method shows the number of successful and failed authentications during the password reset flow by authentication method.



User registration details

Using the controls at the top of the list, you can search for a user and filter the list of users based on the columns shown.

ⓘ Note

User accounts that were recently deleted, also known as [soft-deleted users](#), are not listed in user registration details.

The registration details report shows the following information for each user:

- User principal name
- Name
- MFA Capable (Capable, Not Capable)
- Passwordless Capable (Capable, Not Capable)
- SSPR Registered (Registered, Not Registered)

- SSPR Enabled (Enabled, Not Enabled)
- SSPR Capable (Capable, Not Capable)
- Methods registered (Alternate Mobile Phone, Certificate-based authentication, Email, FIDO2 security key, Hardware OATH token, Microsoft Authenticator app, Microsoft Passwordless phone sign-in, Mobile phone, Office phone, Security questions, Software OATH token, Temporary Access Pass, Windows Hello for Business)
- Last Updated Time (The date and time when the report most recently updated. This value is not related the user's authentication method registration.)

UPN	Name	Multifactor authen...	Passwordless Ca...	SSPR Capable	Default multifactor authen...	Methods Registered	Last Updated
admin@contoso.com	Conf Room ...	Not Capable	Not Capable	Not Capable	Not Capable		2/26/24, 2:05:31
admin@contoso.com	A	Not Capable	Not Capable	Not Capable	Not Capable		2/26/24, 2:05:31
admin@contoso.com		Not Capable	Not Capable	Not Capable	Not Capable		2/26/24, 2:05:31
admin@contoso.com		Not Capable	Not Capable	Not Capable	Not Capable		2/26/24, 2:05:31
admin@contoso.com		Not Capable	Not Capable	Not Capable	Not Capable		2/26/24, 2:05:31
admin@contoso.com		Not Capable	Not Capable	Not Capable	Not Capable		2/26/24, 2:05:31
admin@contoso.com		Not Capable	Not Capable	Not Capable	Not Capable		2/26/24, 2:05:31

Registration and reset events

Registration and reset events shows registration and reset events from the last 24 hours, last seven days, or last 30 days including:

- Date
- User name
- User
- Feature (Registration, Reset)
- Method used (App notification, App code, Phone Call, Office Call, Alternate Mobile Call, SMS, Email, Security questions)
- Status (Success, Failure)
- Reason for failure (explanation)

Authentication methods | Registration and reset events

Date	UPN	Name	Activity type	Method used	Status
Feb 17, 2021, 11:29 PM			Reset	Microsoft Authenticator app (p...)	Failure
Feb 17, 2021, 11:29 PM			Reset	SMS	Success
Feb 17, 2021, 9:10 PM			Registration	FIDO2 security key	Success
Feb 17, 2021, 8:58 PM			Registration	FIDO2 security key	Success
Feb 17, 2021, 8:55 PM			Registration	Temporary Access Pass	Success
Feb 17, 2021, 2:42 PM			Registration	Mobile phone	Success
Feb 17, 2021, 2:41 PM			Registration	Microsoft Authenticator app (p...)	Success
Feb 17, 2021, 2:41 PM			Registration	Software OATH token	Success
Feb 17, 2021, 2:31 PM			Registration	Temporary Access Pass	Success
Feb 17, 2021, 12:09 PM			Registration	Mobile phone	Success
Feb 17, 2021, 12:09 PM			Registration	Mobile phone	Success
Feb 17, 2021, 12:08 PM			Registration	Mobile phone	Success
Feb 17, 2021, 12:08 PM			Registration	Software OATH token	Success
Feb 17, 2021, 12:08 PM			Registration	Microsoft Authenticator app (p...)	Success

Limitations

- The data in the report is not updated in real-time and may reflect a latency of up to a few hours.
- The **PhoneAppNotification** or **PhoneAppOTP** methods that a user might have configured are not displayed in the dashboard on **Microsoft Entra authentication methods - Policies**.
- Bulk operations in the Microsoft Entra admin portal could time out and fail on very large tenants. This limitation is a known issue due to scaling limitations. For more information, see [Bulk operations](#).

Next steps

- [Working with the authentication methods usage report API](#)
- [Choosing authentication methods for your organization](#)
- [Combined registration experience](#)

Feedback

Was this page helpful?

Yes

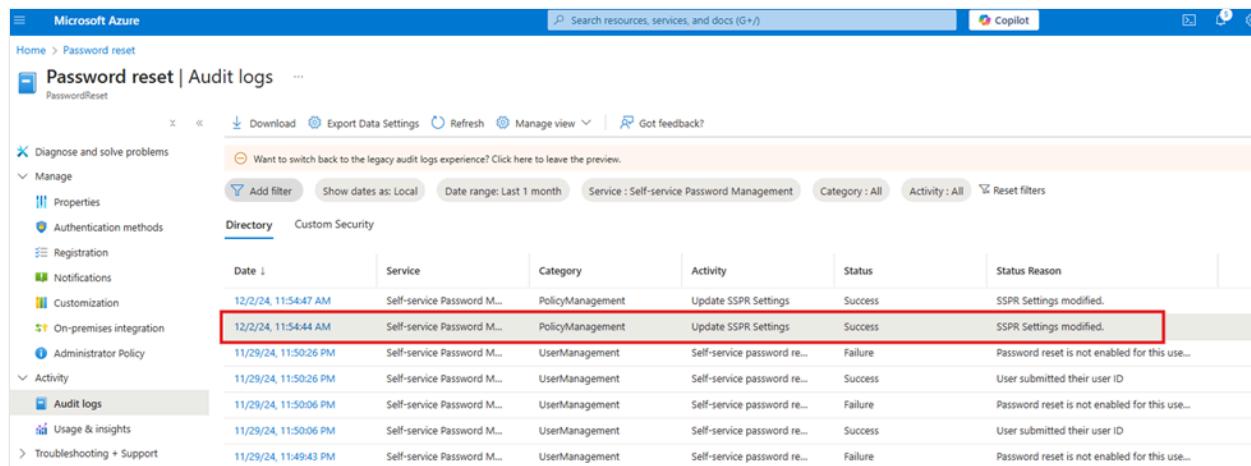
No

[Provide product feedback ↗](#)

Reporting options for Microsoft Entra password management

Article • 03/04/2025

After deployment, many organizations want to know how or if self-service password reset (SSPR) is really being used. The reporting feature that Microsoft Entra ID provides helps you answer questions by using prebuilt reports. If you're appropriately licensed, you can also create custom queries.



Date	Service	Category	Activity	Status	Status Reason
12/24/23, 11:54:47 AM	Self-service Password M...	PolicyManagement	Update SSPR Settings	Success	SSPR Settings modified.
12/24/23, 11:54:44 AM	Self-service Password M...	PolicyManagement	Update SSPR Settings	Success	SSPR Settings modified.
11/29/23, 11:50:26 PM	Self-service Password M...	UserManagement	Self-service password re...	Failure	Password reset is not enabled for this user...
11/29/23, 11:50:26 PM	Self-service Password M...	UserManagement	Self-service password re...	Success	User submitted their user ID
11/29/23, 11:50:06 PM	Self-service Password M...	UserManagement	Self-service password re...	Failure	Password reset is not enabled for this user...
11/29/23, 11:50:06 PM	Self-service Password M...	UserManagement	Self-service password re...	Success	User submitted their user ID
11/29/23, 11:49:43 PM	Self-service Password M...	UserManagement	Self-service password re...	Failure	Password reset is not enabled for this user...

The following questions can be answered by the reports that exist in the [Microsoft Entra admin center](#):

ⓘ Note

You must opt in for this data to be gathered on behalf of your organization. To opt in, you must visit the **Reporting** tab or the audit logs at least once. Until then, data isn't collected for your organization.

- What changes were made to the SSPR policy?
- How many people have registered for password reset?
- Who has registered for password reset?
- What data are people registering?
- How many people reset their passwords in the last seven days?
- What are the most common methods that users or admins use to reset their passwords?
- What are common problems users or admins face when attempting to use password reset?
- What admins are resetting their own passwords frequently?
- Is there any suspicious activity going on with password reset?

How to view password management reports

Use the following steps to find the password reset and password reset registration events:

1. Sign in to the [Microsoft Entra admin center](#) as at least a **Reports Reader**.
2. Browse to **Identity > Users**.
3. Select **Audit Logs** from the **Users** blade. This shows you all of the audit events that occurred against all the users in your directory. You can filter this view to see all the password-related events.
4. From the **Filter** menu at the top of the pane, select the **Service** drop-down list, and change it to the **Self-service Password Management** service type.
5. Optionally, further filter the list by choosing the specific **Activity** you're interested in.

Combined registration

[Combined registration](#) security information registration and management events can be found in the audit logs under **Security > Authentication Methods**.

Description of the report columns

The following list explains each of the report columns in detail:

- **User:** The user who attempted a password reset registration operation.
- **Role:** The role of the user in the directory.
- **Date and Time:** The date and time of the attempt.
- **Data Registered:** The authentication data that the user provided during password reset registration.

Description of the report values

The following table describes the different values that you can set for each column:

[+] Expand table

Column	Permitted values and their meanings
Data registered	Alternate email: The user used an alternate email or authentication email to authenticate. Office phone: The user used an office phone to authenticate.

Column	Permitted values and their meanings
	<p>Mobile phone: The user used a mobile phone or authentication phone to authenticate.</p> <p>Security questions: The user used security questions to authenticate.</p> <p>Any combination of the previous methods, for example, alternate email + mobile phone: Occurs when a two-gate policy is specified and shows which two methods the user used to authentication their password reset request.</p>

Self-Service Password Management activity types

The following activity types appear in the **Self-Service Password Management** audit event category:

- [Self-service password reset policy changes](#): Indicates any changes to the SSPR policy, including old value and new value.
- [Blocked from self-service password reset](#): Indicates that a user tried to reset a password, use a specific gate, or validate a phone number more than five total times in 24 hours.
- [Change password \(self-service\)](#): Indicates that a user performed a voluntary, or forced (due to expiry) password change.
- [Reset password \(by admin\)](#): Indicates that an administrator performed a password reset on behalf of a user.
- [Reset password \(self-service\)](#): Indicates that a user successfully reset their password from [Microsoft Entra password reset](#).
- [Self-service password reset flow activity progress](#): Indicates each specific step a user proceeds through, such as passing a specific password reset authentication gate, as part of the password reset process.
- [Unlock user account \(self-service\)](#): Indicates that a user successfully unlocked their Active Directory account without resetting their password from [Microsoft Entra password reset](#) by using the Active Directory feature of account unlock without reset.
- [User registered for self-service password reset](#): Indicates that a user has registered all the required information to be able to reset their password in accordance with the currently specified tenant password reset policy.

Activity type: Self-service password reset policy changes

The following list explains this activity in detail:

- **Activity description:** Indicates that an administrator updated settings in the SSPR policy.
- **Activity actor:** The display name and user principal name (UPN) of the administrator who made the changes.
- **Activity target:** Updated properties in the SSPR policy.
- **Activity statuses:**
 - *Success:* Indicates that a user successfully changed a setting in the SSPR policy.
 - *Failure:* Indicates that a user failed to change a setting in the SSPR policy.
- **Activity status failure reason:**
 - Permission failure?

Activity type: Blocked from self-service password reset

The following list explains this activity in detail:

- **Activity description:** Indicates that a user tried to reset a password, use a specific gate, or validate a phone number more than five total times in 24 hours.
- **Activity actor:** The user who was throttled from performing additional reset operations. The user can be an end user or an administrator.
- **Activity target:** The user who was throttled from performing additional reset operations. The user can be an end user or an administrator.
- **Activity status:**
 - *Success:* Indicates that a user was throttled from performing any additional resets, attempting any additional authentication methods, or validating any additional phone numbers for the next 24 hours.
- **Activity status failure reason:** Not applicable.

Activity type: Change password (self-service)

The following list explains this activity in detail:

- **Activity description:** Indicates that a user performed a voluntary, or forced (due to expiry) password change.
- **Activity actor:** The user who changed their password. The user can be an end user or an administrator.
- **Activity target:** The user who changed their password. The user can be an end user or an administrator.
- **Activity statuses:**
 - *Success:* Indicates that a user successfully changed their password.
 - *Failure:* Indicates that a user failed to change their password. You can select the row to see the **Activity status reason** category to learn more about why the

failure occurred.

- **Activity status failure reason:**
 - *FuzzyPolicyViolationInvalidPassword*: The user selected a password that was automatically banned because the Microsoft Banned Password Detection capabilities found it to be too common or especially weak.

Activity type: Reset password (by admin)

The following list explains this activity in detail:

- **Activity description:** Indicates that an administrator performed a password reset on behalf of a user.
- **Activity actor:** The administrator who performed the password reset on behalf of another end user or administrator. Must be a password administrator, user administrator, or helpdesk administrator.
- **Activity target:** The user whose password was reset. The user can be an end user or a different administrator.
- **Activity statuses:**
 - *Success*: Indicates that an admin successfully reset a user's password.
 - *Failure*: Indicates that an admin failed to change a user's password. You can select the row to see the **Activity status reason** category to learn more about why the failure occurred.
- **Activity additional details OnPremisesAgent:**
 - *None*: Indicates cloud-only reset.
 - *Microsoft Entra Connect*: Indicates password was reset on-premises via Microsoft Entra Connect writeback agent.
 - *CloudSync*: Indicates password was reset on-premises via Microsoft Entra CloudSync writeback agent.

Activity type: Reset password (self-service)

The following list explains this activity in detail:

- **Activity description:** Indicates that a user successfully reset their password from [Microsoft Entra password reset](#).
- **Activity actor:** The user who reset their password. The user can be an end user or an administrator.
- **Activity target:** The user who reset their password. The user can be an end user or an administrator.
- **Activity statuses:**
 - *Success*: Indicates that a user successfully reset their own password.

- *Failure*: Indicates that a user failed to reset their own password. You can select the row to see the **Activity status reason** category to learn more about why the failure occurred.
- **Activity status failure reason:**
 - *FuzzyPolicyViolation/InvalidPassword*: The admin selected a password that was automatically banned because the Microsoft Banned Password Detection capabilities found it to be too common or especially weak.

Activity type: Self-service password reset flow activity progress

The following list explains this activity in detail:

- **Activity description:** Indicates each specific step a user proceeds through (such as passing a specific password reset authentication gate) as part of the password reset process.
- **Activity actor:** The user who performed part of the password reset flow. The user can be an end user or an administrator.
- **Activity target:** The user who performed part of the password reset flow. The user can be an end user or an administrator.
- **Activity statuses:**
 - *Success*: Indicates that a user successfully completed a specific step of the password reset flow.
 - *Failure*: Indicates that a specific step of the password reset flow failed. You can select the row to see the **Activity status reason** category to learn more about why the failure occurred.
- **Activity status reasons:** See the following table for [all the permissible reset activity status reasons](#).

Activity type: Unlock a user account (self-service)

The following list explains this activity in detail:

- **Activity description:** Indicates that a user successfully unlocked their Active Directory account without resetting their password from [Microsoft Entra password reset](#) by using the Active Directory feature of account unlock without reset.
- **Activity actor:** The user who unlocked their account without resetting their password. The user can be an end user or an administrator.
- **Activity target:** The user who unlocked their account without resetting their password. The user can be an end user or an administrator.
- **Allowed activity statuses:**

- *Success*: Indicates that a user successfully unlocked their own account.
- *Failure*: Indicates that a user failed to unlock their account. You can select the row to see the **Activity status reason** category to learn more about why the failure occurred.

Activity type: User registered for self-service password reset

The following list explains this activity in detail:

- **Activity description**: Indicates that a user has registered all the required information to be able to reset their password in accordance with the currently specified tenant password reset policy.
- **Activity actor**: The user who registered for password reset. The user can be an end user or an administrator.
- **Activity target**: The user who registered for password reset. The user can be an end user or an administrator.
- **Allowed activity statuses**:
 - *Success*: Indicates that a user successfully registered for password reset in accordance with the current policy.
 - *Failure*: Indicates that a user failed to register for password reset. You can select the row to see the **Activity status reason** category to learn more about why the failure occurred.

ⓘ Note

Failure doesn't mean a user is unable to reset their own password. It means that they didn't finish the registration process. If there's unverified data on their account that's correct, such as a phone number that's not validated, even though they haven't verified this phone number, they can still use it to reset their password.

Next steps

- [SSPR and MFA usage and insights reporting](#)
- [How do I complete a successful rollout of SSPR?](#)
- [Reset or change your password ↗](#).
- [Register for self-service password reset ↗](#).

- Do you have a licensing question?
 - What data is used by SSPR and what data should you populate for your users?
 - What authentication methods are available to users?
 - What are the policy options with SSPR?
 - What is password writeback and why do I care about it?
 - What are all of the options in SSPR and what do they mean?
 - I think something is broken. How do I troubleshoot SSPR?
 - I have a question that was not covered somewhere else
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Use the sign-in logs to review Microsoft Entra multifactor authentication events

Article • 04/24/2025

To review and understand Microsoft Entra multifactor authentication events, you can use the Microsoft Entra sign-in logs. This report shows authentication details for events when a user is prompted for multifactor authentication, and if any Conditional Access policies were in use. For detailed information on the sign-in logs, see the [overview of sign-in activity reports in Microsoft Entra ID](#).

View the Microsoft Entra sign-in logs

The sign-in logs provides you with information about the usage of managed applications and user sign-in activities, which includes information about multifactor authentication usage. The MFA data gives you insights into how MFA is working in your organization. It answers questions like:

- Was the sign-in challenged with MFA?
- How did the user complete MFA?
- Which authentication methods were used during a sign-in?
- Why was the user unable to complete MFA?
- How many users are challenged for MFA?
- How many users are unable to complete the MFA challenge?
- What are the common MFA issues end users are running into?

To view the sign-in activity report in the [Microsoft Entra admin center](#), complete the following steps. You can also query data using the [reporting API](#).

1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Policy Administrator](#).
2. Browse to **Entra ID > Users** from the menu on the left-hand side.
3. From the menu on the left-hand side, select **Sign-in logs**.
4. A list of sign-in events is shown, including the status. You can select an event to view more details.

The **Conditional Access** tab of the event details shows you which policy triggered the MFA prompt.

The screenshot shows the Microsoft Entra ID interface for managing users. On the left, there's a sidebar with links like 'All users', 'Audit logs', and 'Sign-in logs' (which is highlighted with a red box). The main area displays a table of sign-in logs with columns for Date, Request ID, User, Application, Status, IP address, and Location. Each log entry shows a successful sign-in from the Azure Portal by a user named MOD Administrator.

If available, the authentication is shown, such as text message, Microsoft Authenticator app notification, or phone call.

The **Authentication Details** tab provides the following information, for each authentication attempt:

- A list of authentication policies applied (such as Conditional Access, per-user MFA, Security Defaults)
- The sequence of authentication methods used to sign-in
- Whether or not the authentication attempt was successful
- Detail about why the authentication attempt succeeded or failed

This information allows admins to troubleshoot each step in a user's sign-in, and track:

- Volume of sign-ins protected by multifactor authentication
- Usage and success rates for each authentication method
- Usage of passwordless authentication methods (such as Passwordless Phone Sign-in, FIDO2, and Windows Hello for Business)
- How frequently authentication requirements are satisfied by token claims (where users aren't interactively prompted to enter a password, enter an SMS OTP, and so on)

While viewing the sign-in logs, select the **Authentication Details** tab:

This screenshot shows a detailed view of a single sign-in log. At the top, it lists basic info like date (4/13/2021), user (admin), location (Office 365 SharePoint Online), and result (Success). Below this, the 'Authentication Details' tab is selected (highlighted with a red box). It shows a table with columns for Authentication Policies Applied, Date, Authentication method, Authentication method detail, Succeeded, Result detail, and Requirement. Two rows of data are visible: one for a previously satisfied requirement and another for a mobile app notification.

! Note

OATH verification code is logged as the authentication method for both OATH hardware and software tokens (such as the Microsoft Authenticator app).

Important

The **Authentication details** tab can initially show incomplete or inaccurate data, until log information is fully aggregated. Known examples include:

- A **satisfied by claim in the token** message is incorrectly displayed when sign-in events are initially logged.
- The **Primary authentication** row isn't initially logged.

The following details are shown on the **Authentication Details** window for a sign-in event that show if the MFA request was satisfied or denied:

- If MFA was satisfied, this column provides more information about how MFA was satisfied.
 - completed in the cloud
 - has expired due to the policies configured on tenant
 - registration prompted
 - satisfied by claim in the token
 - satisfied by claim provided by external provider
 - satisfied by strong authentication
 - skipped as flow exercised was Windows broker logon flow
 - skipped due to app password
 - skipped due to location
 - skipped due to registered device
 - skipped due to remembered device
 - successfully completed
- If MFA was denied, this column would provide the reason for denial.
 - authentication in-progress
 - duplicate authentication attempt
 - entered incorrect code too many times
 - invalid authentication
 - invalid mobile app verification code
 - misconfiguration
 - phone call went to voicemail
 - phone number has an invalid format
 - service error
 - unable to reach the user's phone

- unable to send the mobile app notification to the device
- unable to send the mobile app notification
- user declined the authentication
- user didn't respond to mobile app notification
- user doesn't have any verification methods registered
- user entered incorrect code
- user entered incorrect PIN
- user hung up the phone call without succeeding the authentication
- user is blocked
- user never entered the verification code
- user not found
- verification code already used once

PowerShell reporting on users registered for MFA

First, ensure that you have the [Install the Microsoft Graph PowerShell SDK](#) installed.

Identify users who have registered for MFA using the PowerShell that follows. This set of commands excludes disabled users since these accounts can't authenticate against Microsoft Entra ID:

PowerShell

```
Get-MgUser -All | Where-Object {$_ .StrongAuthenticationMethods -ne $null -and
$_ .BlockCredential -eq $False} | Select-Object -Property UserPrincipalName
```

Identify users who aren't registered for MFA by running the following PowerShell commands. This set of commands excludes disabled users since these accounts can't authenticate against Microsoft Entra ID:

PowerShell

```
Get-MgUser -All | Where-Object {$_ .StrongAuthenticationMethods.Count -eq 0 -and
$_ .BlockCredential -eq $False} | Select-Object -Property UserPrincipalName
```

Identify users and output methods registered:

PowerShell

```
Get-MgUser -All | Select-Object @{N='UserPrincipalName';E=
{$_.UserPrincipalName}},@{N='MFA Status';E={if
($_ .StrongAuthenticationRequirements.State)
{$_ .StrongAuthenticationRequirements.State} else {"Disabled"} }},@{N='MFA
```

```
Methods';E=$_.StrongAuthenticationMethods.methodtype}} | Export-Csv -Path  
c:\MFA_Report.csv -NoTypeInformation
```

Additional MFA reports

NPS extension and AD FS adapter for cloud MFA activity are now included in the Sign-in logs and not with a specific activity report.

Cloud MFA sign-in events from an on-premises AD FS adapter or NPS extension won't have all fields in the sign-in logs populated due to limited data returned by the on-premises component. You can identify these events by the resourceId *adfs* or *radius* in the event properties. They include:

- resultSignature
- appID
- deviceDetail
- conditionalAccessStatus
- authenticationContext
- isInteractive
- tokenIssuerName
- riskDetail, riskLevelAggregated,riskLevelDuringSignIn, riskState,riskEventTypes, riskEventTypes_v2
- authenticationProtocol
- incomingTokenType

Organizations that run the latest version of NPS extension or use Microsoft Entra Connect Health will have location IP address in events.

Next steps

This article provided an overview of the sign-ins activity report. For more detailed information on what this report contains, see [sign-in activity reports in Microsoft Entra ID](#).

Microsoft Entra user data collection for multifactor authentication and self-service password reset

Article • 03/04/2025

This article explains how to find user information collected by Microsoft Entra multifactor authentication (Cloud-based) and self-service password reset (SSPR) in the event you would like to remove it.

Note

For information about viewing or deleting personal data, please review Microsoft's guidance on the [Windows data subject requests for the GDPR](#) site. For general information about GDPR, see the [GDPR section of the Microsoft Trust Center](#) and the [GDPR section of the Service Trust portal](#).

MFA information collected

MFA Server, the NPS Extension, and the Windows Server 2016 Microsoft Entra multifactor authentication AD FS Adapter collect and store the following information for 90 days.

Authentication Attempts (used for reporting and troubleshooting):

- Timestamp
- Username
- First Name
- Last Name
- Email Address
- User Group
- Authentication Method (Phone Call, Text Message, Mobile App, OATH Token)
- Phone Call Mode (Standard, PIN)
- Text Message Direction (One-Way, Two-Way)
- Text Message Mode (OTP, OTP + PIN)
- Mobile App Mode (Standard, PIN)
- OATH Token Mode (Standard, PIN)
- Authentication Type
- Application Name

- Primary Call Country Code
- Primary Call Phone Number
- Primary Call Extension
- Primary Call Authenticated
- Primary Call Result
- Backup Call Country Code
- Backup Call Phone Number
- Backup Call Extension
- Backup Call Authenticated
- Backup Call Result
- Overall Authenticated
- Overall Result
- Results
- Authenticated
- Result
- Initiating IP Address
- Devices
- Device Token
- Device Type
- Mobile App Version
- OS Version
- Result
- Used Check for Notification

Activations (attempts to activate an account in the Microsoft Authenticator mobile app):

- Username
- Account Name
- Timestamp
- Get Activation Code Result
- Activate Success
- Activate Error
- Activation Status Result
- Device Name
- Device Type
- App Version
- OATH Token Enabled

Blocks (used to determine blocked state and for reporting):

- Block Timestamp
- Block By Username

- Username
- Country Code
- Phone Number
- Phone Number Formatted
- Extension
- Clean Extension
- Blocked
- Block Reason
- Completion Timestamp
- Completion Reason
- Account Lockout
- Fraud Alert
- Fraud Alert Not Blocked
- Language

Bypasses (used for reporting):

- Bypass Timestamp
- Bypass Seconds
- Bypass By Username
- Username
- Country Code
- Phone Number
- Phone Number Formatted
- Extension
- Clean Extension
- Bypass Reason
- Completion Timestamp
- Completion Reason
- Bypass Used

Changes (used to sync user changes to MFA Server or Microsoft Entra ID):

- Change Timestamp
- Username
- New Country Code
- New Phone Number
- New Extension
- New Backup Country Code
- New Backup Phone Number
- New Backup Extension
- New PIN

- PIN Change Required
- Old Device Token
- New Device Token

Gather data from NPS extension

Use the Microsoft Privacy portal to make a request for Export.

- MFA information is included in the export, which may take hours or days to complete.
- Occurrences of the username in the AzureMfa/AuthN/AuthNOptCh, AzureMfa/AuthZ/AuthZAdminCh, and AzureMfa/AuthZ/AuthZOptCh event logs are considered operational and duplicative to the information provided in the export.

Delete data from NPS extension

Use the Microsoft Privacy portal to make a request for Account Close to delete all MFA cloud service information collected for this user.

- It may take up to 30 days for data to be fully removed.

Gather data from Microsoft Entra multifactor authentication AD FS adapter

Use the Microsoft Privacy portal to make a request for Export.

- MFA information is included in the export, which may take hours or days to complete.
- Occurrences of the username in the AD FS Tracing/Debug event logs (if enabled) are considered operational and duplicative to the information provided in the export.

Delete data from Microsoft Entra multifactor authentication AD FS adapter

Use the Microsoft Privacy portal to make a request for Account Close to delete all MFA cloud service information collected for this user.

- It may take up to 30 days for data to be fully removed.

Gather data for Microsoft Entra multifactor authentication

Use the Microsoft Privacy portal to make a request for Export.

- MFA information is included in the export, which may take hours or days to complete.

Delete data for Microsoft Entra multifactor authentication

Use the Microsoft Privacy portal to make a request for Account Close to delete all MFA cloud service information collected for this user.

- It may take up to 30 days for data to be fully removed.

Delete data for self-service password reset

Users can add answers to security questions as part of SSPR. Security questions and answers are hashed to prevent unauthorized access. Only the hashed data is saved, so the security questions and answers can't be exported. Users can go to [My sign-ins](#) to edit or delete them. The only other information saved for SSPR is the user email address.

Those assigned the [Privileged Authentication Administrator](#) role can remove data collected for any user. On the [Users](#) page in Microsoft Entra ID, select [Authentication methods](#) and select a user to remove their phone or email address.

Related content

[Authentication methods activity](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Migrate from MFA Server to Microsoft Entra multifactor authentication

Article • 03/04/2025

Multifactor authentication is important to securing your infrastructure and assets from bad actors. Azure Multi-Factor Authentication Server (MFA Server) isn't available for new deployments and is deprecated. Customers who are using MFA Server should move to using cloud-based Microsoft Entra multifactor authentication.

In this article, we assume that you have a hybrid environment where:

- You're using MFA Server for multifactor authentication.
- You're using federation on Microsoft Entra ID with Active Directory Federation Services (AD FS) or another identity provider federation product.
 - While this article is scoped to AD FS, similar steps apply to other identity providers.
- Your MFA Server is integrated with AD FS.
- You might have applications using AD FS for authentication.

There are multiple possible end states to your migration, depending on your goal.

[] Expand table

	Goal: Decommission MFA Server ONLY	Goal: Decommission MFA Server and move to Microsoft Entra authentication	Goal: Decommission MFA Server and AD FS
MFA provider	Change MFA provider from MFA Server to Microsoft Entra multifactor authentication.	Change MFA provider from MFA Server to Microsoft Entra multifactor authentication.	Change MFA provider from MFA Server to Microsoft Entra multifactor authentication.
User authentication	Continue to use federation for Microsoft Entra authentication.	Move to Microsoft Entra ID with Password Hash Synchronization (preferred) or Passthrough Authentication and Seamless single sign-on (SSO).	Move to Microsoft Entra ID with Password Hash Synchronization (preferred) or Passthrough Authentication and SSO.

Goal: Decommission MFA Server ONLY	Goal: Decommission MFA Server and move to Microsoft Entra authentication	Goal: Decommission MFA Server and AD FS
Application authentication	Continue to use AD FS authentication for your applications.	Continue to use AD FS authentication for your applications. Move apps to Microsoft Entra ID before migrating to Microsoft Entra multifactor authentication.

If you can, move both your multifactor authentication and your user authentication to Azure. For step-by-step guidance, see [Moving to Microsoft Entra multifactor authentication and Microsoft Entra user authentication](#).

If you can't move your user authentication, see the step-by-step guidance for [Moving to Microsoft Entra multifactor authentication with federation](#).

Prerequisites

- AD FS environment (required if you don't migrate all your apps to Microsoft Entra before you migrate MFA Server)
 - Upgrade to AD FS for Windows Server 2019, Farm behavior level (FBL) 4. This upgrade enables you to select authentication provider based on group membership for a more seamless user transition. While it's possible to migrate while on AD FS for Windows Server 2016 FBL 3, it isn't as seamless for users. During the migration, users are prompted to select an authentication provider (MFA Server or Microsoft Entra multifactor authentication) until the migration is complete.
- Permissions
 - Enterprise Administrator role in Active Directory to configure AD FS farm for Microsoft Entra multifactor authentication
 - A [Global Administrator](#) is needed to manage this feature.

Considerations for all migration paths

Migrating from MFA Server to Microsoft Entra multifactor authentication involves more than just moving the registered MFA phone numbers. Microsoft's MFA server can be integrated with many systems, and you must evaluate how these systems are using MFA Server to understand the best ways to integrate with Microsoft Entra multifactor authentication.

Migrating MFA user information

Common ways to think about moving users in batches include moving them by regions, departments, or roles such as administrators. You should move user accounts iteratively, starting with test and pilot groups, and make sure you have a rollback plan in place.

You can use the [MFA Server Migration Utility](#) to synchronize MFA data stored in the on-premises Azure MFA Server to Microsoft Entra multifactor authentication and use [Staged Rollout](#) to reroute users to Microsoft Entra multifactor authentication. Staged Rollout helps you test without making any changes to your domain federation settings.

To help users to differentiate the newly added account from the old account linked to the MFA Server, make sure the Account name for the Mobile App on the MFA Server is named in a way to distinguish the two accounts. For example, the Account name that appears under Mobile App on the MFA Server has been renamed to **On-Premises MFA Server**. The account name on Microsoft Authenticator will change with the next push notification to the user.

Migrating phone numbers can also lead to stale numbers being migrated and make users more likely to stay on phone-based MFA instead of setting up more secure methods like Microsoft Authenticator in passwordless mode. We therefore recommend that regardless of the migration path you choose, that you have all users register for [combined security information](#).

Migrating hardware security keys

Microsoft Entra ID provides support for hardware OATH tokens. You can use the [MFA Server Migration Utility](#) to synchronize MFA settings between MFA Server and Microsoft Entra multifactor authentication and use [Staged Rollout](#) to test user migrations without changing domain federation settings.

If you only want to migrate hardware OATH tokens, you need to [upload tokens to Microsoft Entra ID by using a CSV file](#), commonly referred to as a "seed file". The seed file contains the secret keys, token serial numbers, and other necessary information needed to upload the tokens into Microsoft Entra ID.

If you no longer have the seed file with the secret keys, it isn't possible to export the secret keys from MFA Server. If you no longer have access to the secret keys, contact your hardware vendor for support.

The MFA Server Web Service SDK can be used to export the serial number for any OATH tokens assigned to a given user. You can use this information along with the seed file to import the tokens into Microsoft Entra ID and assign the OATH token to the specified

user based on the serial number. The user will also need to be contacted at the time of import to supply OTP information from the device to complete the registration. Refer to the help file topic **GetUserInfo > userSettings > OauthTokenSerialNumber** on your MFA Server.

More migrations

The decision to migrate from MFA Server to Microsoft Entra multifactor authentication opens the door for other migrations. Completing more migrations depends upon many factors, including specifically:

- Your willingness to use Microsoft Entra authentication for users
- Your willingness to move your applications to Microsoft Entra ID

Because MFA Server is integral to both application and user authentication, consider moving both of those functions to Azure as a part of your MFA migration, and eventually decommission AD FS.

Our recommendations:

- Use Microsoft Entra ID for authentication as it enables more robust security and governance
- Move applications to Microsoft Entra ID if possible

To select the best user authentication method for your organization, see [Choose the right authentication method for your Microsoft Entra hybrid identity solution](#). We recommend that you use Password Hash Synchronization (PHS).

Passwordless authentication

As part of enrolling users to use Microsoft Authenticator as a second factor, we recommend you enable passwordless phone sign-in as part of their registration. For more information, including other passwordless methods such as FIDO2 security keys and Windows Hello for Business, visit [Plan a passwordless authentication deployment with Microsoft Entra ID](#).

Microsoft Identity Manager self-service password reset

Microsoft Identity Manager (MIM) SSPR can use MFA Server to invoke SMS one-time passcodes as part of the password reset flow. MIM can't be configured to use Microsoft Entra multifactor authentication. We recommend you evaluate moving your SSPR service to Microsoft Entra SSPR. You can use the opportunity of users registering for Microsoft

Entra multifactor authentication to use the combined registration experience to register for Microsoft Entra SSPR.

If you can't move your SSPR service, or you use MFA Server to invoke MFA requests for Privileged Access Management (PAM) scenarios, we recommend you update to an [alternate third-party MFA option](#).

RADIUS clients and Microsoft Entra multifactor authentication

MFA Server supports RADIUS to invoke multifactor authentication for applications and network devices that support the protocol. If you're using RADIUS with MFA Server, we recommend moving client applications to modern protocols such as SAML, OpenID Connect, or OAuth on Microsoft Entra ID. If the application can't be updated, then you can deploy Network Policy Server (NPS) with the Microsoft Entra multifactor authentication extension. The network policy server (NPS) extension acts as an adapter between RADIUS-based applications and Microsoft Entra multifactor authentication to provide a second factor of authentication. This "adapter" allows you to move your RADIUS clients to Microsoft Entra multifactor authentication and decommission your MFA Server.

Important considerations

There are limitations when using NPS for RADIUS clients, and we recommend evaluating any RADIUS clients to determine if you can upgrade them to modern authentication protocols. Check with the service provider for supported product versions and their capabilities.

- The NPS extension doesn't use Microsoft Entra Conditional Access policies. If you stay with RADIUS and use the NPS extension, all authentication requests going to NPS will require the user to perform MFA.
- Users must register for Microsoft Entra multifactor authentication before using the NPS extension. Otherwise, the extension fails to authenticate the user, which can generate help desk calls.
- When the NPS extension invokes MFA, the MFA request is sent to the user's default MFA method.
 - Because the sign-in happens on non-Microsoft applications, the user often can't see visual notification that multifactor authentication is required and that a request has been sent to their device.
 - During the multifactor authentication requirement, the user must have access to their default authentication method to complete the requirement. They can't

choose an alternative method. Their default authentication method will be used even if it's disabled in the tenant authentication methods and multifactor authentication policies.

- Users can change their default multifactor authentication method in the Security Info page (aka.ms/mysecurityinfo).
- Available MFA methods for RADIUS clients are controlled by the client systems sending the RADIUS access requests.
 - MFA methods that require user input after they enter a password can only be used with systems that support access-challenge responses with RADIUS. Input methods might include OTP, hardware OATH tokens or Microsoft Authenticator.
 - Some systems might limit available multifactor authentication methods to Microsoft Authenticator push notifications and phone calls.

 **Note**

The password encryption algorithm used between the RADIUS client and the NPS system, and the input methods the client can use affect which authentication methods are available. For more information, see [Determine which authentication methods your users can use](#).

Common RADIUS client integrations include applications such as [Remote Desktop Gateways](#) and [VPN Servers](#). Others might include:

- Citrix Gateway
 - [Citrix Gateway](#) supports both RADIUS and NPS extension integration, and a SAML integration.
- Cisco VPN
 - The Cisco VPN supports both RADIUS and [SAML authentication for SSO](#).
 - By moving from RADIUS authentication to SAML, you can integrate the Cisco VPN without deploying the NPS extension.
- All VPNs
 - We recommend federating your VPN as a SAML app if possible. This federation allows you to use Conditional Access. For more information, see a [list of VPN vendors that are integrated into the Microsoft Entra ID App gallery](#).

Resources for deploying NPS

- [Adding new NPS infrastructure](#)
- [NPS deployment best practices](#)
- [Microsoft Entra multifactor authentication NPS extension health check script](#)

- Integrating existing NPS infrastructure with Microsoft Entra multifactor authentication

Next steps

- Moving to Microsoft Entra multifactor authentication with federation
 - Moving to Microsoft Entra multifactor authentication and Microsoft Entra user authentication
 - How to use the MFA Server Migration Utility
-

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Migrate to Microsoft Entra multifactor authentication and Microsoft Entra user authentication

Article • 03/04/2025

Multifactor authentication helps secure your infrastructure and assets from bad actors. Microsoft Multi-Factor Authentication Server (MFA Server) is no longer offered for new deployments. Customers who are using MFA Server should move to Microsoft Entra multifactor authentication (Microsoft Entra multifactor authentication).

There are several options for migrating from MFA Server to Microsoft Entra ID:

- Good: Moving only your [MFA service to Microsoft Entra ID](#).
- Better: Moving your MFA service and user authentication to Microsoft Entra ID, covered in this article.
- Best: Moving all of your applications, your MFA service, and user authentication to Microsoft Entra ID. See the move applications to Microsoft Entra ID section of this article if you plan to move applications, covered in this article.

To select the appropriate MFA migration option for your organization, see the considerations in [Migrate from MFA Server to Microsoft Entra multifactor authentication](#).

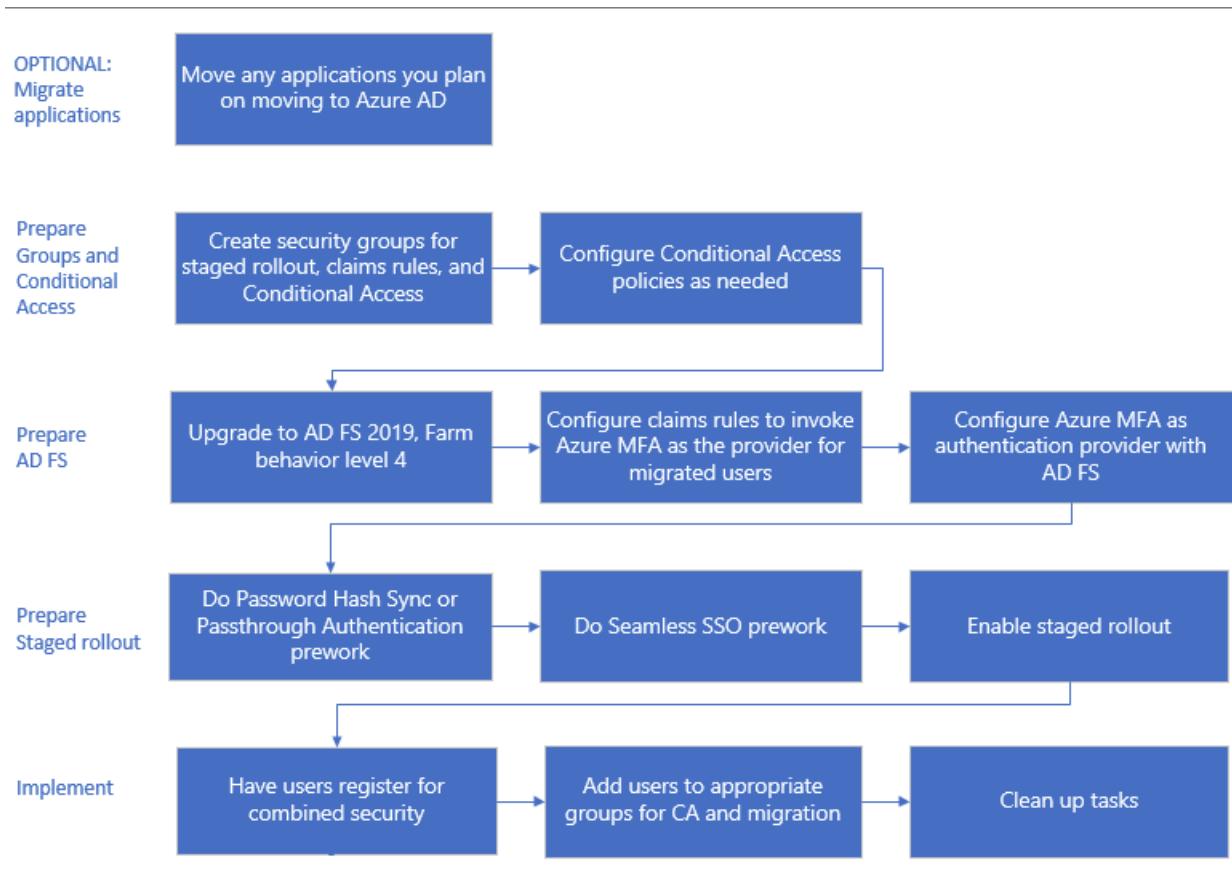
The following diagram shows the process for migrating to Microsoft Entra multifactor authentication and cloud authentication while keeping some of your applications on AD FS. This process enables the iterative migration of users from MFA Server to Microsoft Entra multifactor authentication based on group membership.

Each step is explained in the subsequent sections of this article.

ⓘ Note

If you're planning on moving any applications to Microsoft Entra ID as a part of this migration, you should do so prior to your MFA migration. If you move all of your apps, you can skip sections of the MFA migration process. See the section on moving applications at the end of this article.

Process to migrate to Microsoft Entra ID and user authentication



Prepare groups and Conditional Access

Groups are used in three capacities for MFA migration.

- To iteratively move users to Microsoft Entra multifactor authentication with Staged Rollout.

Use a group created in Microsoft Entra ID, also known as a cloud-only group. You can use Microsoft Entra security groups or Microsoft 365 Groups for both moving users to MFA and for Conditional Access policies.

Important

Nested and dynamic membership groups aren't supported for Staged Rollout. Don't use these types of groups.

- **Conditional Access policies.** You can use either Microsoft Entra ID or on-premises groups for Conditional Access.
- **To invoke Microsoft Entra multifactor authentication for AD FS applications with claims rules.** This step applies only if you use applications with AD FS.

You must use an on-premises Active Directory security group. Once Microsoft Entra multifactor authentication is an additional authentication method, you can designate groups of users to use that method on each relying party trust. For example, you can call Microsoft Entra multifactor authentication for users you already migrated, and MFA Server for users who aren't migrated yet. This strategy is helpful both in testing and during migration.

 **Note**

We don't recommend that you reuse groups that are used for security. Only use the security group to secure a group of high-value apps with a Conditional Access policy.

Configure Conditional Access policies

If you're already using Conditional Access to determine when users are prompted for MFA, you won't need any changes to your policies. As users are migrated to cloud authentication, they'll start using Microsoft Entra multifactor authentication as defined by your Conditional Access policies. They won't be redirected to AD FS and MFA Server anymore.

If your federated domains have the **federatedIdpMfaBehavior** set to `enforceMfaByFederatedIdp` or **SupportsMfa** flag set to `$True` (the **federatedIdpMfaBehavior** overrides **SupportsMfa** when both are set), you're likely enforcing MFA on AD FS by using claims rules. In this case, you'll need to analyze your claims rules on the Microsoft Entra ID relying party trust and create Conditional Access policies that support the same security goals.

If necessary, configure Conditional Access policies before you enable Staged Rollout. For more information, see the following resources:

- [Plan a Conditional Access deployment](#)
- [Common Conditional Access policies](#)

Prepare AD FS

If you don't have any applications in AD FS that require MFA, you can skip this section and go to the section [Prepare Staged Rollout](#).

Upgrade AD FS server farm to 2019, FBL 4

In AD FS 2019, Microsoft released new functionality to help specify additional authentication methods for a relying party, such as an application. You can specify an additional authentication method by using group membership to determine the authentication provider. By specifying an additional authentication method, you can transition to Microsoft Entra multifactor authentication while keeping other authentication intact during the transition.

For more information, see [Upgrading to AD FS in Windows Server 2016 using a WID database](#). The article covers both upgrading your farm to AD FS 2019 and upgrading your FBL to 4.

Configure claims rules to invoke Microsoft Entra multifactor authentication

Now that Microsoft Entra multifactor authentication is an additional authentication method, you can assign groups of users to use Microsoft Entra multifactor authentication by configuring claims rules, also known as *relying party trusts*. By using groups, you can control which authentication provider is called either globally or by application. For example, you can call Microsoft Entra multifactor authentication for users who registered for combined security information or had their phone numbers migrated, while calling MFA Server for users whose phone numbers haven't migrated.

Note

Claims rules require on-premises security group.

Back up rules

Before configuring new claims rules, back up your rules. You'll need to restore claims rules as a part of your clean-up steps.

Depending on your configuration, you may also need to copy the existing rule and append the new rules being created for the migration.

To view global rules, run:

PowerShell

```
Get-AdfsAdditionalAuthenticationRule
```

To view relying party trusts, run the following command and replace RPTrustName with the name of the relying party trust claims rule:

```
PowerShell
```

```
(Get-AdfsRelyingPartyTrust -Name  
"RPTrustName").AdditionalAuthenticationRules
```

Access control policies

ⓘ Note

Access control policies can't be configured so that a specific authentication provider is invoked based on group membership.

To transition from your access control policies to additional authentication rules, run this command for each of your Relying Party Trusts using the MFA Server authentication provider:

```
PowerShell
```

```
Set-AdfsRelyingPartyTrust -**TargetName AppA -AccessControlPolicyName  
$Null**
```

This command will move the logic from your current Access Control Policy into Additional Authentication Rules.

Set up the group, and find the SID

You'll need to have a specific group in which you place users for whom you want to invoke Microsoft Entra multifactor authentication. You'll need to find the security identifier (SID) for that group. To find the group SID, run the following command and replace `GroupName` with your group name:

```
PowerShell
```

```
Get-ADGroup GroupName
```

```
PS C:\> Get-ADGroup -Identity MFAserverUsers

DistinguishedName : CN=MFAserverUsers,OU=Sync,DC=fed,DC=canello,DC=net
GroupCategory    : Security
GroupScope       : Universal
Name             : MFAserverUsers
ObjectClass      : group
ObjectGUID       : 85d97e1b-c0d3-4244-98d2-8920b9c91669
SamAccountName   : MFAserverUsers
SID              : S-1-5-21-814711581-1087136893-663710155-56604
```

Setting the claims rules to call Microsoft Entra multifactor authentication

The following Microsoft Graph PowerShell cmdlets invoke Microsoft Entra multifactor authentication for users in the group when they aren't on the corporate network.

Replace `"YourGroupSid"` with the SID found by running the preceding cmdlet.

Make sure you review the [How to Choose Additional Auth Providers in 2019](#).

ⓘ Important

Back up your claims rules before proceeding.

Set global claims rule

Run the following command and replace RPTrustName with the name of the relying party trust claims rule:

PowerShell

```
(Get-AdfsRelyingPartyTrust -Name  
"RPTrustName").AdditionalAuthenticationRules
```

The command returns your current additional authentication rules for your relying party trust.

You need to append the following rules to your current claim rules:

Console

```
c:[Type ==  
"https://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value  
==
```

```
"YourGroupSID"] => issue(Type =
"https://schemas.microsoft.com/claims/authnmethodsproviders",
Value = "AzureMfaAuthentication");
not exists([Type ==
"https://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
Value=="YourGroupSid"]) => issue(Type =
"https://schemas.microsoft.com/claims/authnmethodsproviders", Value =
"AzureMfaServerAuthentication");'
```

The following example assumes your current claim rules are configured to prompt for MFA when users connect from outside your network. This example includes the additional rules that you need to append.

PowerShell

```
Set-AdfsAdditionalAuthenticationRule -AdditionalAuthenticationRules 'c:[type ==
"https://schemas.microsoft.com/ws/2012/01/insidecorporatenetwork", value ==
>false"] => issue(type =
"https://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmeth
od", value =
"https://schemas.microsoft.com/claims/multipleauthn" );
c:[Type ==
"https://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value ==
==
"YourGroupSID"] => issue(Type =
"https://schemas.microsoft.com/claims/authnmethodsproviders",
Value = "AzureMfaAuthentication");
not exists([Type ==
"https://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
Value=="YourGroupSid"]) => issue(Type =
"https://schemas.microsoft.com/claims/authnmethodsproviders", Value =
"AzureMfaServerAuthentication");'
```

Set per-application claims rule

This example modifies claim rules on a specific relying party trust (application). It includes the additional rules you need to append.

PowerShell

```
Set-AdfsRelyingPartyTrust -TargetName AppA -AdditionalAuthenticationRules
'c:[type ==
"https://schemas.microsoft.com/ws/2012/01/insidecorporatenetwork", value ==
>false"] => issue(type =
"https://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmeth
od", value =
"https://schemas.microsoft.com/claims/multipleauthn" );
c:[Type ==
```

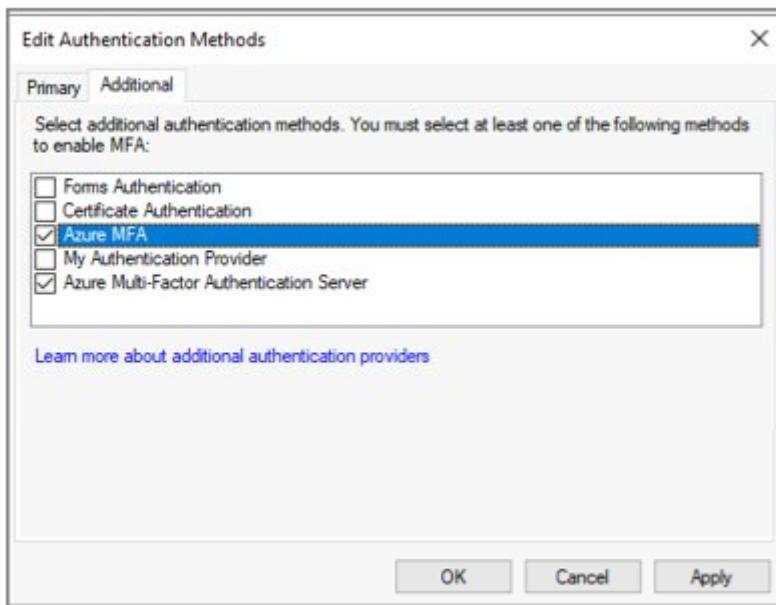
```
"https://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value  
==  
"YourGroupSID"] => issue(Type =  
"https://schemas.microsoft.com/claims/authnmethodsproviders",  
Value = "AzureMfaAuthentication");  
not exists([Type ==  
"https://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",  
Value=="YourGroupSid"]) => issue(Type =  
"https://schemas.microsoft.com/claims/authnmethodsproviders", Value =  
"AzureMfaServerAuthentication");'
```

Configure Microsoft Entra multifactor authentication as an authentication provider in AD FS

In order to configure Microsoft Entra multifactor authentication for AD FS, you must configure each AD FS server. If multiple AD FS servers are in your farm, you can configure them remotely using Microsoft Graph PowerShell.

For step-by-step directions on this process, see [Configure the AD FS servers](#).

After you configure the servers, you can add Microsoft Entra multifactor authentication as an additional authentication method.



Prepare Staged Rollout

Now you're ready to enable [Staged Rollout](#). Staged Rollout helps you to iteratively move your users to either PHS or PTA while also migrating their on-premises MFA settings.

- Be sure to review the [supported scenarios](#).
- First, you'll need to do either the [prework for PHS](#) or the [prework for PTA](#). We recommend PHS.
- Next, you'll do the [prework for seamless SSO](#).
- [Enable the Staged Rollout of cloud authentication](#) for your selected authentication method.
- Add the group(s) you created for Staged Rollout. Remember that you'll add users to groups iteratively, and that they can't be nested or dynamic membership groups.

Register users for Microsoft Entra multifactor authentication

This section covers how users can register for combined security (MFA and self-service-password reset) and how to migrate their MFA settings. Microsoft Authenticator can be used as in passwordless mode. It can also be used as a second factor for MFA with either registration method.

Register for combined security registration (recommended)

We recommend having your users register for combined security information, which is a single place to register their authentication methods and devices for both MFA and SSPR.

Microsoft provides communication templates that you can provide to your users to guide them through the combined registration process. These include templates for email, posters, table tents, and various other assets. Users register their information at <https://aka.ms/mysecurityinfo>, which takes them to the combined security registration screen.

We recommend that you [secure the security registration process with Conditional Access](#) that requires the registration to occur from a trusted device or location. For information on tracking registration statuses, see [Authentication method activity for Microsoft Entra ID](#).

Note

Users who MUST register their combined security information from a non-trusted location or device can be issued a Temporary Access Pass or alternatively,

temporarily excluded from the policy.

Migrate MFA settings from MFA Server

You can use the [MFA Server Migration utility](#) to synchronize registered MFA settings for users from MFA Server to Microsoft Entra ID. You can synchronize phone numbers, hardware tokens, and device registrations such as Microsoft Authenticator app settings.

Add users to the appropriate groups

- If you created new Conditional Access policies, add the appropriate users to those groups.
- If you created on-premises security groups for claims rules, add the appropriate users to those groups.
- Only after you add users to the appropriate Conditional Access rules, add users to the group that you created for Staged Rollout. Once done, they'll begin to use the Azure authentication method that you selected (PHS or PTA) and Microsoft Entra multifactor authentication when they're required to perform MFA.

i Important

Nested and dynamic membership groups aren't supported for Staged Rollout. Do not use these types of groups.

We don't recommend that you reuse groups that are used for security. If you're using a security group to secure a group of high-value apps with a Conditional Access policy, only use the group for that purpose.

Monitoring

Many [Azure Monitor workbooks](#) and [Usage & insights](#) reports are available to monitor your deployment. These reports can be found in Microsoft Entra ID in the navigation pane under **Monitoring**.

Monitoring Staged Rollout

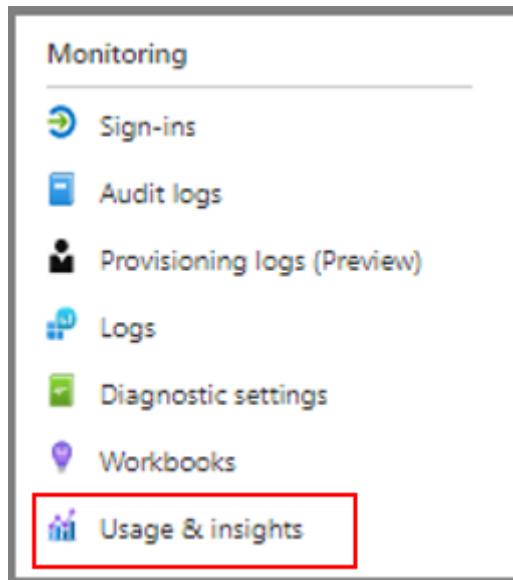
In the [workbooks](#) section, select **Public Templates**. Under **Hybrid Auth** section select the **Groups, Users and Sign-ins in Staged Rollout** workbook.

This workbook can be used to monitor the following activities:

- Users and groups added to Staged Rollout.
- Users and groups removed from Staged Rollout.
- Sign-in failures for users in Staged Rollout, and the reasons for failures.

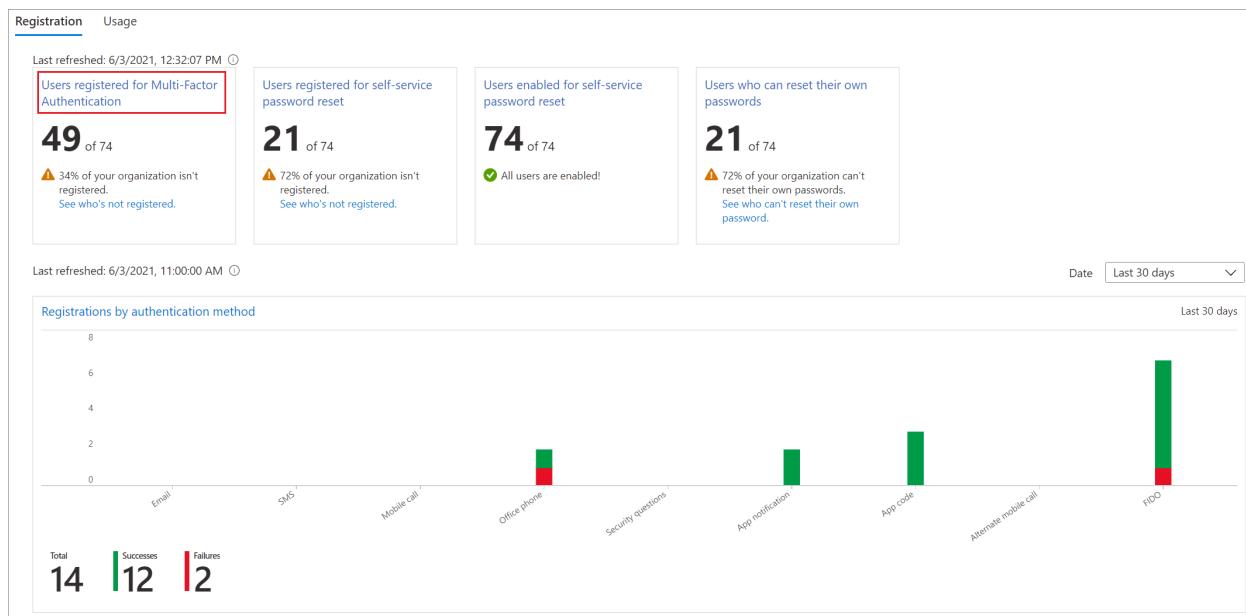
Monitoring Microsoft Entra multifactor authentication registration

Microsoft Entra multifactor authentication registration can be monitored using the [Authentication methods usage & insights report](#). This report can be found in Microsoft Entra ID. Select **Monitoring**, then select **Usage & insights**.



In Usage & insights, select **Authentication methods**.

Detailed Microsoft Entra multifactor authentication registration information can be found on the Registration tab. You can drill down to view a list of registered users by selecting the **Users registered for Azure multifactor authentication** hyperlink.



Monitoring app sign-in health

Monitor applications you moved to Microsoft Entra ID with the App sign-in health workbook or the application activity usage report.

- **App sign-in health workbook.** See [Monitoring application sign-in health for resilience](#) for detailed guidance on using this workbook.
- **Microsoft Entra application activity usage report.** This [report](#) can be used to view the successful and failed sign-ins for individual applications as well as the ability to drill down and view sign-in activity for a specific application.

Clean up tasks

After you move all users to Microsoft Entra cloud authentication and Microsoft Entra multifactor authentication, you're ready to decommission your MFA Server. We recommend reviewing MFA Server logs to ensure no users or applications are using it before you remove the server.

Convert your domains to managed authentication

You should now [convert your federated domains in Microsoft Entra ID to managed](#) and remove the Staged Rollout configuration. This conversion ensures new users use cloud authentication without being added to the migration groups.

Revert claims rules on AD FS and remove MFA Server authentication provider

Follow the steps under [Configure claims rules to invoke Microsoft Entra multifactor authentication](#) to revert the claims rules and remove any AzureMFA Server Authentication claims rules.

For example, remove the following section from the rule(s):

```
Console

c:[Type ==
"https://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value
==

"**YourGroupSID**"] => issue(Type =
"https://schemas.microsoft.com/claims/authnmethodsproviders",
Value = "AzureMfaAuthentication");
not exists([Type ==
"https://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
Value=="YourGroupSid"]) => issue(Type =
"https://schemas.microsoft.com/claims/authnmethodsproviders", Value =
"AzureMfaServerAuthentication");'
```

Disable MFA Server as an authentication provider in AD FS

This change ensures only Microsoft Entra multifactor authentication is used as an authentication provider.

1. Open the **AD FS management console**.
2. Under **Services**, right-click on **Authentication Methods**, and select **Edit multifactor authentication Methods**.
3. Clear the **Azure Multi-Factor Authentication Server** checkbox.

Decommission the MFA Server

Follow your enterprise server decommissioning process to remove the MFA Servers in your environment.

Possible considerations when decommissions the MFA Server include:

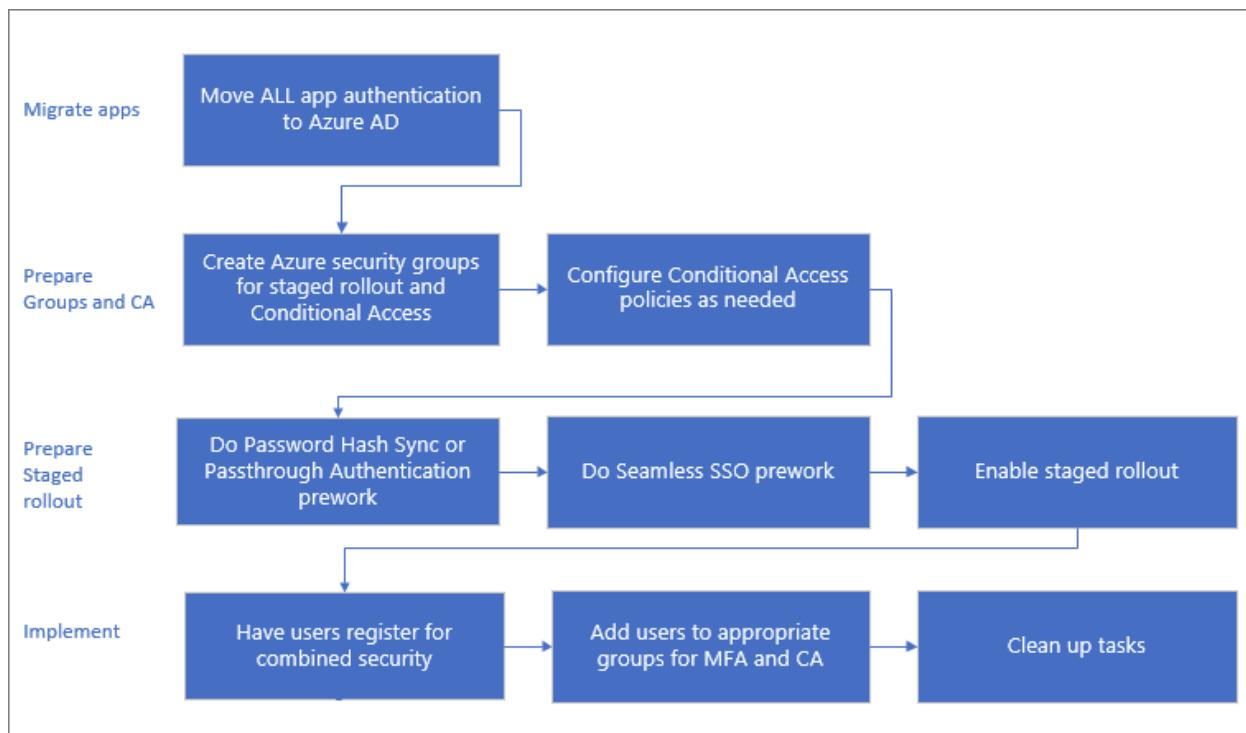
- We recommend reviewing MFA Server logs to ensure no users or applications are using it before you remove the server.
- Uninstall Multi-Factor Authentication Server from the Control Panel on the server.
- Optionally clean up logs and data directories that are left behind after backing them up first.

- Uninstall the multifactor authentication Web Server SDK, if applicable including any files left over inetpub\wwwroot\MultiFactorAuthWebServiceSdk and/or MultiFactorAuth directories.
- For pre-8.0.x versions of MFA Server, it may also be necessary to remove the multifactor authentication Phone App Web Service.

Move application authentication to Microsoft Entra ID

If you migrate all your application authentication with your MFA and user authentication, you'll be able to remove significant portions of your on-premises infrastructure, reducing costs and risks. If you move all application authentication, you can skip the [Prepare AD FS](#) stage and simplify your MFA migration.

The process for moving all application authentication is shown in the following diagram.



If you can't move all your applications before the migration, move as many as possible before you start. For more information about migrating applications to Azure, see [Resources for migrating applications to Microsoft Entra ID](#).

Next steps

- Migrate from Microsoft MFA Server to Microsoft Entra multifactor authentication (Overview)
- Migrate applications from Windows Active Directory to Microsoft Entra ID

- Plan your cloud authentication strategy
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

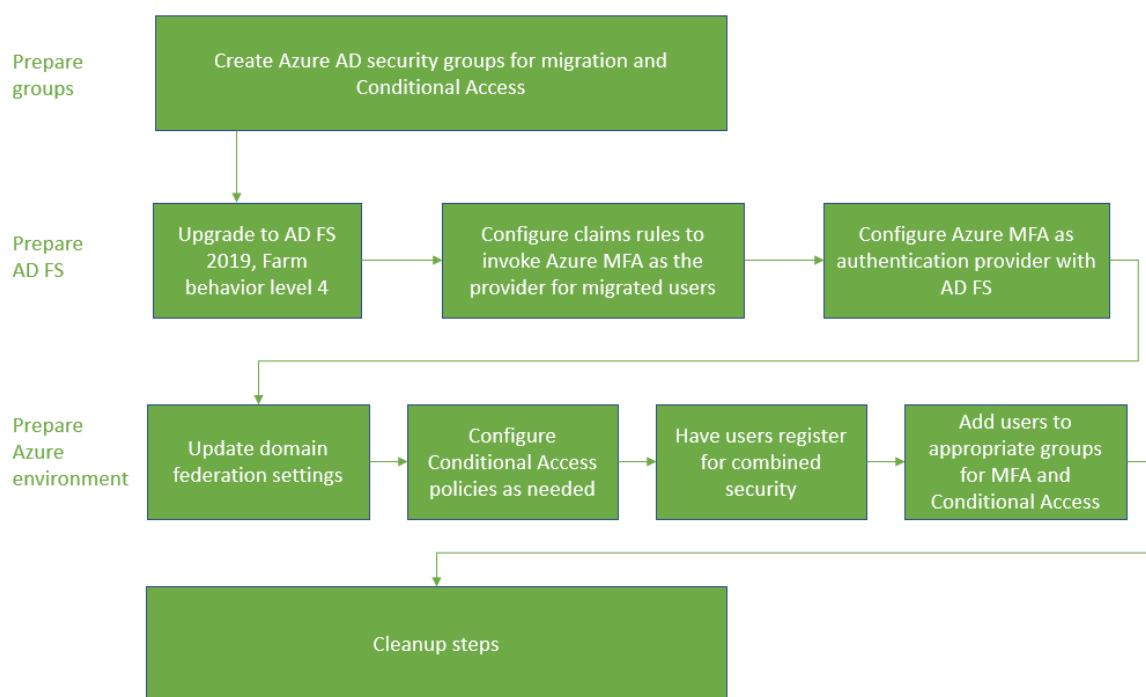
Migrate to Microsoft Entra multifactor authentication with federation

Article • 03/04/2025

Moving your multifactor-authentication (MFA) solution to Microsoft Entra ID is a great first step in your journey to the cloud. Consider also moving to Microsoft Entra ID for user authentication in the future. For more information, see the process for migrating to Microsoft Entra multifactor authentication with cloud authentication.

To migrate to Microsoft Entra multifactor authentication with federation, the Microsoft Entra multifactor authentication authentication provider is installed on AD FS. The Microsoft Entra ID relying party trust and other relying party trusts are configured to use Microsoft Entra multifactor authentication for migrated users.

The following diagram shows the migration process.



Create migration groups

To create new Conditional Access policies, you'll need to assign those policies to groups. You can use Microsoft Entra security groups or Microsoft 365 Groups for this purpose. You can also create or sync new ones.

You'll also need a Microsoft Entra security group for iteratively migrating users to Microsoft Entra multifactor authentication. These groups are used in your claims rules.

Don't reuse groups that are used for security. If you're using a security group to secure a group of high-value apps with a Conditional Access policy, only use the group for that purpose.

Prepare AD FS

Upgrade AD FS server farm to 2019, FBL 4

In AD FS 2019, you can specify additional authentication methods for a relying party, such as an application. You use group membership to determine authentication provider. By specifying an additional authentication method, you can transition to Microsoft Entra multifactor authentication while keeping other authentication intact during the transition. For more information, see [Upgrading to AD FS in Windows Server 2016 using a WID database](#). The article covers both upgrading your farm to AD FS 2019 and upgrading your FBL to 4.

Configure claims rules to invoke Microsoft Entra multifactor authentication

Now that Microsoft Entra multifactor authentication is an additional authentication method, you can assign groups of users to use it. You do so by configuring claims rules, also known as relying party trusts. By using groups, you can control which authentication provider is called globally or by application. For example, you can call Microsoft Entra multifactor authentication for users who have registered for combined security information, while calling MFA Server for those who haven't.

ⓘ Note

Claims rules require on-premises security group. Before making changes to claims rules, back them up.

Back up rules

Before configuring new claims rules, back up your rules. You'll need to restore these rules as a part of your clean-up steps.

Depending on your configuration, you may also need to copy the rule and append the new rules being created for the migration.

To view global rules, run:

```
PowerShell
```

```
Get-AdfsAdditionalAuthenticationRule
```

To view relying party trusts, run the following command and replace RPTrustName with the name of the relying party trust claims rule:

```
PowerShell
```

```
(Get-AdfsRelyingPartyTrust -Name  
"RPTrustName").AdditionalAuthenticationRules
```

Access control policies

① Note

Access control policies can't be configured so that a specific authentication provider is invoked based on group membership.

To transition from access control policies to additional authentication rules, run the following command for each of your Relying Party Trusts using the MFA Server authentication provider:

```
PowerShell
```

```
Set-AdfsRelyingPartyTrust -TargetName AppA -AccessControlPolicyName $Null
```

This command will move the logic from your current Access Control Policy into Additional Authentication Rules.

Set up the group, and find the SID

You'll need to have a specific group in which you place users for whom you want to invoke Microsoft Entra multifactor authentication. You'll need the security identifier (SID) for that group.

To find the group SID, use the following command, with your group name

```
Get-ADGroup "GroupName"
```

```
PS C:\> Get-ADGroup -Identity MFAServerUsers

DistinguishedName : CN=MFAServerUsers,OU=Sync,DC=fed,DC=canello,DC=net
GroupCategory     : Security
GroupScope        : Universal
Name              : MFAServerUsers
ObjectClass       : group
ObjectGUID        : 85d97e1b-c0d3-4244-98d2-8920b9c91669
SamAccountName   : MFAServerUsers
SID               : S-1-5-21-814711581-1087136893-663710155-56604
```

Setting the claims rules to call Microsoft Entra multifactor authentication

The following PowerShell cmdlets invoke Microsoft Entra multifactor authentication for users in the group when not on the corporate network. Replace "YourGroupSid" with the SID found by running the above cmdlet.

Make sure you review the [How to Choose Additional Auth Providers in 2019](#).

ⓘ Important

Back up your claims rules

Set global claims rule

Run the following PowerShell cmdlet:

```
PowerShell

(Get-AdfsRelyingPartyTrust -Name
"RPTrustName").AdditionalAuthenticationRules
```

The command returns your current additional authentication rules for your relying party trust. Append the following rules to your current claim rules:

```
Console

c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value ==
"YourGroupSID"] => issue(Type =
```

```
"http://schemas.microsoft.com/claims/authnmethodsproviders",
Value = "AzureMfaAuthentication");
not exists([Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
Value=="YourGroupSid"]) => issue(Type =
"http://schemas.microsoft.com/claims/authnmethodsproviders", Value =
"AzureMfaServerAuthentication");'
```

The following example assumes your current claim rules are configured to prompt for MFA when users connect from outside your network. This example includes the additional rules that you need to append.

PowerShell

```
Set-AdfsAdditionalAuthenticationRule -AdditionalAuthenticationRules 'c:[type ==
"http://schemas.microsoft.com/ws/2012/01/insidecorporatenetwork", value ==
>false"] => issue(type =
"http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmetho
d", value =
"http://schemas.microsoft.com/claims/multipleauthn" );
c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value ==
>YourGroupSID"] => issue(Type =
"http://schemas.microsoft.com/claims/authnmethodsproviders",
Value = "AzureMfaAuthentication");
not exists([Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
Value=="YourGroupSid"]) => issue(Type =
"http://schemas.microsoft.com/claims/authnmethodsproviders", Value =
"AzureMfaServerAuthentication");'
```

Set per-application claims rule

This example modifies claim rules on a specific relying party trust (application), and includes the information you must append.

PowerShell

```
Set-AdfsRelyingPartyTrust -TargetName AppA -AdditionalAuthenticationRules
'c:[type ==
"http://schemas.microsoft.com/ws/2012/01/insidecorporatenetwork", value ==
>false"] => issue(type =
"http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmetho
d", value =
"http://schemas.microsoft.com/claims/multipleauthn" );
c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value ==
>YourGroupSID"] => issue(Type =
```

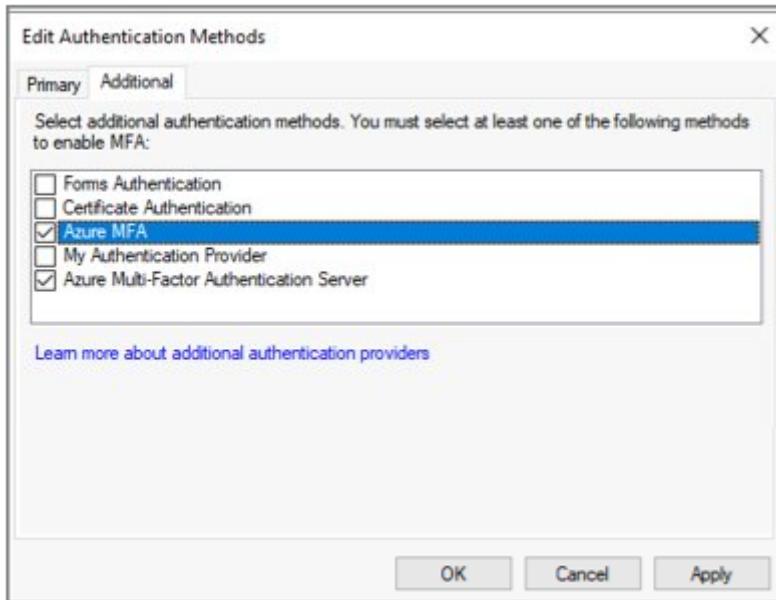
```
"http://schemas.microsoft.com/claims/authnmethodsproviders",
Value = "AzureMfaAuthentication");
not exists([Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
Value=="YourGroupSid"]) => issue(Type =
"http://schemas.microsoft.com/claims/authnmethodsproviders", Value =
"AzureMfaServerAuthentication");'
```

Configure Microsoft Entra multifactor authentication as an authentication provider in AD FS

To configure Microsoft Entra multifactor authentication for AD FS, you must configure each AD FS server. If you have multiple AD FS servers in your farm, you can configure them remotely using Microsoft Entra PowerShell.

For step-by-step directions on this process, see [Configure the AD FS servers](#) in the article [Configure Microsoft Entra multifactor authentication as authentication provider with AD FS](#).

Once you've configured the servers, you can add Microsoft Entra multifactor authentication as an additional authentication method.



Prepare Microsoft Entra ID and implement migration

This section covers final steps before migrating user MFA settings.

Set federatedIdpMfaBehavior to enforceMfaByFederatedIdp

For federated domains, MFA may be enforced by Microsoft Entra Conditional Access or by the on-premises federation provider. Each federated domain has a Microsoft Graph PowerShell security setting named **federatedIdpMfaBehavior**. You can set **federatedIdpMfaBehavior** to `enforceMfaByFederatedIdp` so Microsoft Entra ID accepts MFA that's performed by the federated identity provider. If the federated identity provider didn't perform MFA, Microsoft Entra ID redirects the request to the federated identity provider to perform MFA. For more information, see [federatedIdpMfaBehavior](#).

ⓘ Note

The **federatedIdpMfaBehavior** setting is a new version of the **SupportsMfa** property of the [New-MgDomainFederationConfiguration](#) cmdlet.

For domains that set the **SupportsMfa** property, these rules determine how **federatedIdpMfaBehavior** and **SupportsMfa** work together:

- Switching between **federatedIdpMfaBehavior** and **SupportsMfa** isn't supported.
- Once **federatedIdpMfaBehavior** property is set, Microsoft Entra ID ignores the **SupportsMfa** setting.
- If the **federatedIdpMfaBehavior** property is never set, Microsoft Entra ID will continue to honor the **SupportsMfa** setting.
- If **federatedIdpMfaBehavior** or **SupportsMfa** isn't set, Microsoft Entra ID will default to `acceptIfMfaDoneByFederatedIdp` behavior.

You can check the status of **federatedIdpMfaBehavior** by using [Get-MgDomainFederationConfiguration](#).

PowerShell

```
Get-MgDomainFederationConfiguration -DomainID yourdomain.com
```

You can also check the status of your **SupportsMfa** flag with [Get-MgDomainFederationConfiguration](#):

PowerShell

```
Get-MgDomainFederationConfiguration -DomainName yourdomain.com
```

The following example shows how to set `federatedIdpMfaBehavior` to `enforceMfaByFederatedIdp` by using Graph PowerShell.

Request

HTTP

PATCH

```
https://graph.microsoft.com/beta/domains/contoso.com/federationConfiguration  
/6601d14b-d113-8f64-fda2-9b5ddda18ecc  
Content-Type: application/json  
{  
    "federatedIdpMfaBehavior": "enforceMfaByFederatedIdp"  
}
```

Response

Note: The response object shown here might be shortened for readability.

HTTP

```
HTTP/1.1 200 OK  
Content-Type: application/json  
{  
    "@odata.type": "#microsoft.graph.internalDomainFederation",  
    "id": "6601d14b-d113-8f64-fda2-9b5ddda18ecc",  
    "issuerUri": "http://contoso.com/adfs/services/trust",  
    "metadataExchangeUri": "https://sts.contoso.com/adfs/services/trust/mex",  
    "signingCertificate": "A1bC2dE3fH4iJ5kL6mN7oP8qR9sT0u",  
    "passiveSignInUri": "https://sts.contoso.com/adfs/ls",  
    "preferredAuthenticationProtocol": "wsFed",  
    "activeSignInUri":  
        "https://sts.contoso.com/adfs/services/trust/2005/usernamemixed",  
        "signOutUri": "https://sts.contoso.com/adfs/ls",  
        "promptLoginBehavior": "nativeSupport",  
        "isSignedAuthenticationRequestRequired": true,  
        "nextSigningCertificate": "A1bC2dE3fH4iJ5kL6mN7oP8qR9sT0u",  
        "signingCertificateUpdateStatus": {  
            "certificateUpdateResult": "Success",  
            "lastRunDateTime": "2021-08-25T07:44:46.2616778Z"  
        },  
    "federatedIdpMfaBehavior": "enforceMfaByFederatedIdp"  
}
```

Configure Conditional Access policies if needed

If you use Conditional Access to determine when users are prompted for MFA, you shouldn't need to change your policies.

If your federated domain(s) have `SupportsMfa` set to false, analyze your claims rules on the Microsoft Entra ID relying party trust and create Conditional Access policies that support the same security goals.

After creating Conditional Access policies to enforce the same controls as AD FS, you can back up and remove your claim rules customizations on the Microsoft Entra ID Relying Party.

For more information, see the following resources:

- [Plan a Conditional Access deployment](#)
- [Common Conditional Access policies](#)

Register users for Microsoft Entra multifactor authentication

This section covers how users can register for combined security (MFA and self-service-password reset) and how to migrate their MFA settings. Microsoft Authenticator can be used as in passwordless mode. It can also be used as a second factor for MFA with either registration method.

Register for combined security registration (recommended)

We recommend having your users register for combined security information, which is a single place to register their authentication methods and devices for both MFA and SSPR.

Microsoft provides communication templates that you can provide to your users to guide them through the combined registration process. These include templates for email, posters, table tents, and various other assets. Users register their information at <https://aka.ms/mysecurityinfo>, which takes them to the combined security registration screen.

We recommend that you [secure the security registration process with Conditional Access](#) that requires the registration to occur from a trusted device or location. For information on tracking registration statuses, see [Authentication method activity for Microsoft Entra ID](#).

Note

Users who must register their combined security information from a non-trusted location or device can be issued a Temporary Access Pass or alternatively, temporarily excluded from the policy.

Migrate MFA settings from MFA Server

You can use the [MFA Server Migration utility](#) to synchronize registered MFA settings for users from MFA Server to Microsoft Entra ID. You can synchronize phone numbers, hardware tokens, and device registrations such as Microsoft Authenticator settings.

Add users to the appropriate groups

- If you created new Conditional Access policies, add the appropriate users to those groups.
- If you created on-premises security groups for claims rules, add the appropriate users to those groups.

We don't recommend that you reuse groups that are used for security. If you're using a security group to secure a group of high-value apps with a Conditional Access policy, only use the group for that purpose.

Monitoring

Microsoft Entra multifactor authentication registration can be monitored using the [Authentication methods usage & insights report](#). This report can be found in Microsoft Entra ID. Select **Monitoring**, then select **Usage & insights**.

In Usage & insights, select **Authentication methods**.

Detailed Microsoft Entra multifactor authentication registration information can be found on the Registration tab. You can drill down to view a list of registered users by selecting the **Users capable of Azure multifactor authentication** hyperlink.

The screenshot shows the 'Authentication methods | Activity' section of the Microsoft Entra ID portal. It displays three main categories of users:

- Users capable of Azure multifactor authentication:** 430 of 1169 total. A note indicates 60% of the organization isn't capable.
- Users capable of passwordless authentication:** 42 of 1169 total. A note indicates 99% of the organization isn't capable.
- Users capable of self-service password reset:** 384 of 1169 total. A note indicates 67% of the organization isn't enabled.

Cleanup steps

Once you have completed migration to Microsoft Entra multifactor authentication and are ready to decommission the MFA Server, do the following three things:

1. Revert your claim rules on AD FS to their pre-migration configuration and remove the MFA Server authentication provider.
2. Remove MFA server as an authentication provider in AD FS. This will ensure all users use Microsoft Entra multifactor authentication as it will be the only additional authentication method enabled.
3. Decommission the MFA Server.

Revert claims rules on AD FS and remove MFA Server authentication provider

Follow the steps under Configure claims rules to invoke Microsoft Entra multifactor authentication to revert back to the backed up claims rules and remove any AzureMFAServerAuthentication claims rules.

For example, remove the following from the rule(s):

```
Console

c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value ==
"**YourGroupSID**"] => issue(Type =
"http://schemas.microsoft.com/claims/authnmethodsproviders",
Value = "AzureMfaAuthentication");
not exists([Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
Value=="YourGroupSid"]) => issue(Type =
"http://schemas.microsoft.com/claims/authnmethodsproviders", Value =
"AzureMfaServerAuthentication");'
```

Disable MFA Server as an authentication provider in AD FS

This change ensures only Microsoft Entra multifactor authentication is used as an authentication provider.

1. Open the **AD FS management console**.
2. Under **Services**, right-click on **Authentication Methods**, and select **Edit multifactor authentication Methods**.
3. Uncheck the box next to **Azure Multi-Factor Authentication Server**.

Decommission the MFA Server

Follow your enterprise server decommissioning process to remove the MFA Servers in your environment.

Possible considerations when decommissions the MFA Servers include:

- Review MFA Servers' logs to ensure no users or applications are using it before you remove the server.
- Uninstall Multi-Factor Authentication Server from the Control Panel on the server
- Optionally clean up logs and data directories that are left behind after backing them up first.
- Uninstall the multifactor authentication Web Server SDK if applicable, including any files left over in etpub\wwwroot\MultiFactorAuthWebServiceSdk and or MultiFactorAuth directories
- For MFA Server versions prior to 8.0, it may also be necessary to remove the multifactor authentication Phone App Web Service

Next Steps

- [Deploy password hash synchronization](#)
- [Learn more about Conditional Access](#)
- [Migrate applications to Microsoft Entra ID](#)

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback 

MFA Server migration

Article • 03/04/2025

This topic covers how to migrate MFA settings for Microsoft Entra users from on-premises Microsoft Entra multifactor authentication Server to Microsoft Entra multifactor authentication.

Solution overview

The MFA Server Migration Utility helps synchronize multifactor authentication data stored in the on-premises Microsoft Entra multifactor authentication Server directly to Microsoft Entra multifactor authentication. After the authentication data is migrated to Microsoft Entra ID, users can perform cloud-based MFA seamlessly without having to register again or confirm authentication methods. Admins can use the MFA Server Migration Utility to target single users or groups of users for testing and controlled rollout without having to make any tenant-wide changes.

Video: How to use the MFA Server Migration Utility

Take a look at our video for an overview of the MFA Server Migration Utility and how it works.

<https://learn-video.azurefd.net/vod/player?id=76b2e4d2-514f-4f8a-8675-2931c013c672&locale=en-us&embedUrl=%2Fentra%2Fidentity%2Fauthentication%2Fhow-to-mfa-server-migration-utility>

Limitations and requirements

- The MFA Server Migration Utility requires a new build of the MFA Server solution to be installed on your Primary MFA Server. The build makes updates to the MFA Server data file, and includes the new MFA Server Migration Utility. You don't have to update the WebSDK or User portal. Installing the update *doesn't* start the migration automatically.

 Note

The MFA Server Migration Utility can be executed on a secondary MFA Server. For more information, please check [Run a secondary MFA Server \(optional\)](#).

- The MFA Server Migration Utility copies the data from the database file onto the user objects in Microsoft Entra ID. During migration, users can be targeted for Microsoft Entra multifactor authentication for testing purposes using [Staged Rollout](#). Staged migration lets you test without making any changes to your domain federation settings. Once migrations are complete, you must finalize your migration by making changes to your domain federation settings.
- AD FS running Windows Server 2016 or higher is required to provide MFA authentication on any AD FS relying parties, not including Microsoft Entra ID and Office 365.
- Review your AD FS access control policies and make sure none requires MFA to be performed on-premises as part of the authentication process.
- Staged rollout can target a maximum of 500,000 users (10 groups containing a maximum of 50,000 users each).

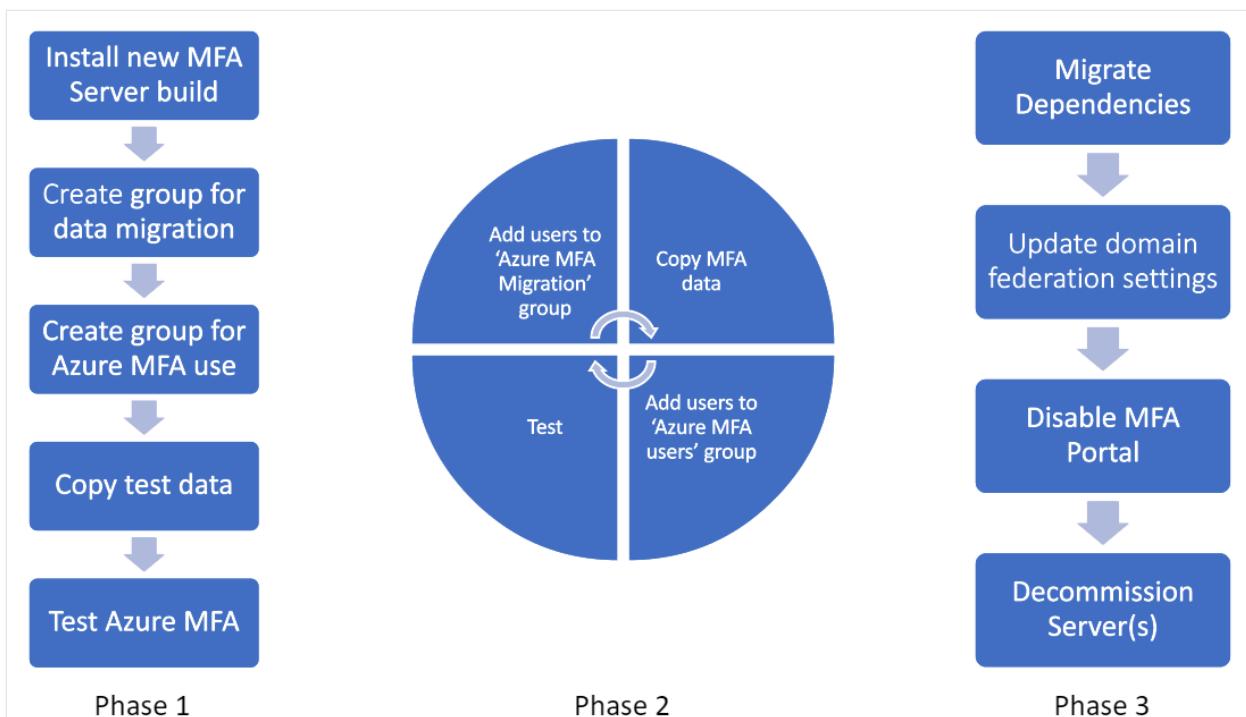
Migration guide

[] [Expand table](#)

Phase	Steps
Preparations	Identify Microsoft Entra multifactor authentication Server dependencies Backup Microsoft Entra multifactor authentication Server datafile Install MFA Server update Configure MFA Server Migration Utility
Migrations	Migrate user data Validate and test Staged Rollout Educate users Complete user migration
Finalize	Migrate MFA Server dependencies

Phase	Steps
	Update domain federation settings
	Disable MFA Server User portal
	Decommission MFA server

An MFA Server migration generally includes the steps in the following process:



A few important points:

Phase 1 should be repeated as you add test users.

- The migration tool uses Microsoft Entra groups for determining the users for which authentication data should be synced between MFA Server and Microsoft Entra multifactor authentication. After user data is synced, that user is then ready to use Microsoft Entra multifactor authentication.
- Staged Rollout allows you to reroute users to Microsoft Entra multifactor authentication, also using Microsoft Entra groups. While you certainly could use the same groups for both tools, we recommend against it as users could potentially be redirected to Microsoft Entra multifactor authentication before the tool has synched their data. We recommend setting up Microsoft Entra groups for syncing authentication data by the MFA Server Migration Utility, and another set of groups for Staged Rollout to direct targeted users to Microsoft Entra multifactor authentication rather than on-premises.

Phase 2 should be repeated as you migrate your user base. By the end of Phase 2, your entire user base should be using Microsoft Entra multifactor authentication for all

workloads federated against Microsoft Entra ID.

During the previous phases, you can remove users from the Staged Rollout folders to take them out of scope of Microsoft Entra multifactor authentication and route them back to your on-premises Microsoft Entra multifactor authentication server for all MFA requests originating from Microsoft Entra ID.

Phase 3 requires moving all clients that authenticate to the on-premises MFA Server (VPNs, password managers, and so on) to Microsoft Entra federation via SAML/OAUTH. If modern authentication standards aren't supported, you're required to stand up NPS server(s) with the Microsoft Entra multifactor authentication extension installed. Once dependencies are migrated, users should no longer use the User portal on the MFA Server, but rather should manage their authentication methods in Microsoft Entra ID ([aka.ms/mfasetup](#) ↗). Once users begin managing their authentication data in Microsoft Entra ID, those methods aren't synced back to MFA Server. If you roll back to the on-premises MFA Server after users have made changes to their Authentication Methods in Microsoft Entra ID, those changes are lost. After user migrations are complete, change the `federatedIdpMfaBehavior` domain federation setting. The change tells Microsoft Entra ID to no longer perform MFA on-premises and to perform *all* MFA requests with Microsoft Entra multifactor authentication, regardless of group membership.

The following sections explain the migration steps in more detail.

Identify Microsoft Entra multifactor authentication Server dependencies

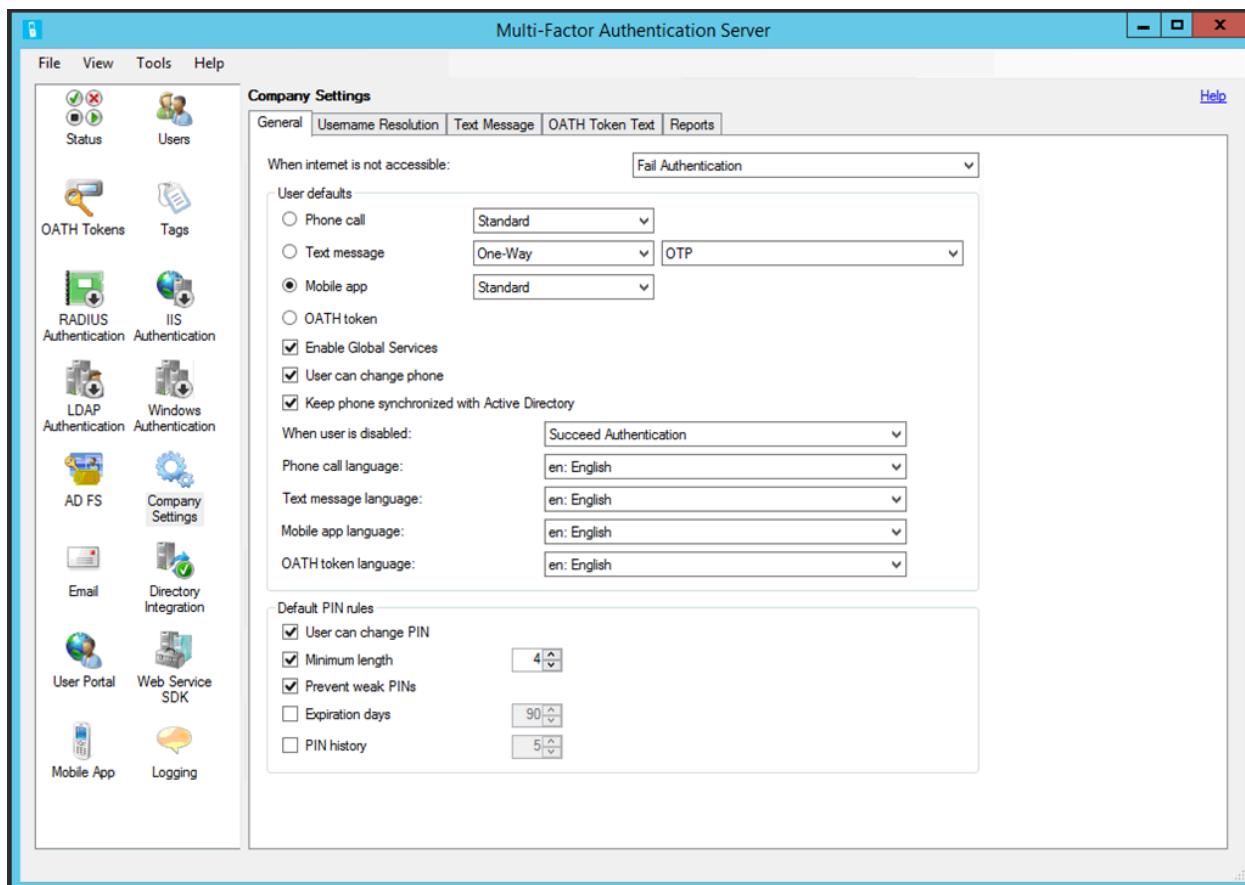
We've worked hard to ensure that moving onto our cloud-based Microsoft Entra multifactor authentication solution maintains and improves your security posture. There are three broad categories that should be used to group dependencies:

- [MFA methods](#)
- [User portal](#)
- [Authentication services](#)

To help your migration, we've matched widely used MFA Server features with the functional equivalent in Microsoft Entra multifactor authentication for each category.

MFA methods

Open MFA Server, select **Company Settings**:



[Expand table](#)

MFA Server	Microsoft Entra multifactor authentication
General Tab	
User Defaults section	
Phone call (Standard)	No action needed
Text message (OTP)*	No action needed
Mobile app (Standard)	No action needed
Phone Call (PIN)*	Enable Voice OTP
Text message (OTP + PIN)**	No action needed
Mobile app (PIN)*	Enable number matching
Phone call/text message/mobile app/OATH token language	Language settings are automatically applied to a user based on the locale settings in their browser
Default PIN rules section	Not applicable; see updated methods in the preceding screenshot
Username Resolution tab	Not applicable; username resolution isn't required for Microsoft Entra multifactor authentication

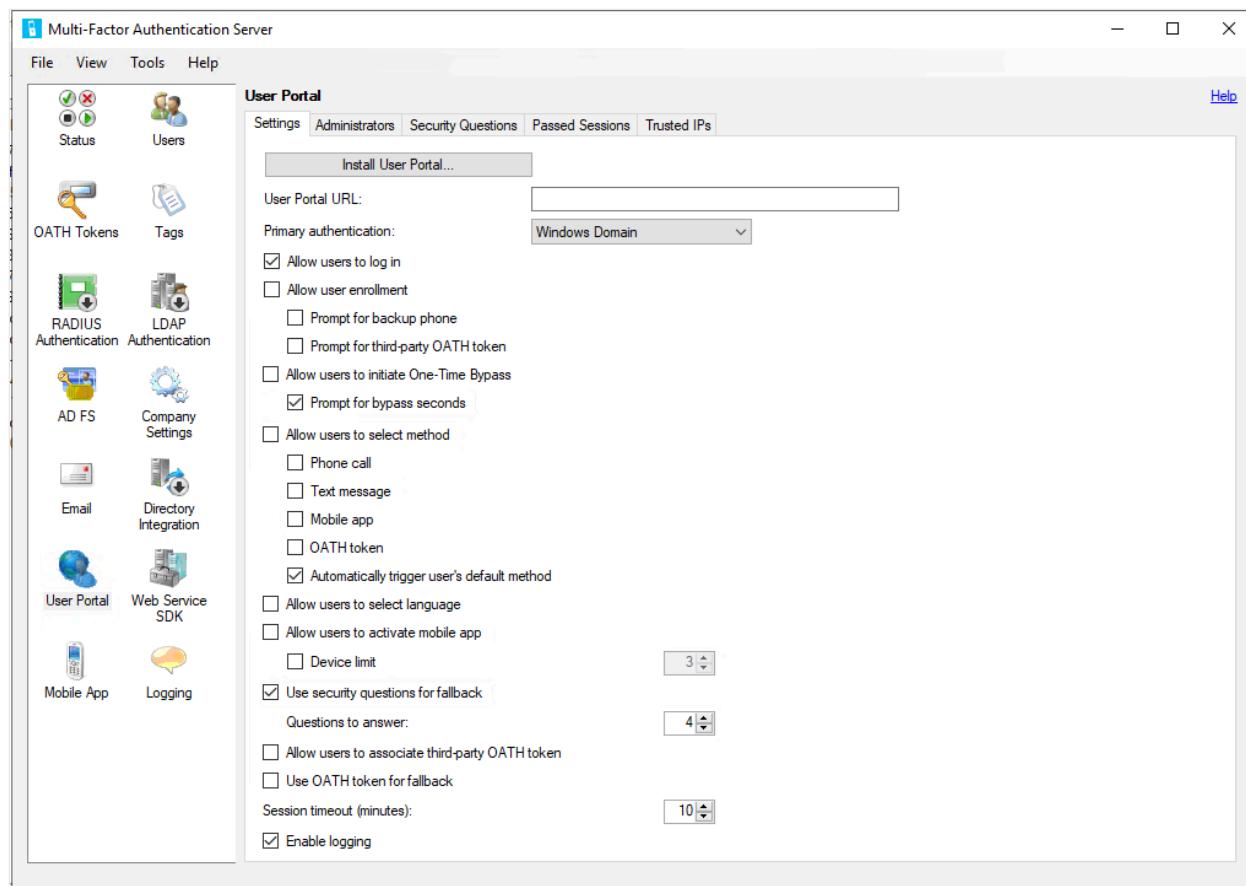
MFA Server	Microsoft Entra multifactor authentication
Text Message tab	Not applicable; Microsoft Entra multifactor authentication uses a default message for text messages
OATH Token tab	Not applicable; Microsoft Entra multifactor authentication uses a default message for OATH tokens
Reports	Microsoft Entra authentication Methods Activity reports

*When a PIN is used to provide proof-of-presence functionality, the functional equivalent is provided above. PINs that aren't cryptographically tied to a device don't sufficiently protect against scenarios where a device has been compromised. To protect against these scenarios, including [SIM swap attacks](#), move users to more secure methods according to Microsoft authentication methods [best practices](#).

**The default Text MFA experience in Microsoft Entra multifactor authentication sends users a code, which they're required to enter in the login window as part of authentication. The requirement to roundtrip the code provides proof-of-presence functionality.

User portal

Open MFA Server, select User Portal:



MFA Server	Microsoft Entra multifactor authentication
Settings Tab	
User portal URL	aka.ms/mfasetup ↗
Allow user enrollment	See Combined security information registration
- Prompt for backup phone	See MFA Service settings
- Prompt for third-party OATH token	See MFA Service settings
Allow users to initiate a One-Time Bypass	See Microsoft Entra ID TAP functionality
Allow users to select method	See MFA Service settings
- Phone call	See Phone call documentation
- Text message	See MFA Service settings
- Mobile app	See MFA Service settings
- OATH token	See OATH token documentation
Allow users to select language	Language settings are automatically applied to a user based on the locale settings in their browser
Allow users to activate mobile app	See MFA Service settings
- Device limit	Microsoft Entra ID limits users to five cumulative devices (mobile app instances + hardware OATH token + software OATH token) per user
Use security questions for fallback	Microsoft Entra ID allows users to choose a fallback method at authentication time should the chosen authentication method fail
- Questions to answer	Security Questions in Microsoft Entra ID can only be used for SSPR. See more details for Microsoft Entra Custom Security Questions
Allow users to associate third-party OATH token	See OATH token documentation

MFA Server	Microsoft Entra multifactor authentication
Use OATH token for fallback	See OATH token documentation
Session Timeout	
Security Questions tab	Security questions in MFA Server were used to gain access to the User portal. Microsoft Entra multifactor authentication only supports security questions for self-service password reset. See security questions documentation .
Passed Sessions tab	All authentication method registration flows are managed by Microsoft Entra ID and don't require configuration
Trusted IPs	Microsoft Entra ID trusted IPs

Any MFA methods available in MFA Server must be enabled in Microsoft Entra multifactor authentication by using [MFA Service settings](#). Users can't try their newly migrated MFA methods unless they're enabled.

Authentication services

Microsoft Entra multifactor authentication Server can provide MFA functionality for third-party solutions that use RADIUS or LDAP by acting as an authentication proxy. To discover RADIUS or LDAP dependencies, select **RADIUS Authentication** and **LDAP Authentication** options in MFA Server. For each of these dependencies, determine if these third parties support modern authentication. If so, consider federation directly with Microsoft Entra ID.

For RADIUS deployments that can't be upgraded, you'll need to deploy an NPS Server and install the [Microsoft Entra multifactor authentication NPS extension](#).

For LDAP deployments that can't be upgraded or moved to RADIUS, [determine if Microsoft Entra Domain Services can be used](#). In most cases, LDAP was deployed to support in-line password changes for end users. Once migrated, end users can manage their passwords by using [self-service password reset in Microsoft Entra ID](#).

If you enabled the [MFA Server Authentication provider in AD FS 2.0](#) on any relying party trusts except for the Office 365 relying party trust, you'll need to upgrade to [AD FS 3.0](#) or federate those relying parties directly to Microsoft Entra ID if they support modern authentication methods. Determine the best plan of action for each of the dependencies.

Backup Microsoft Entra multifactor authentication Server datafile

Make a backup of the MFA Server data file located at %programfiles%\Multi-Factor Authentication Server\Data\PhoneFactor.pfdata (default location) on your primary MFA Server. Make sure you have a copy of the installer for your currently installed version in case you need to roll back. If you no longer have a copy, contact Customer Support Services.

Depending on user activity, the data file can become outdated quickly. Any changes made to MFA Server, or any end-user changes made through the portal after the backup won't be captured. If you roll back, any changes made after this point won't be restored.

Install MFA Server update

Run the new installer on the Primary MFA Server. Before you upgrade a server, remove it from load balancing or traffic sharing with other MFA Servers. You don't need to uninstall your current MFA Server before running the installer. The installer performs an in-place upgrade using the current installation path (for example, C:\Program Files\Multi-Factor Authentication Server). If you're prompted to install a Microsoft Visual C++ 2015 Redistributable update package, accept the prompt. Both the x86 and x64 versions of the package are installed. It isn't required to install updates for User portal, Web SDK, or AD FS Adapter.

Note

After you run the installer on your primary server, secondary servers may begin to log **Unhandled SB** entries. This is due to schema changes made on the primary server that aren't recognized by secondary servers. These errors are expected. In environments with 10,000 users or more, the amount of log entries can increase significantly. To mitigate this issue, you can increase the file size of your MFA Server logs, or upgrade your secondary servers.

Configure the MFA Server Migration Utility

After installing the MFA Server update, open an elevated PowerShell command prompt: hover over the PowerShell icon, right-select, and select **Run as Administrator**. Run the .\Configure-MultiFactorAuthMigrationUtility.ps1 script found in your MFA Server installation directory (C:\Program Files\Multi-Factor Authentication Server by default).

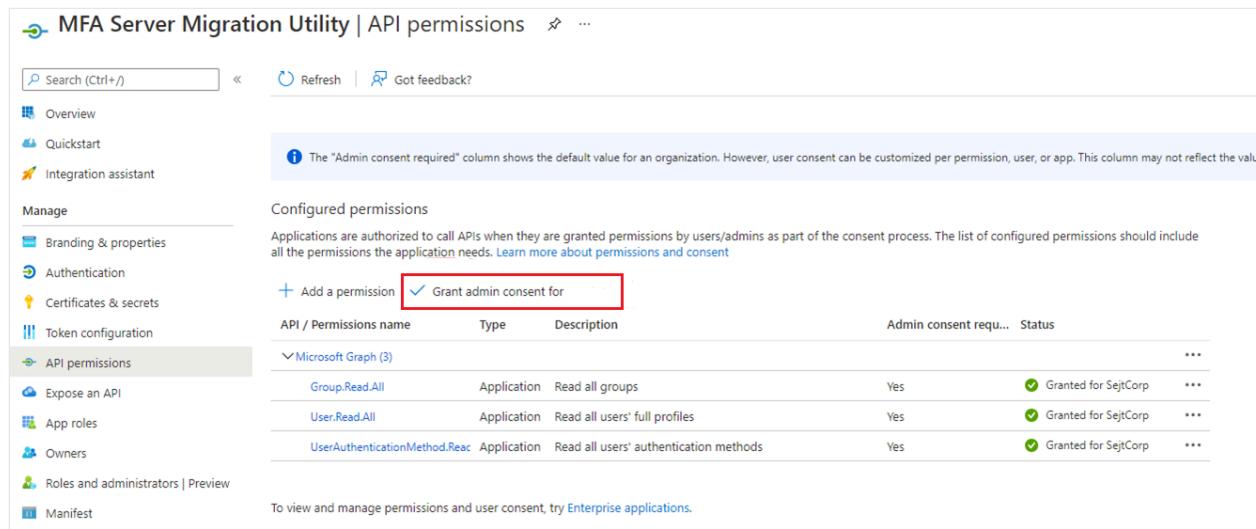
This script requires you to provide credentials for an Application Administrator in your Microsoft Entra tenant. It creates a new MFA Server Migration Utility application within Microsoft Entra ID, which is used to write user authentication methods to each Microsoft Entra user object.

For government cloud customers who wish to carry out migrations, replace ".com" entries in the script with ".us". This script writes the HKLM:\SOFTWARE\WOW6432Node\Positive Networks\PhoneFactor\ StsUrl and GraphUrl registry entries and instructs the Migration Utility to use the appropriate GRAPH endpoints.

You'll also need access to the following URLs:

- https://graph.microsoft.com/* (or https://graph.microsoft.us/* for government cloud customers)
- https://login.microsoftonline.com/* (or https://login.microsoftonline.us/* for government cloud customers)

The script instructs you to grant admin consent to the newly created application. Navigate to the URL provided, or within the Microsoft Entra admin center, select **Application Registrations**, find and select the **MFA Server Migration Utility** app, select on **API permissions** and then grant the appropriate permissions.

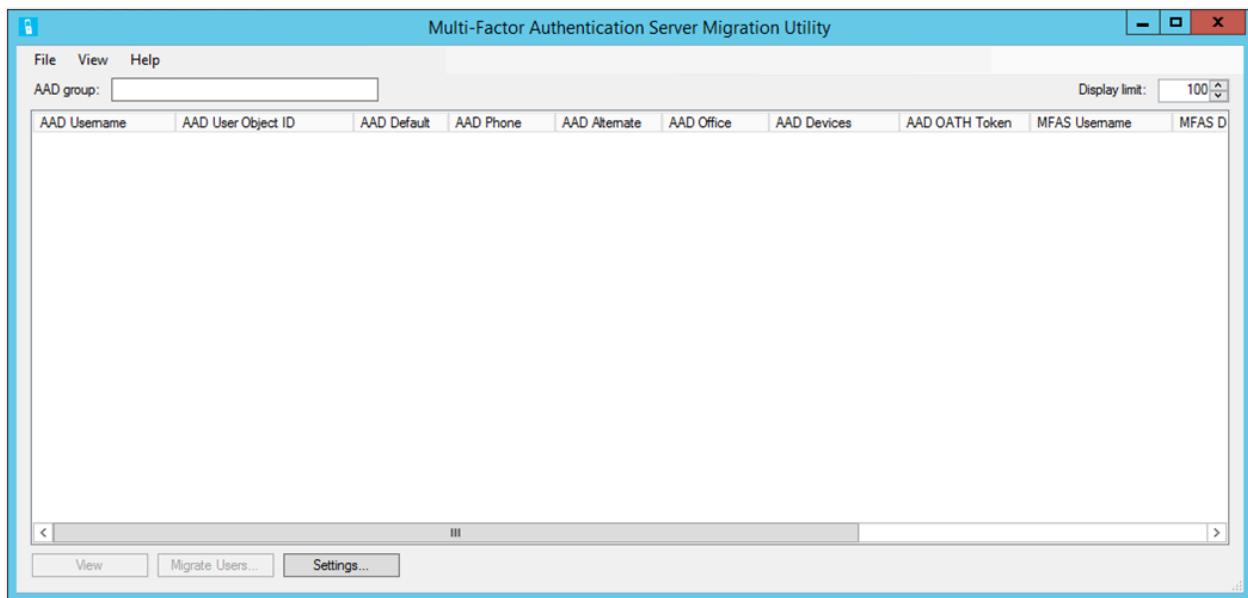


The screenshot shows the Microsoft Entra admin center interface. The left sidebar has a 'Manage' section with various options like Branding & properties, Authentication, Certificates & secrets, Token configuration, API permissions (which is selected), Expose an API, App roles, Owners, Roles and administrators | Preview, and Manifest. The main content area is titled 'MFA Server Migration Utility | API permissions'. It has a search bar, refresh button, and feedback link. A note says 'The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value.' Below this is a 'Configured permissions' section with a 'Grant admin consent for' button, which is highlighted with a red box. A table lists API permissions under 'Microsoft Graph (3)':

API / Permissions name	Type	Description	Admin consent requ...	Status
Group.Read.All	Application	Read all groups	Yes	Granted for SejtCorp
User.Read.All	Application	Read all users' full profiles	Yes	Granted for SejtCorp
UserAuthenticationMethod.Read	Application	Read all users' authentication methods	Yes	Granted for SejtCorp

At the bottom, it says 'To view and manage permissions and user consent, try Enterprise applications.'

Once complete, navigate to the Multi-Factor Authentication Server folder, and open the **MultiFactorAuthMigrationUtilityUI** application. You should see the following screen:



You've successfully installed the Migration Utility.

① Note

To ensure no changes in behavior during migration, if your MFA Server is associated with an MFA Provider with no tenant reference, you'll need to update the default MFA settings (such as custom greetings) for the tenant you're migrating to match the settings in your MFA Provider. We recommend doing this before migrating any users.

Run a secondary MFA Server (optional)

If your MFA Server implementation has a large number of users or a busy primary MFA Server, you may want to consider deploying a dedicated secondary MFA Server for running the MFA Server Migration Utility and Migration Sync services. After upgrading your primary MFA Server, either upgrade an existing secondary server or deploy a new secondary server. The secondary server you choose shouldn't be handling other MFA traffic.

The `Configure-MultiFactorAuthMigrationUtility.ps1` script should be run on the secondary server to register a certificate with the MFA Server Migration Utility app registration. The certificate is used to authenticate to Microsoft Graph. Running the Migration Utility and Sync services on a secondary MFA Server should improve performance of both manual and automated user migrations.

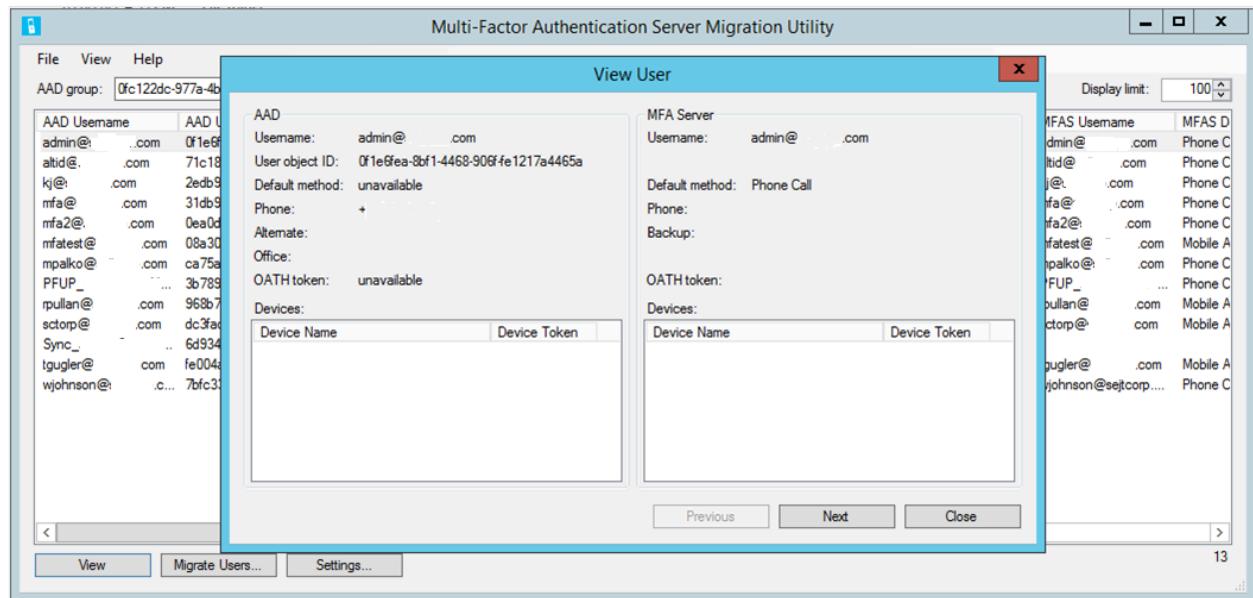
Migrate user data

Migrating user data doesn't remove or alter any data in the Multi-Factor Authentication Server database. Likewise, this process won't change where a user performs MFA. This process is a one-way copy of data from the on-premises server to the corresponding user object in Microsoft Entra ID.

The MFA Server Migration utility targets a single Microsoft Entra group for all migration activities. You can add users directly to this group, or add other groups. You can also add them in stages during the migration.

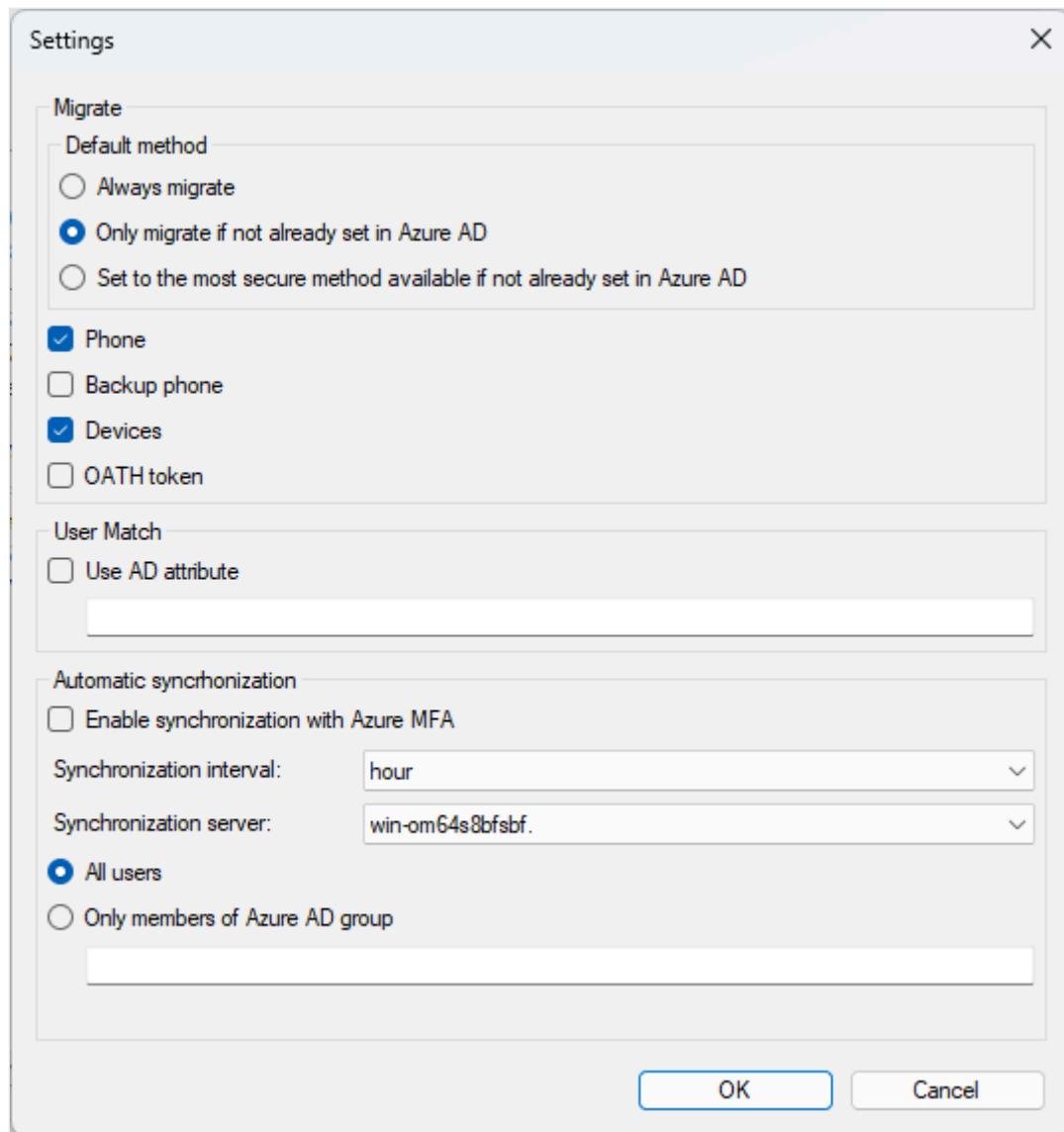
To begin the migration process, enter the name or GUID of the Microsoft Entra group you want to migrate. Once complete, press Tab or select outside the window to begin searching for the appropriate group. All users in the group are populated. A large group can take several minutes to finish.

To view attribute data for a user, highlight the user, and select **View**:



This window displays the attributes for the selected user in both Microsoft Entra ID and the on-premises MFA Server. You can use this window to view how data was written to a user after migration.

The **Settings** option allows you to change the settings for the migration process:



- Migrate – there are three options for migrating the user's default authentication method:
 - Always migrate
 - Only migrate if not already set in Microsoft Entra ID
 - Set to the most secure method available if not already set in Microsoft Entra ID

These options provide flexibility when you migrate the default method. In addition, the Authentication methods policy is checked during migration. If the default method being migrated isn't allowed by policy, it's set to the most secure method available instead.

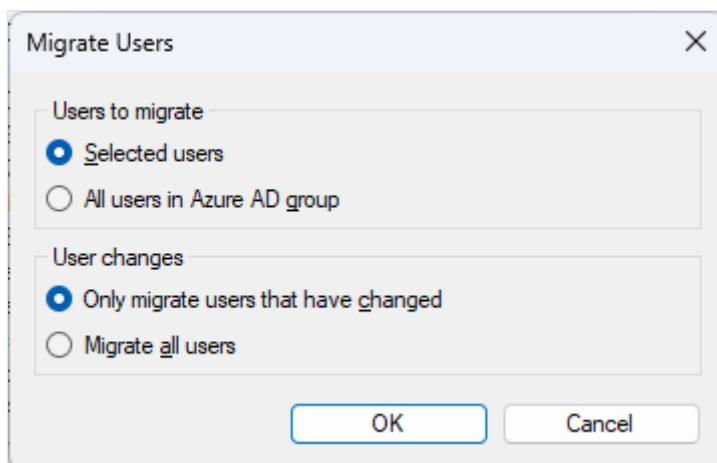
- User Match – Allows you to specify a different on-premises Active Directory attribute for matching Microsoft Entra UPN instead of the default match to `userPrincipalName`:
 - The migration utility tries direct matching to UPN before using the on-premises Active Directory attribute.
 - If no match is found, it calls a Windows API to find the Microsoft Entra UPN and get the SID, which it uses to search the MFA Server user list.

- If the Windows API doesn't find the user or the SID isn't found in the MFA Server, then it uses the configured Active Directory attribute to find the user in the on-premises Active Directory, and then use the SID to search the MFA Server user list.
- Automatic synchronization – Starts a background service that continually monitors any authentication method changes to users in the on-premises MFA Server, and writes them to Microsoft Entra ID at the specified time interval defined.
- Synchronization server – Allows the MFA Server Migration Sync service to run on a secondary MFA Server rather than only run on the primary. To configure the Migration Sync service to run on a secondary server, the `Configure-MultiFactorAuthMigrationUtility.ps1` script must be run on the server to register a certificate with the MFA Server Migration Utility app registration. The certificate is used to authenticate to Microsoft Graph.

The migration process can be automatic or manual.

The manual process steps are:

1. To begin the migration process for a user or selection of multiple users, press and hold the Ctrl key while selecting each of the user(s) you wish to migrate.
2. After you select the desired users, select **Migrate Users > Selected users > OK**.
3. To migrate all users in the group, select **Migrate Users > All users in Microsoft Entra group > OK**.
4. You can migrate users even if they're unchanged. By default, the utility is set to **Only migrate users that have changed**. Select **Migrate all users** to re-migrate previously migrated users that are unchanged. Migrating unchanged users can be useful during testing if an administrator needs to reset a user's Microsoft Entra multifactor authentication settings and wants to re-migrate them.



For the automatic process, select **Automatic synchronization** in **Settings**, and then select whether you want all users to be synced, or only members of a given Microsoft Entra group.

The following table lists the sync logic for the various methods.

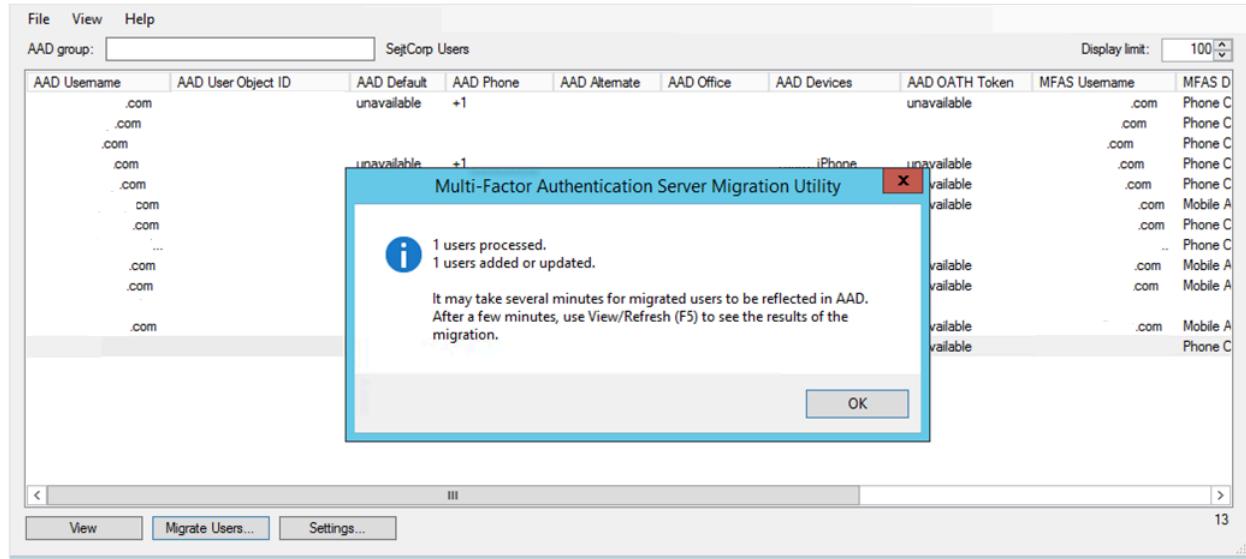
[] Expand table

Method	Logic
Phone	If there's no extension, update MFA phone. If there's an extension, update Office phone. Exception: If the default method is Text Message, drop extension and update MFA phone.
Backup Phone	If there's no extension, update Alternate phone. If there's an extension, update Office phone. Exception: If both Phone and Backup Phone have an extension, skip Backup Phone.
Mobile App	Maximum of five devices are migrated or only four if the user also has a hardware OATH token. If there are multiple devices with the same name, only migrate the most recent one. Devices are ordered from newest to oldest. If devices already exist in Microsoft Entra ID, match on OATH Token Secret Key and update. <ul style="list-style-type: none">- If there's no match on OATH Token Secret Key, match on Device Token-- If found, create a Software OATH Token for the MFA Server device to allow OATH Token method to work. Notifications still work using the existing Microsoft Entra multifactor authentication device.-- If not found, create a new device. If adding a new device exceeds the five-device limit, the device is skipped.
OATH Token	If devices already exist in Microsoft Entra ID, match on OATH Token Secret Key and update. <ul style="list-style-type: none">- If not found, add a new Hardware OATH Token device. If adding a new device exceeds the five-device limit, the OATH token is skipped.

MFA Methods are updated based on what was migrated and the default method is set. MFA Server tracks the last migration timestamp and only migrate the user again if the user's MFA settings change or an admin modifies what to migrate in the **Settings** dialog.

During testing, we recommend doing a manual migration first, and test to ensure a given number of users behave as expected. Once testing is successful, turn on automatic synchronization for the Microsoft Entra group you wish to migrate. As you add users to this group, their information is automatically synchronized to Microsoft Entra ID. MFA Server Migration Utility targets one Microsoft Entra group, however that group can encompass both users and nested groups of users.

Once complete, a confirmation informs you of the tasks completed:



As mentioned in the confirmation message, it can take several minutes for the migrated data to appear on user objects within Microsoft Entra ID. Users can view their migrated methods by navigating to aka.ms/mfasetup.

💡 Tip

You can reduce the time required to display groups if you don't need to view Microsoft Entra MFA methods. Select **View > Azure AD MFA Methods** to toggle the display of columns for **AAD Default**, **AAD Phone**, **AAD Alternate**, **AAD Office**, **AAD Devices**, and **AAD OATH Token**. When columns are hidden, some Microsoft Graph API calls are skipped, which greatly improves user load time.

View migration details

You can use Audit logs or Log Analytics to view details of MFA Server to Microsoft Entra multifactor authentication user migrations.

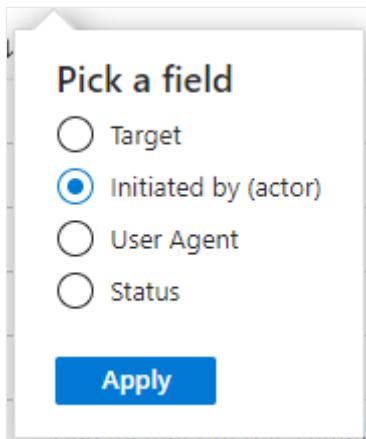
Use Audit logs

To access the Audit logs in the Microsoft Entra admin center to view details of MFA Server to Microsoft Entra multifactor authentication user migrations, follow these steps:

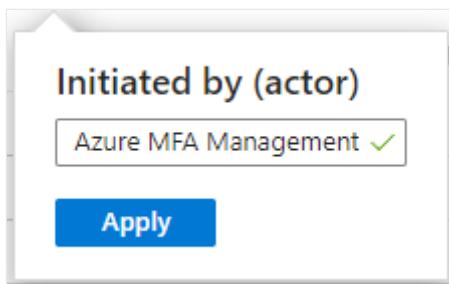
1. Sign in to the [Microsoft Entra admin center](#) as at least an [Authentication Administrator](#).
2. Browse to **Identity > Monitoring & health > Audit logs**. To filter the logs, select **Add filters**.

 Add filters

3. Select **Initiated by (actor)** and select **Apply**.



4. Type *Microsoft Entra multifactor authentication Management* and select **Apply**.



5. This filter displays only MFA Server Migration Utility logs. To view details for a user migration, select a row, and then choose the **Modified Properties** tab. This tab shows changes to registered MFA methods and phone numbers.

Audit Log Details			
Activity	Target(s)	Modified Properties	X
Target	Property Name	Old Value	New Value
	StrongAuthenticationMethod	[]	[{"MethodType":0,"Default":false}, {"MethodType":5,"Default":false}, {"MethodType":6,"Default":true}, {"MethodType":7,"Default":false}]
	StrongAuthenticationUserDetails	[]	[{"PhoneNumber":"+1384","AlternativePhoneNumber":null, "Email":null,"VoiceOnlyPhoneNumber":null}]
	Included Updated Properties		"StrongAuthenticationMethod, StrongAuthenticationUserDetails"
	TargetId.UserType		"Member"

The following table lists the authentication method for each code.

[Expand table](#)

Code	Method
0	Voice mobile
2	Voice office
3	Voice alternate mobile
5	SMS
6	Microsoft Authenticator push notification
7	Hardware or software token OTP

6. If any user devices were migrated, there's a separate log entry.

Audit Log Details

Activity	Target(s)	Modified Properties		
	Target	Property Name	Old Value	New Value
		StrongAuthenti...	[]	[{"DeviceName": "iPhone 14"}]
		Included Updat...		"StrongAuthenticationPhoneAppDetail"
		TargetId.UserTy...		"Member"

Use Log Analytics

The details of MFA Server to Microsoft Entra multifactor authentication user migrations can also be queried using Log Analytics.

Kusto

```
AuditLogs
| where ActivityDateTime > ago(7d)
| extend InitiatedBy = tostring(InitiatedBy["app"]["displayName"])
| where InitiatedBy == "Microsoft Entra multifactor authentication Management"
| extend UserObjectId = tostring(TargetResources[0]["id"])
| extend Upn = tostring(TargetResources[0]["userPrincipalName"])
```

```

| extend ModifiedProperties = TargetResources[0]["modifiedProperties"]
| project ActivityDateTime, InitiatedBy, UserObjectId, Upn,
ModifiedProperties
| order by ActivityDateTime asc

```

This screenshot shows changes for user migration:

8/9/2023, 5:29:38.233 PM	Azure MFA Management	44ee44ee-ff55-aa66-bb77-88cc88cc88cc-00000000-aaaa-1111-bbbb-222222cccccc	[{"displayName": "StrongAuthenticationMethod", "oldValue": "[]", "newValue": "[{\\"MethodType\\": \"0\", \"Default\\": false}, {\\"MethodType\\": \"5\", \"Default\\": false}, {\\"MethodType\\": \"6\", \"Default\\": true}, {\\"MethodType\\": \"7\", \"Default\\": false}], {\\"MethodType\\": \"1\", \"Default\\": false}, {\\"MethodType\\": \"2\", \"Default\\": false}, {\\"MethodType\\": \"3\", \"Default\\": false}, {\\"MethodType\\": \"4\", \"Default\\": false}]"}
ActivityDateTime [UTC]		2023-08-09T17:29:38.233582Z	
InitiatedBy	Azure MFA Management		
UserObjectId	aaaaaaaa-0000-1111-2222-bbbbbbbbbb		
Upn	A1B2C3D4E5@woodgrove		
ModifiedProperties			
0	{"displayName": "StrongAuthenticationMethod", "oldValue": "[]", "newValue": "[{\\"MethodType\\": \"0\", \"Default\\": false}, {\\"MethodType\\": \"5\", \"Default\\": false}, {\\"MethodType\\": \"6\", \"Default\\": true}, {\\"MethodType\\": \"7\", \"Default\\": false}], {\\"MethodType\\": \"1\", \"Default\\": false}, {\\"MethodType\\": \"2\", \"Default\\": false}, {\\"MethodType\\": \"3\", \"Default\\": false}, {\\"MethodType\\": \"4\", \"Default\\": false}]"}		
displayName	StrongAuthenticationMethod		
newValue	[{"MethodType": "0", "Default": false}, {"MethodType": "5", "Default": false}, {"MethodType": "6", "Default": true}, {"MethodType": "7", "Default": false}, {"MethodType": "1", "Default": false}, {"MethodType": "2", "Default": false}, {"MethodType": "3", "Default": false}, {"MethodType": "4", "Default": false}]		
> 0	{"MethodType": "0", "Default": false}		
> 1	{"MethodType": "5", "Default": false}		
> 2	{"MethodType": "6", "Default": true}		
> 3	{"MethodType": "7", "Default": false}		
oldValue	[]		
1	{"displayName": "StrongAuthenticationUserDetails", "oldValue": "[]", "newValue": "[{\\"PhoneNumber\\": \"567-555-0199\", \\"AlternativePhoneNumber\\": null, \\"Email\\": null, \\"VoiceOnlyPhoneNumber\\": null}]"} displayName StrongAuthenticationUserDetails		
newValue	"PhoneNumber": "567-555-0199", "AlternativePhoneNumber": null, "Email": null, "VoiceOnlyPhoneNumber": null}		
0	{"PhoneNumber": "567-555-0199", "AlternativePhoneNumber": null, "Email": null, "VoiceOnlyPhoneNumber": null}		
AlternativePhoneNumber	null		
Email	null		
PhoneNumber	null		
VoiceOnlyPhoneNumber	null		
oldValue	[]		

This screenshot shows changes for device migration:

8/9/2023, 5:29:38.434 PM	Azure MFA Management	44ee44ee-ff55-aa66-bb77-88cc88cc88cc-00000000-aaaa-1111-bbbb-222222cccccc	[{"displayName": "StrongAuthenticationPhoneAppDetail", "oldValue": "[]", "newValue": "[{\\"DeviceName\\": \"iPhone 14\", \\"DeviceToken\\": \"apns2-abc123def456ghi789abc123def456ghi789abc123def456ghi789abc123def4560\", \\"DeviceTag\\": \"\", \\"SoftwareTokenActivated\\": 0, \\"PhoneAppVersion\\": \"6.7.12\", \\"OathTokenTimeDrift\\": 0, \\"DeviceId\\": \"00000000-0000-0000-0000-000000000000\", \\"DeviceName\\": \"iPhone 14\", \\"DeviceToken\\": \"apns2-abc123def456ghi789abc123def456ghi789abc123def456ghi789abc123def4560\", \\"DeviceTag\\": \"\", \\"SoftwareTokenActivated\\": 0, \\"PhoneAppVersion\\": \"6.7.12\", \\"OathTokenTimeDrift\\": 0, \\"DeviceId\\": \"00000000-0000-0000-0000-000000000000\", \\"AuthenticationType\\": 3, \\"AuthenticatorFlavor\\": \"Authenticator\", \\"DeviceId\\": \"00000000-0000-0000-0000-000000000000\", \\"DeviceName\\": \"iPhone 14\", \\"DeviceTag\\": \"SoftwareTokenActivated\", \\"PhoneAppVersion\\": \"6.7.12\", \\"OathTokenTimeDrift\\": 0, \\"DeviceId\\": \"00000000-0000-0000-0000-000000000000\", \\"HashFunction\\": \"\", \\"Id\\": \"3d3d3d3d-4444-eeee-5555-6f6f6f6f6f\", \\"LastAuthenticatedTimestamp\\": \"2001-01-01T00:00:00\", \\"NotificationType\\": 2, \\"OathTokenTimeDrift\\": 0, \\"PhoneAppVersion\\": \"6.7.12\", \\"SecuredKeyId\\": 0, \\"SecuredPartitionId\\": 0, \\"TenantDeviceId\\": null, \\"TimeInterval\\": 0}]"}
ActivityDateTime [UTC]		2023-08-09T17:29:38.434584Z	
InitiatedBy	Azure MFA Management		
UserObjectId	aaaaaaaa-0000-1111-2222-bbbbbbbbbb		
Upn	A1B2C3D4E5@woodgrove		
ModifiedProperties			
0	{"displayName": "StrongAuthenticationPhoneAppDetail", "oldValue": "[]", "newValue": "[{\\"DeviceName\\": \"iPhone 14\", \\"DeviceToken\\": \"apns2-abc123def456ghi789abc123def456ghi789abc123def456ghi789abc123def4560\", \\"DeviceTag\\": \"\", \\"SoftwareTokenActivated\\": 0, \\"PhoneAppVersion\\": \"6.7.12\", \\"OathTokenTimeDrift\\": 0, \\"DeviceId\\": \"00000000-0000-0000-0000-000000000000\", \\"DeviceName\\": \"iPhone 14\", \\"DeviceToken\\": \"apns2-abc123def456ghi789abc123def456ghi789abc123def456ghi789abc123def4560\", \\"DeviceTag\\": \"\", \\"SoftwareTokenActivated\\": 0, \\"PhoneAppVersion\\": \"6.7.12\", \\"OathTokenTimeDrift\\": 0, \\"DeviceId\\": \"00000000-0000-0000-0000-000000000000\", \\"AuthenticationType\\": 3, \\"AuthenticatorFlavor\\": \"Authenticator\", \\"DeviceId\\": \"00000000-0000-0000-0000-000000000000\", \\"DeviceName\\": \"iPhone 14\", \\"DeviceTag\\": \"SoftwareTokenActivated\", \\"PhoneAppVersion\\": \"6.7.12\", \\"OathTokenTimeDrift\\": 0, \\"DeviceId\\": \"00000000-0000-0000-0000-000000000000\", \\"HashFunction\\": \"\", \\"Id\\": \"3d3d3d3d-4444-eeee-5555-6f6f6f6f6f\", \\"LastAuthenticatedTimestamp\\": \"2001-01-01T00:00:00\", \\"NotificationType\\": 2, \\"OathTokenTimeDrift\\": 0, \\"PhoneAppVersion\\": \"6.7.12\", \\"SecuredKeyId\\": 0, \\"SecuredPartitionId\\": 0, \\"TenantDeviceId\\": null, \\"TimeInterval\\": 0}]"}		
displayName	StrongAuthenticationPhoneAppDetail		
newValue	[{"DeviceName": "iPhone 14", "DeviceToken": "apns2-abc123def456ghi789abc123def456ghi789abc123def456ghi789abc123def4560", "DeviceTag": "SoftwareTokenActivated", "PhoneAppVersion": "6.7.12", "OathTokenTimeDrift": 0, "DeviceId": "00000000-0000-0000-0000-000000000000", "AuthenticationType": 3, "AuthenticatorFlavor": "Authenticator", "DeviceId": "00000000-0000-0000-0000-000000000000", "DeviceName": "iPhone 14", "DeviceTag": "SoftwareTokenActivated", "PhoneAppVersion": "6.7.12", "OathTokenTimeDrift": 0, "DeviceId": "00000000-0000-0000-0000-000000000000", "HashFunction": "", "Id": "3d3d3d3d-4444-eeee-5555-6f6f6f6f6f", "LastAuthenticatedTimestamp": "2001-01-01T00:00:00", "NotificationType": 2, "OathTokenTimeDrift": 0, "PhoneAppVersion": "6.7.12", "SecuredKeyId": 0, "SecuredPartitionId": 0, "TenantDeviceId": null, "TimeInterval": 0}]		
0	{"DeviceName": "iPhone 14", "DeviceToken": "apns2-abc123def456ghi789abc123def456ghi789abc123def456ghi789abc123def4560", "DeviceTag": "SoftwareTokenActivated", "PhoneAppVersion": "6.7.12", "OathTokenTimeDrift": 0, "DeviceId": "00000000-0000-0000-0000-000000000000", "AuthenticationType": 3, "AuthenticatorFlavor": "Authenticator", "DeviceId": "00000000-0000-0000-0000-000000000000", "DeviceName": "iPhone 14", "DeviceTag": "SoftwareTokenActivated", "PhoneAppVersion": "6.7.12", "OathTokenTimeDrift": 0, "DeviceId": "00000000-0000-0000-0000-000000000000", "HashFunction": "", "Id": "3d3d3d3d-4444-eeee-5555-6f6f6f6f6f", "LastAuthenticatedTimestamp": "2001-01-01T00:00:00", "NotificationType": 2, "OathTokenTimeDrift": 0, "PhoneAppVersion": "6.7.12", "SecuredKeyId": 0, "SecuredPartitionId": 0, "TenantDeviceId": null, "TimeInterval": 0}		
AuthenticationType	3		
AuthenticatorFlavor	Authenticator		
DeviceId	00000000-0000-0000-0000-000000000000		
DeviceName	iPhone 14		
DeviceTag	SoftwareTokenActivated		
DeviceToken	apns2-abc123def456ghi789abc123def456ghi789abc123def456ghi789abc123def4560		
HashFunction	null		
Id	3d3d3d3d-4444-eeee-5555-6f6f6f6f6f		
LastAuthenticatedTimestamp	2001-01-01T00:00:00		
NotificationType	2		
OathTokenTimeDrift	0		
PhoneAppVersion	6.7.12		
SecuredKeyId	0		
SecuredPartitionId	0		
TenantDeviceId	null		
TimeInterval	0		
oldValue	[]		

Log Analytics can also be used to summarize user migration activity.

Kusto	
AuditLogs	
where ActivityDateTime > ago(7d)	
extend InitiatedBy = tostring(InitiatedBy["app"]["displayName"])	
where InitiatedBy == "Microsoft Entra multifactor authentication Management"	
extend UserObjectId = tostring(TargetResources[0]["id"])	

```
| summarize UsersMigrated = dcount(UserObjectId) by InitiatedBy,  
bin(ActivityDateTime, 1d)
```

InitiatedBy	ActivityDateTime [UTC]	UsersMigrated
> Azure MFA Management	8/10/2023, 12:00:00.000 AM	5
> Azure MFA Management	8/9/2023, 12:00:00.000 AM	4

Validate and test

Once you've successfully migrated user data, you can validate the end-user experience using Staged Rollout before making the global tenant change. The following process allows you to target specific Microsoft Entra group(s) for Staged Rollout for MFA. Staged Rollout tells Microsoft Entra ID to perform MFA by using Microsoft Entra multifactor authentication for users in the targeted groups, rather than sending them on-premises to perform MFA. You can validate and test—we recommend using the Microsoft Entra admin center, but if you prefer, you can also use Microsoft Graph.

Enable Staged Rollout

1. Navigate to the following url: [Enable staged rollout features - Microsoft Azure](#).
2. Change **Azure multifactor authentication** to **On**, and then select **Manage groups**.

Enable staged rollout features

Microsoft Entra ID

[Troubleshoot](#)[Got feedback?](#)

This feature is intended to help you transition from federation to cloud authentication. When your transition is complete, please change the tenant wide authentication method to cloud authentication. [Learn more.](#)

Password Hash Sync ⓘ

On Off
[Manage groups](#)

Pass-through authentication ⓘ

On Off
[Manage groups](#)

Seamless single sign-on ⓘ

On Off
[Manage groups](#)

Certificate-based authentication ⓘ

On Off
[Manage groups](#)

Azure multifactor authentication ⓘ

On Off
[Manage groups](#)

3. Select **Add groups** and add the group(s) containing users you wish to enable for Microsoft Entra multifactor authentication. Selected groups appear in the displayed list.

ⓘ Note

Any groups you target using the following Microsoft Graph method also appear in this list.

Manage groups for Azure multifactor authentication

Microsoft Entra ID

[Add Groups](#)[Refresh](#)[Remove Groups](#)[Troubleshoot](#)

- The maximum number of users in the initial configuration is 200 users. You can add additional users afterwards. For best results, validate in batches of 1,000 users.
- Dynamic and nested groups are not supported for staged rollout.

 Name

Group Id

 MFA users

7a12e67f-24f2-43a9-acdf-8358db3fdf87

Enable Staged Rollout using Microsoft Graph

1. Create the featureRolloutPolicy

a. Navigate to aka.ms/ge and login to Graph Explorer using a Hybrid Identity Administrator account in the tenant you wish to setup for Staged Rollout.

b. Ensure POST is selected targeting the following endpoint:

```
https://graph.microsoft.com/v1.0/policies/featureRolloutPolicies
```

c. The body of your request should contain the following (change **MFA rollout policy** to a name and description for your organization):

The screenshot shows the Microsoft Graph Explorer interface. The top bar has 'msgraph' selected. The main area shows a JSON object:

```
{  
    "displayName": "MFA rollout policy",  
    "description": "MFA rollout policy",  
    "feature": "multiFactorAuthentication",  
    "isEnabled": true,  
    "isAppliedToOrganization": false  
}
```

The bottom part of the interface shows the request configuration: 'POST' method, 'v1.0' version, and the URL 'https://graph.microsoft.com/v1.0/policies/featureRolloutPolicies'. The 'Request body' tab is selected, displaying the JSON above. Other tabs include 'Request headers', 'Modify permissions (Preview)', and 'Access token'.

d. Perform a GET with the same endpoint and make note of the ID value (crossed out in the following image):

The screenshot shows the Microsoft Graph Explorer interface with the 'Response preview' tab selected. The response is a JSON object:

```
{  
    "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#policies/featureRolloutPolicies",  
    "value": [  
        {  
            "id": "crossed_out_value",  
            "displayName": "MultiFactorAuthentication rollout policy2",  
            "description": "MultiFactorAuthentication rollout policy2",  
            "feature": "multiFactorAuthentication",  
            "isEnabled": true,  
            "isAppliedToOrganization": false  
        }  
    ]  
}
```

The 'id' field is crossed out with a red marker.

2. Target the Microsoft Entra group(s) that contain the users you wish to test

a. Create a POST request with the following endpoint (replace {ID of policy} with the ID value you copied from step 1d):

```
https://graph.microsoft.com/v1.0/policies/featureRolloutPolicies/{ID of  
policy}/appliesTo/$ref
```

- b. The body of the request should contain the following (replace {ID of group} with the object ID of the group you wish to target for staged rollout):

```
msgraph  
  
{  
  "@odata.id": "https://graph.microsoft.com/v1.0/directoryObjects/{ID  
  of group}"  
}
```

- c. Repeat steps a and b for any other groups you wish to target with staged rollout.
- d. You can view the current policy in place by doing a GET against the following URL:

```
https://graph.microsoft.com/v1.0/policies/featureRolloutPolicies/{policyID}  
?$.expand=appliesTo
```

The preceding process uses the [featureRolloutPolicy resource](#). The public documentation hasn't yet been updated with the new multifactorAuthentication feature, but it has detailed information on how to interact with the API.

3. Confirm that the end-user MFA experience. Here are a few things to check:
- Do users see their methods in [aka.ms/mfasetup](#)?
 - Do users receive phone calls/text messages?
 - Are they able to successfully authenticate using the above methods?
 - Do users successfully receive Authenticator notifications? Are they able to approve these notifications? Is authentication successful?
 - Are users able to authenticate successfully using Hardware OATH tokens?

Educate users

Ensure users know what to expect when they're moved to Microsoft Entra multifactor authentication, including new authentication flows. You may also wish to instruct users to use the Microsoft Entra ID Combined Registration portal ([aka.ms/mfasetup](#)) to manage their authentication methods rather than the User portal once migrations are complete. Any changes made to authentication methods in Microsoft Entra ID won't propagate back to your on-premises environment. In a situation where you had to roll back to MFA Server, any changes users have made in Microsoft Entra ID won't be available in the MFA Server User portal.

If you use third-party solutions that depend on Microsoft Entra multifactor authentication Server for authentication (see [Authentication services](#)), you'll want users to continue to make changes to their MFA methods in the User portal. These changes are synced to Microsoft Entra ID automatically. Once you've migrated these third party solutions, you can move users to the Microsoft Entra ID combined registration page.

Complete user migration

Repeat migration steps found in [Migrate user data](#) and [Validate and test](#) sections until all user data is migrated.

Migrate MFA Server dependencies

Using the data points you collected in [Authentication services](#), begin carrying out the various migrations necessary. Once this is completed, consider having users manage their authentication methods in the combined registration portal, rather than in the User portal on MFA server.

Update domain federation settings

Once you've completed user migrations, and moved all of your [Authentication services](#) off of MFA Server, it's time to update your domain federation settings. After the update, Microsoft Entra no longer sends MFA request to your on-premises federation server.

To configure Microsoft Entra ID to ignore MFA requests to your on-premises federation server, install the [Microsoft Graph PowerShell SDK](#) and set `federatedIdpMfaBehavior` to `rejectMfaByFederatedIdp`, as shown in the following example.

Request

```
HTTP  
  
PATCH  
https://graph.microsoft.com/beta/domains/contoso.com/federationConfiguration/6601d14b-d113-8f64-fda2-9b5ddda18ecc  
Content-Type: application/json  
{  
    "federatedIdpMfaBehavior": "rejectMfaByFederatedIdp"  
}
```

Response

ⓘ Note

The response object shown here might be shortened for readability.

HTTP

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "@odata.type": "#microsoft.graph.internalDomainFederation",
    "id": "6601d14b-d113-8f64-fda2-9b5ddda18ecc",
    "issuerUri": "http://contoso.com/adfs/services/trust",
    "metadataExchangeUri": "https://sts.contoso.com/adfs/services/trust/mex",
    "signingCertificate": "A1bC2dE3fH4iJ5kL6mN7oP8qR9sT0u",
    "passiveSignInUri": "https://sts.contoso.com/adfs/ls",
    "preferredAuthenticationProtocol": "wsFed",
    "activeSignInUri": "https://sts.contoso.com/adfs/services/trust/2005 usernamemixed",
    "signOutUri": "https://sts.contoso.com/adfs/ls",
    "promptLoginBehavior": "nativeSupport",
    "isSignedAuthenticationRequestRequired": true,
    "nextSigningCertificate": "A1bC2dE3fH4iJ5kL6mN7oP8qR9sT0u",
    "signingCertificateUpdateStatus": {
        "certificateUpdateResult": "Success",
        "lastRunDateTime": "2021-08-25T07:44:46.2616778Z"
    },
    "federatedIdpMfaBehavior": "rejectMfaByFederatedIdp"
}
```

Users are no longer redirected to your on-premises federation server for MFA, whether they're targeted by the Staged Rollout tool or not. Note this can take up to 24 hours to take effect.

ⓘ Note

The update of the domain federation setting can take up to 24 hours to take effect.

Optional: Disable MFA Server User portal

Once you've completed migrating all user data, end users can begin using the Microsoft Entra ID combined registration pages to manage MFA Methods. There are a couple ways to prevent users from using the User portal in MFA Server:

- Redirect your MFA Server User portal URL to aka.ms/mfasetup

- Clear the **Allow users to log in** checkbox under the **Settings** tab in the User portal section of MFA Server to prevent users from logging into the portal altogether.

Decommission MFA Server

When you no longer need the Microsoft Entra multifactor authentication server, follow your normal server deprecation practices. No special action is required in Microsoft Entra ID to indicate MFA Server retirement.

Rollback plan

If the upgrade had issues, follow these steps to roll back:

1. Uninstall MFA Server 8.1.
2. Replace PhoneFactor.pfdata with the backup made before upgrading.

ⓘ Note

Any changes since the backup was made are lost, but should be minimal if backup was made right before upgrade and upgrade was unsuccessful.

3. Run the installer for your previous version (for example, 8.0.x.x).
4. Configure Microsoft Entra ID to accept MFA requests to your on-premises federation server. Use Graph PowerShell to set `federatedIdpMfaBehavior` to `enforceMfaByFederatedIdp`, as shown in the following example.

Request

```
HTTP  
  
PATCH  
https://graph.microsoft.com/beta/domains/contoso.com/federationConfiguration/6601d14b-d113-8f64-fda2-9b5ddda18ecc  
Content-Type: application/json  
{  
    "federatedIdpMfaBehavior": "enforceMfaByFederatedIdp"  
}
```

The following response object is shortened for readability.

Response

HTTP

```
HTTP/1.1 200 OK
Content-Type: application/json
{
    "@odata.type": "#microsoft.graph.internalDomainFederation",
    "id": "6601d14b-d113-8f64-fda2-9b5ddda18ecc",
    "issuerUri": "http://contoso.com/adfs/services/trust",
    "metadataExchangeUri": "https://sts.contoso.com/adfs/services/trust/mex",
    "signingCertificate": "A1bC2dE3fH4iJ5kL6mN7oP8qR9sT0u",
    "passiveSignInUri": "https://sts.contoso.com/adfs/ls",
    "preferredAuthenticationProtocol": "wsFed",
    "activeSignInUri": "https://sts.contoso.com/adfs/services/trust/2005/usernamemixed",
    "signOutUri": "https://sts.contoso.com/adfs/ls",
    "promptLoginBehavior": "nativeSupport",
    "isSignedAuthenticationRequestRequired": true,
    "nextSigningCertificate": "A1bC2dE3fH4iJ5kL6mN7oP8qR9sT0u",
    "signingCertificateUpdateStatus": {
        "certificateUpdateResult": "Success",
        "lastRunDateTime": "2021-08-25T07:44:46.2616778Z"
    },
    "federatedIdpMfaBehavior": "enforceMfaByFederatedIdp"
}
```

Set the **Staged Rollout for Microsoft Entra multifactor authentication** to **Off**. Users are again redirected to your on-premises federation server for MFA.

Next steps

- Overview of how to migrate from MFA Server to Microsoft Entra multifactor authentication
- Migrate to cloud authentication using Staged Rollout

Feedback

Was this page helpful?

 Yes

 No

Provide product feedback ↗

Troubleshoot self-service password reset in Microsoft Entra ID

Article • 03/04/2025

Microsoft Entra self-service password reset (SSPR) lets users reset their passwords in the cloud.

For problems with SSPR, the following troubleshooting steps and common errors may help. You can also watch this short video on the [How to resolve the six most common SSPR end-user error messages](#).

If you can't find the answer to your problem, [our support teams are always available](#) to help.

SSPR configuration in the Microsoft Entra admin center

If you don't see certain SSPR options or you can't configure them in the Microsoft Entra admin center, review the following troubleshooting steps:

I don't see Password reset under Protection in the Microsoft Entra admin center

You don't see **Password reset** if you don't have a Microsoft Entra ID license assigned to the administrator performing the operation.

To assign a license to an administrator account, see [Assign, verify, and resolve problems with licenses](#).

I don't see a particular configuration option

Many elements of the UI are hidden until they're needed. Make sure the option is enabled before you look for the specific configuration options.

I don't see the On-premises integration tab

On-premises password writeback is only visible if you downloaded and configured Microsoft Entra Connect.

For more information, see [Getting started with Microsoft Entra Connect](#).

SSPR reporting

If you have problems with SSPR reporting in the Microsoft Entra admin center, review the following troubleshooting steps:

I see an authentication method that I disabled in the Add method option in combined registration

The combined registration takes into account three policies to determine what methods are shown in **Add method**:

- [Self-service password reset](#)
- [MFA](#)
- [Authentication methods](#)

If you disable app notifications in SSPR but enable it in MFA policy, that option appears in combined registration. For another example, if a user disables **Office phone** in SSPR, it's still displayed as an option if the user has the **Phone/Office** phone property set.

I don't see any password management activity types in the Self-Service Password Management audit event category

A Microsoft Entra ID license isn't assigned to the administrator performing the operation.

To assign a license to the administrator account in question, see [Assign, verify, and resolve problems with licenses](#).

User registrations show multiple times

When a user registers, we currently log each individual piece of data registered as a separate event.

If you want to aggregate this data and have greater flexibility in how you can view it, you can download the report and open the data as a pivot table in Excel.

SSPR registration portal

If your users have problems registering for SSPR, review the following troubleshooting steps:

The directory isn't enabled for password reset. The user may see an error that reports, "Your administrator has not enabled you to use this feature."

You can enable SSPR for all users, no users, or for selected groups of users. Only one Microsoft Entra group can currently be enabled for SSPR using the Microsoft Entra admin center. As part of a wider deployment of SSPR, nested groups are supported. Make sure that the users in the groups you choose are assigned the appropriate licenses.

In the Microsoft Entra admin center, change the **Self-service password reset enabled** configuration to *Selected* or *All* and then select **Save**.

The user doesn't have a Microsoft Entra ID license assigned. The user may see an error that reports, "Your administrator has not enabled you to use this feature."

Only one Microsoft Entra group can currently be enabled for SSPR using the Microsoft Entra admin center. As part of a wider deployment of SSPR, nested groups are supported. Make sure that the users in the groups you choose are assigned the appropriate licenses. Review the previous troubleshooting step to enable SSPR as required.

Also review troubleshooting steps to make sure that the administrator performing the configuration options has a license assigned. To assign a license to the administrator account in question, follow the steps to [Assign, verify, and resolve problems with licenses](#).

SSPR usage

To help resolve problems with SSPR, review these steps.

[+] Expand table

Error	Solution
The directory isn't enabled for password reset.	In the Microsoft Entra admin center, change the Self-service password reset enabled configuration to <i>Selected</i> or <i>All</i> and

Error	Solution
	then select Save .
The user doesn't have a Microsoft Entra ID license assigned.	A Microsoft Entra ID license isn't assigned to the desired user. To assign a license to the administrator account in question, follow the steps to Assign, verify, and resolve problems with licenses .
The directory is enabled for password reset, but the authentication information for the user is missing or malformed.	Make sure that user account is properly formed contact data on file in the directory. For more information, see Data used by Microsoft Entra self-service password reset .
The directory is enabled for password reset, but the user has only one piece of contact data on file when the policy is set to require two verification methods.	Make sure that the user has at least two properly configured contact methods. An example is having both a mobile phone number <i>and</i> an office phone number.
The directory is enabled for password reset and the user is properly configured, but the user is unable to be contacted.	A temporary service error, or there's incorrect contact data that we can't properly detect. If the user waits 10 seconds, a link is displayed to "Try again" and "Contact your administrator". If the user selects "Try again," it retries the call. If the user selects "Contact your administrator," it sends a form email to the administrators requesting a password reset to be performed for that user account.
The user never receives the password reset SMS or phone call.	The phone number in the directory may be malformed. Make sure the phone number is in the format "+1 4251234567". Password reset doesn't support extensions, even if you specify one in the directory. The extensions are stripped before the call is made. Use a number without an extension, or integrate the extension into the phone number in your private branch exchange (PBX).
The user never receives the password reset email.	The most common cause for this problem is that the message is rejected by a spam filter. Check your spam, junk, or deleted items folder for the email. Also, make sure the user checks the correct email account as registered with SSPR.
I set a password reset policy, but when an admin account uses password reset, that policy isn't applied.	Microsoft manages and controls the administrator password reset policy to ensure the highest level of security.

Error	Solution
The user is prevented from attempting a password reset too many times in a day.	An automatic throttling mechanism is used to block users from attempting to reset their passwords too many times in a short period of time. Throttling occurs the following scenarios: <ul style="list-style-type: none"> - The user attempts to validate a phone number five times in one hour. - The user attempts to use the security questions gate five times in one hour. - The user attempts to reset a password for the same user account five times in one hour. - If a user encounters this problem, they must wait 24 hours after the last attempt. The user can then reset their password.
The user sees an error when validating their phone number.	This error occurs when the phone number entered doesn't match the phone number on file. Make sure the user is entering the complete phone number, including the area and country code, when they attempt to use a phone-based method for password reset.
The user sees an error when using their email address.	If the UPN differs from the primary ProxyAddress/SMTPAddress of the user, the Sign-in to Microsoft Entra ID with email as an alternate login ID setting must be enabled for the tenant.
There's an error processing the request.	Generic SSPR registration errors can be caused by many issues, but generally this error is caused by either a service outage or a configuration issue. If you continue to see this generic error when you re-try the SSPR registration process, contact Microsoft support for help.
On-premises policy violation	The password doesn't meet the on-premises Active Directory password policy. The user must define a password that meets the complexity or strength requirements.
Password doesn't comply with fuzzy policy	The password that was used appears in the banned password list and can't be used. The user must define a password that meets or exceeds the banned password list policy.

SSPR errors that a user might see

The following errors and technical details may be shown to a user as part of the SSPR process. Often, the error isn't something they can resolve themselves, as the SSPR feature needs to be enabled, configured, or registered for their account.

Use the following information to understand the problem and what needs to be corrected on the Microsoft Entra tenant or individual user account.

Error	Details	Technical details
TenantSSPRFlagDisabled = 9	<p>We're sorry, you can't reset your password at this time because your administrator has disabled password reset for your organization.</p> <p>There's no further action you can take to resolve this situation. Contact your admin and ask them to enable this feature.</p> <p>To learn more, see Help, I forgot my Microsoft Entra password.</p>	SSPR_0009: We've detected that password reset hasn't been enabled by your administrator. Contact your admin and ask them to enable password reset for your organization.
WritebackNotEnabled = 10	<p>We're sorry, you can't reset your password at this time because your administrator hasn't enabled a necessary service for your organization.</p> <p>There's no further action you can take to resolve this situation. Contact your admin and ask them to check your organization's configuration.</p> <p>To learn more about this necessary service, see Configuring password writeback.</p>	SSPR_0010: We've detected that password writeback hasn't been enabled. Contact your admin and ask them to enable password writeback.

Error	Details	Technical details
SsprNotEnabledInUserPolicy = 11	<p>We're sorry, you can't reset your password at this time because your administrator hasn't configured password reset for your organization. There's no further action you can take to resolve this situation. Contact your admin and ask them to configure password reset.</p> <p>To learn more about password reset configuration, see Quickstart: Microsoft Entra self-service password reset.</p>	SSPR_0011: Your organization hasn't defined a password reset policy. Contact your admin and ask them to define a password reset policy.
UserNotLicensed = 12	<p>We're sorry, you can't reset your password at this time because required licenses are missing from your organization. There's no further action you can take to resolve this situation. Contact your admin and ask them to check your license assignment.</p> <p>To learn more about licensing, see Licensing requirements for Microsoft Entra</p>	SSPR_0012: Your organization doesn't have the required licenses necessary to perform password reset. Contact your admin and ask them to review the license assignments.

Error	Details	Technical details
	<p>self-service password reset.</p>	
UserNotMemberOfScopedAccessGroup = 13	<p>We're sorry, you can't reset your password at this time because your administrator hasn't configured your account to use password reset. There's no further action you can take to resolve this situation. Contact your admin and ask them to configure your account for password reset.</p> <p>To learn more about account configuration for password reset, see Roll out password reset for users.</p>	SSPR_0013: You aren't a member of a group enabled for password reset. Contact your admin and request to be added to the group.
UserNotProperlyConfigured = 14	<p>We're sorry, you can't reset your password at this time because necessary information is missing from your account. There's no further action you can take to resolve this situation. Contact your admin and ask them to reset your password for you. After you can access to your account again, you need to register the necessary</p>	SSPR_0014: Additional security info is needed to reset your password. To proceed, contact your admin and ask them to reset your password. After you can access to your account, you can register additional security info at https://aka.ms/ssprsetup . Your admin can add additional security info to your account by following the steps in Set and read authentication data for password reset .

Error	Details	Technical details
	<p>information.</p> <p>To register information, follow the steps in the Register for self-service password reset article.</p>	
OnPremisesAdminActionRequired = 29	<p>We're sorry, we can't reset your password at this time because of a problem with your organization's password reset configuration. There's no further action you can take to resolve this situation. Contact your admin and ask them to investigate.</p> <p>Or</p> <p>We can't reset your password at this time because of a problem with your organization's password reset configuration. There's no further action you can take to resolve this issue. Contact your admin and ask them to investigate.</p> <p>To learn more about the potential problem, see Troubleshoot</p>	<p>SSPR_0029: We're unable to reset your password due to an error in your on-premises configuration. Contact your admin and ask them to investigate.</p>

Error	Details	Technical details
	<p>password writeback.</p>	
OnPremisesConnectivityError = 30	<p>We're sorry, we can't reset your password at this time because of connectivity issues to your organization. There's no action to take right now, but the problem might be resolved if you try again later. If the problem persists, contact your admin and ask them to investigate.</p> <p>To learn more about connectivity issues, see Troubleshoot password writeback connectivity.</p>	SSPR_0030: We can't reset your password due to a poor connection with your on-premises environment. Contact your admin and ask them to investigate.
OnPremisesSuccessCloudFailure	<p>We've reset your password successfully, but you have to wait a few minutes before the changes are committed to the cloud. After these changes are committed, you can use your new password wherever you sign in with a work or school account.</p>	Password reset was successful on-premises, but there was an error while writing to the cloud. The error might be caused by a time-out, or a cloud password policy, throttling, or other reasons.

For general questions about Microsoft Entra ID and self-service password reset, you can ask the community for assistance on the [Microsoft Q&A question page for Microsoft Entra ID](#). Members of the community include engineers, product managers, MVPs, and fellow IT professionals.

Contact Microsoft support

If you can't find the answer to a problem, our support teams are always available to assist you further.

To properly assist you, we ask that you provide as much detail as possible when opening a case. These details include the following:

- **General description of the error:** What is the error? What was the behavior that was noticed? How can we reproduce the error? Provide as much detail as possible.
- **Page:** What page were you on when you noticed the error? Include the URL if you're able to and a screenshot of the page.
- **Support code:** What was the support code that was generated when the user saw the error?
 - To find this code, reproduce the error, then select the **Support code** link at the bottom of the screen and send the support engineer the GUID that results.

The screenshot shows a Microsoft browser window titled "Microsoft Online Password Reset". The URL is https://passwordreset.microsoftonline.com. A "Guest" user is logged in. The main content is a "Get back into your account" page. It asks "Who are you?" and provides instructions to enter a User ID and verify characters from a CAPTCHA image or audio file. The CAPTCHA image shows the letters "6R SQ". Below it is a text input field and a red box containing the text "1de0e8a3-d162-4acd-abb7-0a361721de1a". At the bottom, there are "Next" and "Cancel" buttons, and a "Support code" link.

- If you're on a page without a support code at the bottom, select F12 and search for the SID and CID and send those two results to the support engineer.
- **Date, time, and time zone:** Include the precise date and time *with the time zone* that the error occurred.
- **User ID:** Who was the user who saw the error? An example is *user@contoso.com*.
 - Is this a federated user?
 - Is this a pass-through authentication user?
 - Is this a password-hash-synchronized user?
 - Is this a cloud-only user?
- **Licensing:** Is the user assigned a Microsoft Entra ID license?
- **Application event log:** If you're using password writeback and the error is in your on-premises infrastructure, include a zipped copy of your Application event log from the Microsoft Entra Connect server.

Next steps

To learn more about SSPR, see [How it works: Microsoft Entra self-service password reset](#) or [How does self-service password reset writeback work in Microsoft Entra ID?](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Troubleshoot self-service password reset writeback in Microsoft Entra ID

Article • 03/04/2025

Microsoft Entra self-service password reset (SSPR) lets users reset their passwords in the cloud. Password writeback is a feature enabled with [Microsoft Entra Connect](#) or [cloud sync](#) that allows password changes in the cloud to be written back to an existing on-premises directory in real time.

If you have problems with SSPR writeback, the following troubleshooting steps and common errors may help. If you can't find the answer to your problem, [our support teams are always available](#) to assist you further.

Troubleshoot connectivity

If you have problems with password writeback for Microsoft Entra Connect, review the following steps that may help resolve the problem. To recover your service, we recommend that you follow these steps in order:

- [Confirm network connectivity](#)
- [Check TLS 1.2](#)
- [Update Microsoft .NET 4.8](#)
- [Restart the Microsoft Entra Connect Sync service](#)
- [Disable and re-enable the password writeback feature](#)
- [Install the latest Microsoft Entra Connect release](#)
- [Troubleshoot password writeback](#)

Confirm network connectivity

The most common point of failure is that firewall or proxy ports, or idle time-outs are incorrectly configured.

For Microsoft Entra Connect version *1.1.443.0* and higher, *outbound HTTPS* access is required to the following addresses:

- **.passwordreset.microsoftonline.com*
- **.servicebus.windows.net*

[Azure for US Government endpoints:](#)

- **.passwordreset.microsoftonline.us*

- *.servicebus.usgovcloudapi.net

Azure China 21Vianet endpoints:

- ssprdedicatedsbmcprodne.servicebus.chinacloudapi.cn
- ssprdedicatedsbmcprodnn.servicebus.chinacloudapi.cn

If you need more granularity, see the [list of Microsoft Azure IP Ranges and Service Tags for Public Cloud ↗](#).

For Azure for US Government, see the [list of Microsoft Azure IP Ranges and Service Tags for Azure for US Government Cloud ↗](#).

These files are updated weekly.

To determine if access to a URL and port is restricted in an environment such as global Azure cloud, follow these steps:

1. On the Entra connect server, open the event viewer logs (Windows logs, application) and locate one of these event IDs: 31034 or 31019.
2. From these Event IDs, identify the name of the service bus listener:

Event ID	Task Category
31030	None
31030	None
31019	None
31019	None

3. Run the following cmdlet:

```
PowerShell
Test-NetConnection -ComputerName <namespace>.servicebus.windows.net -
Port 443
```

Or run the following:

```
PowerShell
```

```
Invoke-WebRequest -Uri https://<namespace>.servicebus.windows.net -  
Verbose
```

Replace the <namespace> with the same you extracted from the event IDs previously. For example, in the preceding case, the command is:

PowerShell

```
Test-NetConnection -ComputerName ssprdedicatedsbprodfra-  
1.servicebus.windows.net -Port 443
```

For more information, see the [connectivity prerequisites for Microsoft Entra Connect](#).

Check if TLS 1.2 is enabled

An additional troubleshooting step is to check that TLS 1.2 is enabled correctly on the Sync Server. Run [PowerShell Script to check TLS 1.2 on Entra Connect Server](#). Make sure to run the script in Admin Mode.

Output from check script that must look like the following image (the path, name and value columns) to be enabled correctly. If it doesn't, run the [PowerShell Script to enable TLS 1.2 on Entra Connect Server](#). Then reboot the server, and run script to check TLS 1.2 again.

Make sure Microsoft .NET Framework 4.8 or higher is enabled (Sync Server)

Make sure Microsoft .NET Framework 4.8 or higher is enabled on the Sync Server.

- [How to check that .NET is already installed](#)
- [Query the registry using PowerShell](#)
- [Download .NET framework ↗](#)

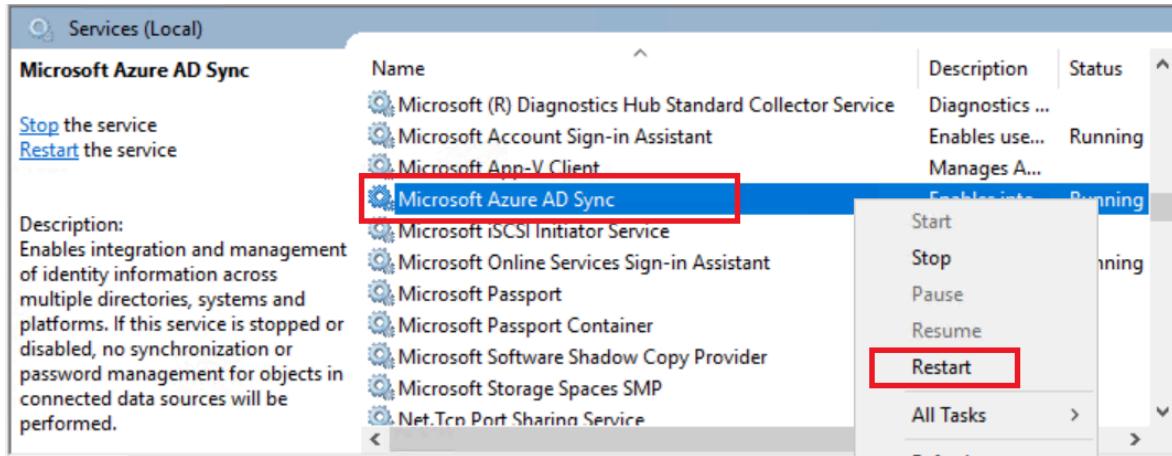
Restart the Microsoft Entra Connect Sync service

To resolve connectivity issues or other transient problems with the service, complete the following steps to restart the Microsoft Entra Connect Sync service:

1. As an administrator on the server that runs Microsoft Entra Connect, select **Start**.
2. Enter *services.msc* in the search field and select **Enter**.

3. Look for the *Azure AD Sync* entry.

4. Right-click the service entry, select **Restart**, and wait for the operation to finish.



These steps re-establish your connection with Microsoft Entra ID and should resolve your connectivity issues.

If restarting the Microsoft Entra Connect Sync service doesn't resolve your problem, try to disable and then re-enable the password writeback feature in the next section.

Disable and re-enable the password writeback feature

To continue to troubleshoot issues, complete the following steps to disable and then re-enable the password writeback feature:

1. As an administrator on the server that runs Microsoft Entra Connect, open the **Microsoft Entra Connect Configuration wizard**.
2. In **Connect to Microsoft Entra ID**, enter your Microsoft Entra Hybrid Administrator credentials.
3. In **Connect to AD DS**, enter your on-premises Active Directory Domain Services admin credentials.
4. In **Uniquely identifying your users**, select the **Next** button.
5. In **Optional features**, clear the **Password writeback** check box.
6. Select **Next** through the remaining dialog pages without changing anything until you get to the **Ready to configure** page.
7. Check that the **Ready to configure** page shows the *Password writeback* option as *disabled*. Select the green **Configure** button to commit your changes.
8. In **Finished**, clear the **Synchronize now** option, and then select **Finish** to close the wizard.
9. Reopen the **Microsoft Entra Connect Configuration wizard**.
10. Repeat steps 2-8, this time selecting the *Password writeback* option on the **Optional features** page to re-enable the service.

These steps re-establish your connection with Microsoft Entra ID and should resolve your connectivity issues.

If disabling and then re-enabling the password writeback feature doesn't resolve your problem, reinstall Microsoft Entra Connect in the next section.

Install the latest Microsoft Entra Connect release

Reinstalling Microsoft Entra Connect can resolve configuration and connectivity issues between Microsoft Entra ID and your local Active Directory Domain Services environment. We recommend that you perform this step only after you attempt the previous steps to verify and troubleshoot connectivity.

Warning

If you customized the out-of-the-box sync rules, *back them up before you proceed with the upgrade, then manually redeploy them after you're finished.*

1. Download the latest version of Microsoft Entra Connect from the [Microsoft Download Center](#).
2. As you already installed Microsoft Entra Connect, perform an in-place upgrade to update your Microsoft Entra Connect installation to the latest version.

Run the downloaded package and follow the on-screen instructions to update Microsoft Entra Connect.

These steps should re-establish your connection with Microsoft Entra ID and resolve your connectivity issues.

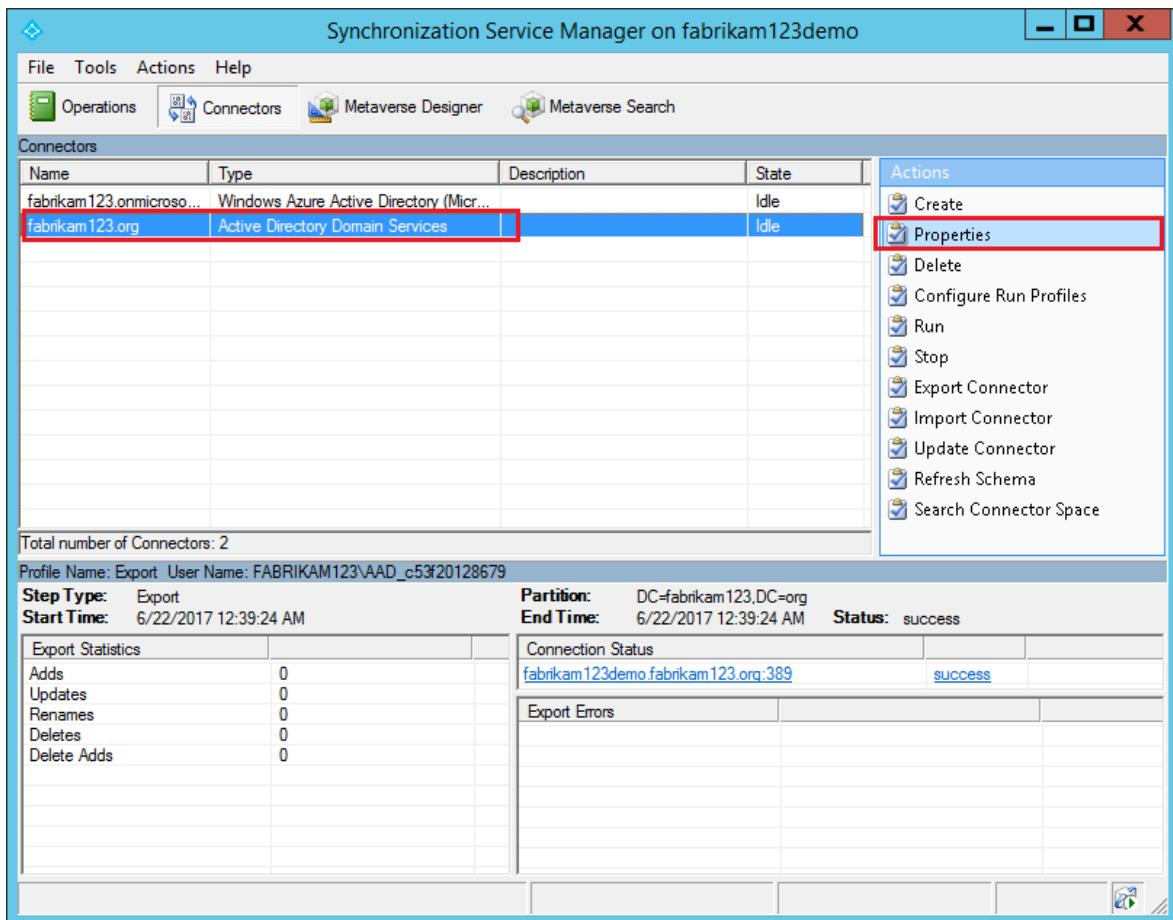
If installing the latest version of the Microsoft Entra Connect server doesn't resolve your problem, try disabling and then re-enabling password writeback as a final step after you install the latest release.

Verify that Microsoft Entra Connect has the required permissions

Microsoft Entra Connect requires AD DS Reset password permission to perform password writeback. To check if Microsoft Entra Connect has the required permission for a given on-premises AD DS user account, use the [Windows Effective Permission](#) feature:

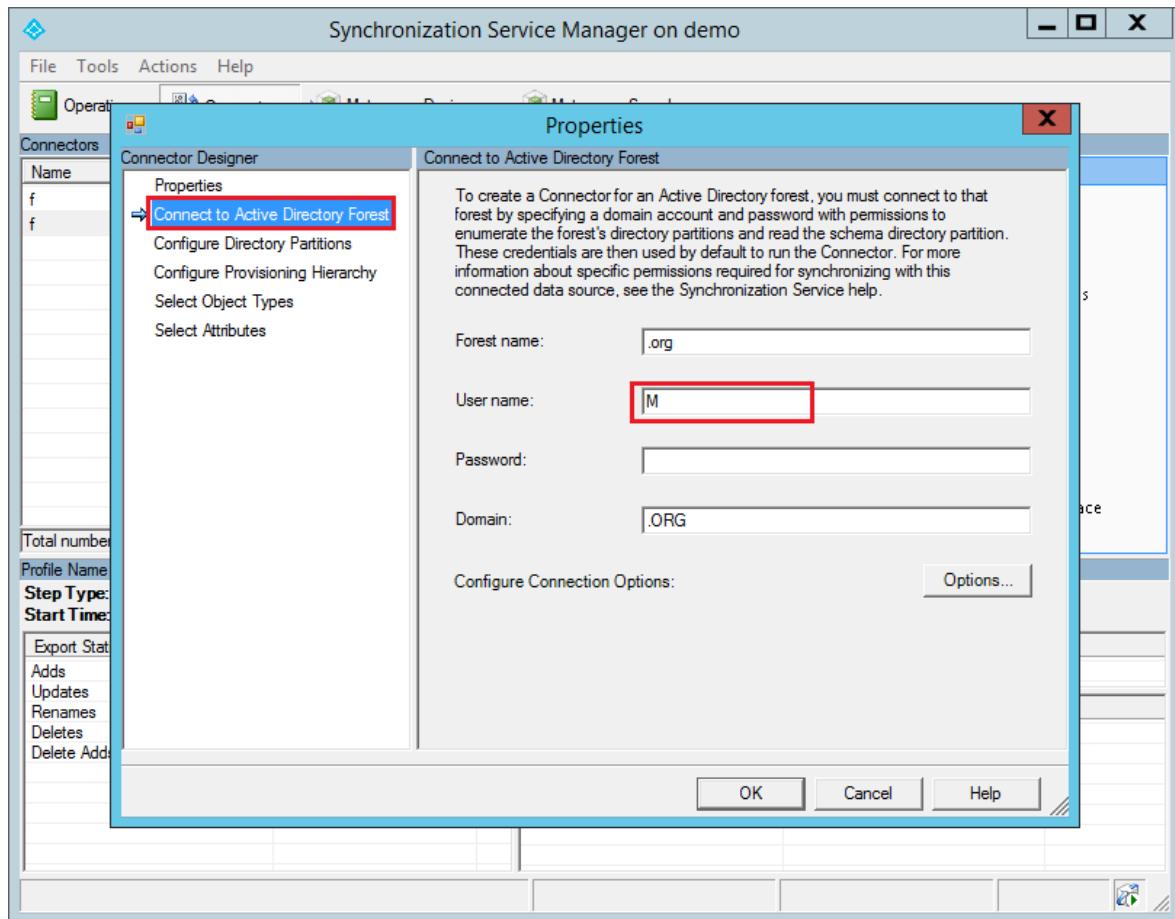
1. Sign in to the Microsoft Entra Connect server and start the **Synchronization Service Manager** by selecting **Start > Synchronization Service**.

2. Under the **Connectors** tab, select the on-premises **Active Directory Domain Services** connector, and then select **Properties**.

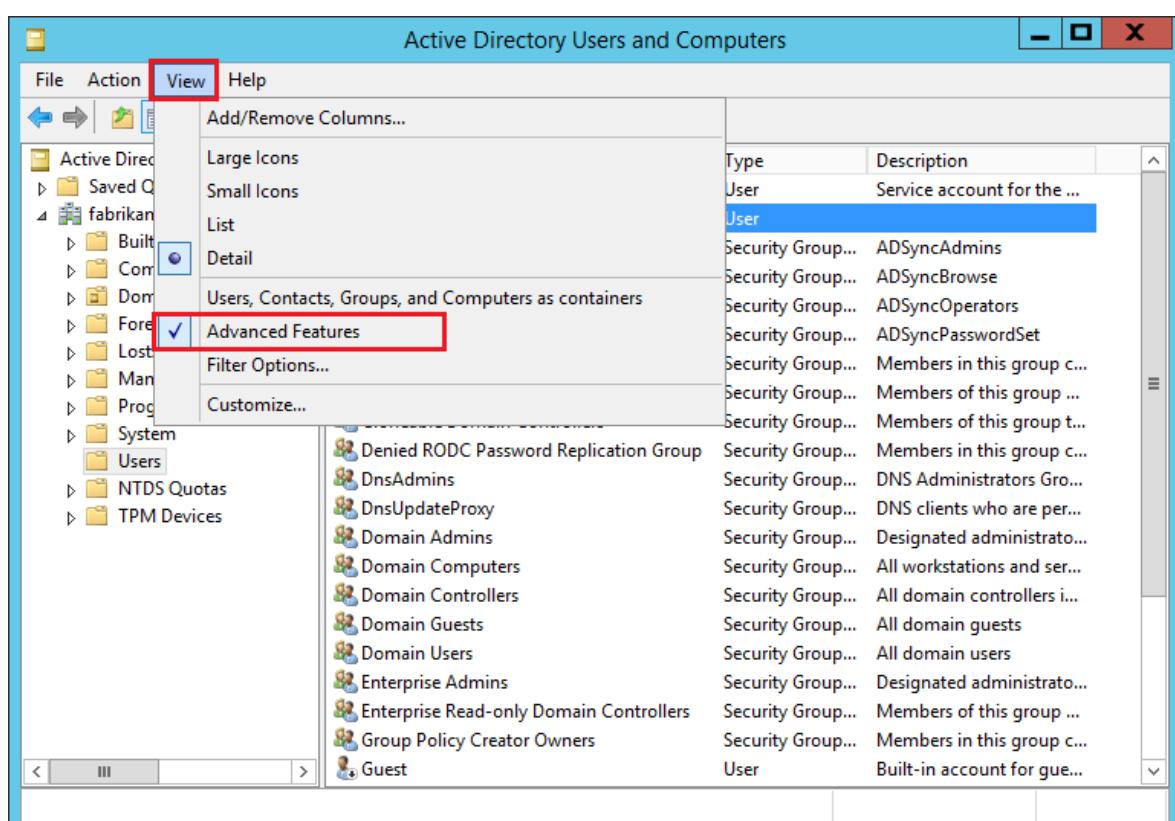


3. In the pop-up window, select **Connect to Active Directory Forest** and make note of the **User name** property. This property is the AD DS account used by Microsoft Entra Connect to perform directory synchronization.

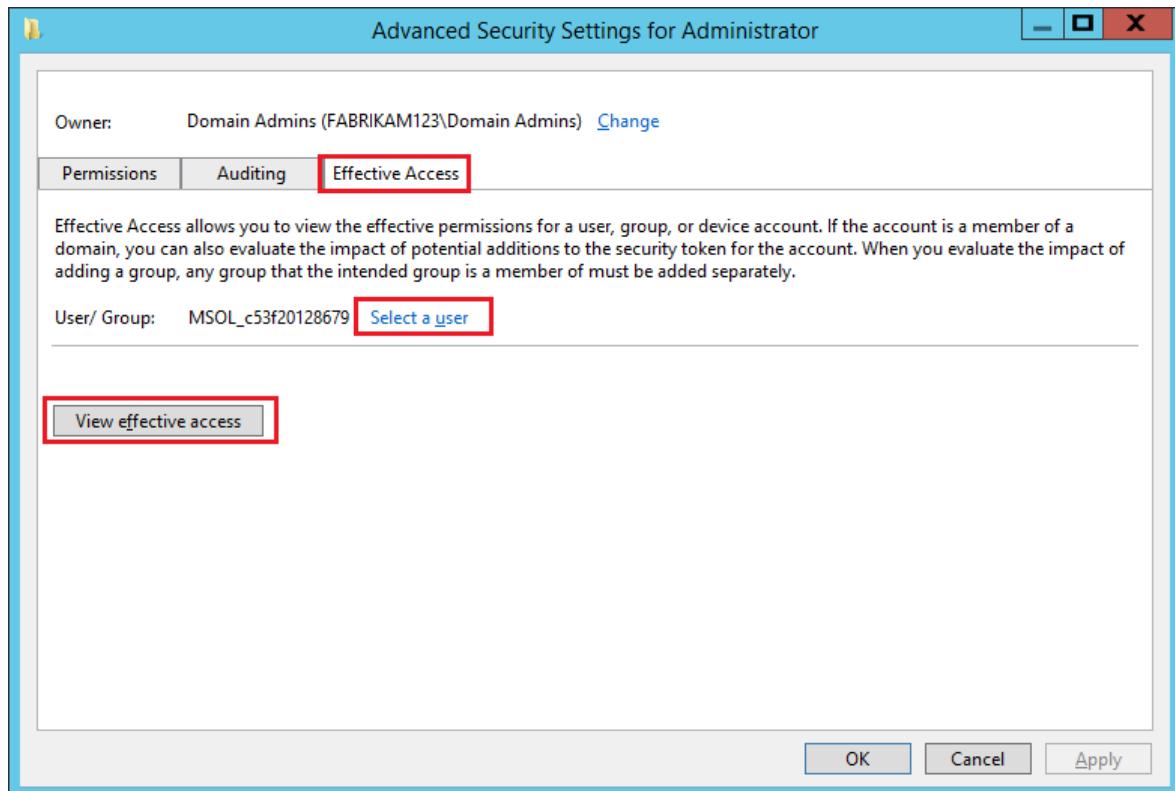
For Microsoft Entra Connect to perform password writeback, the AD DS account must have reset password permission. You check the permissions on this user account in the following steps.



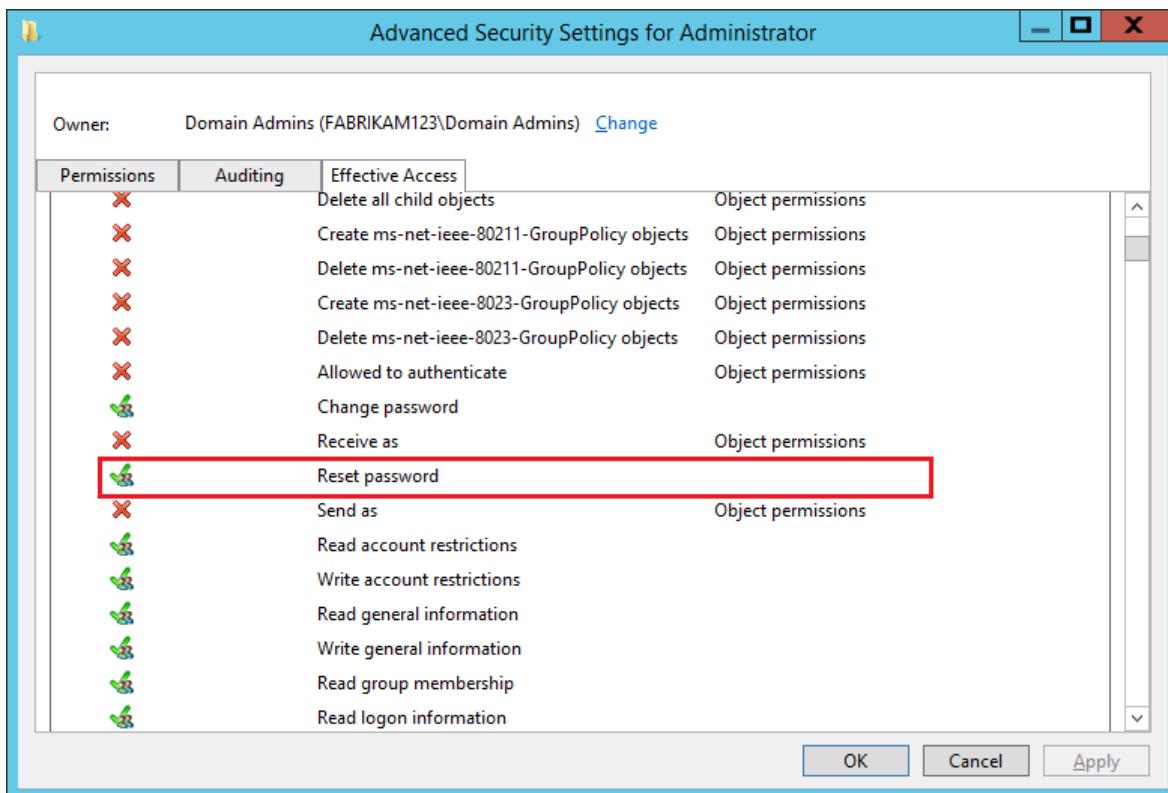
4. Sign in to an on-premises domain controller and start the **Active Directory Users and Computers** application.
5. Select **View** and make sure the **Advanced Features** option is enabled.



6. Look for the AD DS user account you want to verify. Right-click the account name and select **Properties**.
7. In the pop-up window, go to the **Security** tab and select **Advanced**.
8. In the **Advanced Security Settings for Administrator** pop-up window, go to the **Effective Access** tab.
9. Choose **Select a user**, select the AD DS account used by Microsoft Entra Connect, and then select **View effective access**.



10. Scroll down and look for **Reset password**. If the entry has a check mark, the AD DS account has permission to reset the password of the selected Active Directory user account.



Common password writeback errors

The following more specific issues may occur with password writeback. If you have one of these errors, review the proposed solution and check if password writeback then works correctly.

[\[+\] Expand table](#)

Error	Solution
The password reset service doesn't start on-premises. Error 6800 appears in the Microsoft Entra Connect machine's Application event log.	When password writeback is enabled, the sync engine calls the writeback library to perform the configuration (onboarding) by communicating to the cloud onboarding service. Any errors encountered during onboarding or while starting the Windows Communication Foundation (WCF) endpoint for password writeback results in errors in the event log, on your Microsoft Entra Connect machine.
After onboarding, federated, pass-through authentication, or password-hash-synchronized users can't reset their passwords.	During restart of the Azure AD Sync (ADSync) service, if writeback was configured, the WCF endpoint starts up. But, if the startup of the endpoint fails, we log event 6800 and let the sync service start up. The presence of this event means that the password writeback endpoint didn't start up. Event log details for this event 6800, along with event log entries generate by the PasswordResetService component, indicate why you can't start up the endpoint. Review these event log errors and try to restart the Microsoft Entra

Error	Solution
	Connect if password writeback still isn't working. If the problem persists, try to disable and then re-enable password writeback.
When a user attempts to reset a password or unlock an account with password writeback enabled, the operation fails. In addition, you see an event in the Microsoft Entra Connect event log that contains: "Synchronization Engine returned an error hr=800700CE, message="The filename or extension is too long" after the unlock operation occurs.	Find the Active Directory account for Microsoft Entra Connect and reset the password so that it contains no more than 256 characters. Next, open the Synchronization Service from the Start menu. Browse to Connectors and find the Active Directory Connector . Select it and then select Properties . Browse to the Credentials page and enter the new password. Select OK to close the page.
At the last step of the Microsoft Entra Connect installation process, you see an error indicating that password writeback couldn't be configured. The Microsoft Entra Connect Application event log contains error 32009 with the text "Error getting auth token."	This error occurs in the following two cases: <ul style="list-style-type: none"> • You specified an incorrect password for the Hybrid Administrator account provided at the beginning of the Microsoft Entra Connect installation process. • You attempted to use a federated user for the Hybrid Administrator account specified at the beginning of the Microsoft Entra Connect installation process. To fix this problem, make sure that you're not using a federated account for the Hybrid Administrator you specified at the beginning of the installation process, and that the password specified is correct.
The Microsoft Entra Connect machine event log contains error 32002 that is thrown by running PasswordResetService. The error reads: "Error Connecting to ServiceBus. The token provider was unable to provide a security token."	Your on-premises environment isn't able to connect to the Azure Service Bus endpoint in the cloud. A firewall rule blocking an outbound connection to a particular port or web address normally causes this error. See Connectivity prerequisites for more info. After you update these rules, restart the Microsoft Entra Connect server and password writeback should start working again.
After working for some time, federated, pass-through authentication, or	In some rare cases, the password writeback service can fail to restart when Microsoft Entra Connect has restarted. In these cases, first check if password writeback is enabled on-premises. You can

Error	Solution
password-hash-synchronized users can't reset their passwords.	check by using either the Microsoft Entra Connect wizard or PowerShell. If the feature appears to be enabled, try enabling or disabling the feature again either. If this troubleshooting step doesn't work, try a complete uninstall and reinstall of Microsoft Entra Connect.
<p>Federated, pass-through authentication, or password-hash-synchronized users who attempt to reset their passwords see an error after attempting to submit their password. The error indicates that there was a service problem.</p> <p>In addition to this problem, during password reset operations, you might see an error that the management agent was denied access in your on-premises event logs.</p>	<p>If you see these errors in your event log, check that the Active Directory Management Agent (ADMA) account specified during configuration has the necessary permissions for password writeback.</p> <p>After this permission is given, it can take up to one hour for the permissions to trickle down via the <code>sdprop</code> background task on the domain controller (DC).</p> <p>For password reset to work, the permission needs to be stamped on the security descriptor of the user object whose password is being reset. Until this permission shows up on the user object, password reset continues to fail with an access denied message.</p>
<p>Federated, pass-through authentication, or password-hash-synchronized users who attempt to reset their passwords, see an error after they submit their password. The error indicates that there was a service problem.</p> <p>In addition to this problem, during password reset operations, you might see an error in your event logs from the Microsoft Entra Connect service indicating an "Object could not be found" error.</p>	<p>This error usually indicates that the sync engine is unable to find either the user object in the Microsoft Entra connector space or the linked metaverse (MV) or Microsoft Entra connector space object.</p> <p>To troubleshoot this problem, make sure that the user is indeed synchronized from on-premises to Microsoft Entra ID via the current instance of Microsoft Entra Connect and inspect the state of the objects in the connector spaces and MV. Confirm that the Active Directory Certificate Services (AD CS) object is connected to the MV object via the "Microsoft.InfromADUserAccountEnabled.xxx" rule.</p>
Federated, pass-through authentication, or password-hash-	This indicates that the sync engine detected that the MV object is connected to more than one AD CS object via "Microsoft.InfromADUserAccountEnabled.xxx". This means that the

Error	Solution
<p>synchronized users who attempt to reset their passwords see an error after they submit their password. The error indicates that there was a service problem.</p> <p>In addition to this problem, during password reset operations, you might see an error in your event logs from the Microsoft Entra Connect service that indicates that there's a "Multiple matches found" error.</p>	<p>user has an enabled account in more than one forest. This scenario isn't supported for password writeback.</p>
<p>Password operations fail with a configuration error. The Application event log contains Microsoft Entra Connect error 6329 with the text "0x8023061f (The operation failed because password synchronization isn't enabled on this Management Agent)".</p>	<p>This error occurs if the Microsoft Entra Connect configuration is changed to add a new Active Directory forest (or to remove and read an existing forest) after the password writeback feature has already been enabled. Password operations for users in these recently added forests fail. To fix the problem, disable and then re-enable the password writeback feature after the forest configuration changes are complete.</p>
<p>SSPR_0029: We are unable to reset your password due to an error in your on-premises configuration. Please contact your admin and ask them to investigate.</p>	<p>Problem: Password writeback is enabled following all of the required steps, but when attempting to change a password you receive "SSPR_0029: Your organization hasn't properly set up the on-premises configuration for password reset." Checking the event logs on the Microsoft Entra Connect system shows that the management agent credential was denied access. Possible Solution: Use RSOP on the Microsoft Entra Connect system and your domain controllers to see if the policy "Network access: Restrict clients allowed to make remote calls to SAM" found under Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options is enabled. Edit the policy to include the MSOL_XXXXXXX management account as an allowed user. For more information, see Troubleshoot error SSPR_0029: Your organization hasn't properly set up the on-premises configuration for password reset.</p>

Password writeback event log error codes

A best practice when you troubleshoot problems with password writeback is to inspect the Application event log, on your Microsoft Entra Connect machine. This event log contains events from two sources for password writeback. The *PasswordResetService* source describes operations and problems related to the operation of password writeback. The *ADSync* source describes operations and problems related to setting passwords in your Active Directory Domain Services environment.

If the source of the event is ADSync

[] [Expand table](#)

Code	Name or message	Description
6329	BAIL: MMS(4924) 0x80230619: "A restriction prevents the password from being changed to the current one specified."	<p>This event occurs when the password writeback service attempts to set a password on your local directory that doesn't meet the password age, history, complexity, or filtering requirements of the domain. This event can also occur if a password can't be changed for a user.</p> <p>If you have a minimum password age and have recently changed the password within that window of time, you're not able to change the password again until it reaches the specified age in your domain. For testing purposes, the minimum age should be set to 0.</p> <p>If you have password history requirements enabled, then you must select a password that hasn't been used in the last N times, where N is the password history setting. If you do select a password that's used in the last N times, then you see a failure in this case. For testing purposes, the password history should be set to 0.</p> <p>If you have password complexity requirements, all of them are enforced when the user attempts to change or reset a password.</p> <p>If you have password filters enabled and a user selects a password that doesn't meet the filtering criteria, then the reset or change operation fails.</p> <p>If the user has the PASSWD_CANT_CHANGE property flag set, their password can't be synced. For testing purposes, remove the</p>

Code	Name or message	Description
		PASSWD_CANT_CHANGE property flag. For more information, see Property flag descriptions .
6329	MMS(3040): admaexport.cpp(2837): The server doesn't contain the LDAP password policy control.	This problem occurs if LDAP_SERVER_POLICY_HINTS_OID control (1.2.840.113556.1.4.2066) isn't enabled on the DCs. To use the password writeback feature, you must enable the control. To do so, the DCs must be on Windows Server 2016 or later.

HR 8023042	Synchronization Engine returned an error hr=80230402, message=An attempt to get an object failed because there are duplicated entries with the same anchor.	<p>This error occurs when the same user ID is enabled in multiple domains. An example is if you're syncing account and resource forests and have the same user ID present and enabled in each forest.</p> <p>This error can also occur if you use a non-unique anchor attribute, like an alias or UPN, and two users share that same anchor attribute.</p> <p>To resolve this problem, ensure that you don't have any duplicated users within your domains and that you use a unique anchor attribute for each user.</p>
---------------	---	--

If the source of the event is PasswordResetService

[] Expand table

Code	Name or message	Description
31001	PasswordResetStart	This event indicates that the on-premises service detected a password reset request for a federated, pass-through authentication, or password-hash-synchronized user that originates from the cloud. This event is the first event in every password-reset writeback operation.
31002	PasswordResetSuccess	This event indicates that a user selected a new password during a password-reset operation. We determined that this password meets corporate password requirements. The password is successfully written back to the local Active Directory environment.
31003	PasswordResetFail	This event indicates that a user selected a password and the password arrived successfully to the on-premises environment. But when we attempted to set the password in the local Active Directory environment, a failure occurred. This failure can happen for several reasons:

Code	Name or message	Description
		<ul style="list-style-type: none"> The user's password doesn't meet the age, history, complexity, or filter requirements for the domain. To resolve this problem, create a new password. The ADMA service account doesn't have the appropriate permissions to set the new password on the user account in question. The user's account is in a protected group, such as domain or enterprise admin group, which disallows password set operations.
31004	OnboardingEventStart	This event occurs if you enable password writeback with Microsoft Entra Connect and we've started onboarding your organization to the password writeback web service.
31005	OnboardingEventSuccess	This event indicates that the onboarding process was successful and that the password writeback capability is ready to use.
31006	ChangePasswordStart	This event indicates that the on-premises service detected a password change request for a federated, pass-through authentication, or password-hash-synchronized user that originates from the cloud. This event is the first event in every password-change writeback operation.
31007	ChangePasswordSuccess	This event indicates that a user selected a new password during a password change operation, we determined that the password meets corporate password requirements, and that the password is successfully written back to the local Active Directory environment.
31008	ChangePasswordFail	<p>This event indicates that a user selected a password and that the password arrived successfully to the on-premises environment, but when we attempted to set the password in the local Active Directory environment, a failure occurred. This failure can happen for several reasons:</p> <ul style="list-style-type: none"> The user's password doesn't meet the age, history, complexity, or filter requirements for the domain. To resolve this problem, create a new password. The ADMA service account doesn't have the appropriate permissions to set the new password on the user account in question. The user's account is in a protected group, such as domain or enterprise admins, which disallow password set operations.
31009	ResetUserPasswordByAdminStart	The on-premises service detected a password reset request for a federated, pass-through authentication, or

Code	Name or message	Description
		password-hash-synchronized user originating from the administrator on behalf of a user. This event is the first event in every password-reset writeback operation that is initiated by an administrator.
31010	ResetUserPasswordByAdminSuccess	The admin selected a new password during an admin-initiated password-reset operation. We determined that this password meets corporate password requirements. The password is successfully written back to the local Active Directory environment.
31011	ResetUserPasswordByAdminFail	<p>The admin selected a password on behalf of a user. The password arrived successfully to the on-premises environment. But when we attempted to set the password in the local Active Directory environment, a failure occurred. This failure can happen for several reasons:</p> <ul style="list-style-type: none"> • The user's password doesn't meet the age, history, complexity, or filter requirements for the domain. Try a new password to resolve this problem. • The ADMA service account doesn't have the appropriate permissions to set the new password on the user account in question. • The user's account is in a protected group, such as domain or enterprise admins, which disallow password set operations.
31012	OffboardingEventStart	This event occurs if you disable password writeback with Microsoft Entra Connect and indicates that we started offboarding your organization to the password writeback web service.
31013	OffboardingEventSuccess	This event indicates that the offboarding process was successful and that password writeback capability is successfully disabled.
31014	OffboardingEventFail	This event indicates that the offboarding process wasn't successful. This might be due to a permissions error on the cloud or on-premises administrator account specified during configuration. The error can also occur if you're attempting to use a federated cloud Hybrid Administrator when disabling password writeback. To fix this problem, check your administrative permissions and ensure that you're not using a federated account while configuring the password writeback capability.
31015	WriteBackServiceStarted	This event indicates that the password writeback service started successfully. It is ready to accept password

Code	Name or message	Description
		management requests from the cloud.
31016	WriteBackServiceStopped	This event indicates that the password writeback service stopped. Any password management requests from the cloud won't be successful.
31017	AuthTokenSuccess	This event indicates that we successfully retrieved an authorization token for the Hybrid Administrator specified during Microsoft Entra Connect setup to start the offboarding or onboarding process.
31018	KeyPairCreationSuccess	This event indicates that we successfully created the password encryption key. This key is used to encrypt passwords from the cloud to be sent to your on-premises environment.
31019	ServiceBusHeartBeat	This event indicates that we successfully sent a request to your tenant's Service Bus instance.
31034	ServiceBusListenerError	This event indicates that there was an error connecting to your tenant's Service Bus listener. If the error message includes "The remote certificate is invalid", check to make sure that your Microsoft Entra Connect server has all the required Root CAs as described in Azure TLS certificate changes .
31044	PasswordResetService	This event indicates that password writeback isn't working. The Service Bus listens for requests on two separate relays for redundancy. Each relay connection is managed by a unique Service Host. The writeback client returns an error if either Service Host isn't running.
32000	UnknownError	This event indicates an unknown error occurred during a password management operation. Look at the exception text in the event for more details. If you're having problems, try disabling and then re-enabling password writeback. If this doesn't help, include a copy of your event log along with the tracking ID specified when you open a support request.
32001	ServiceError	This event indicates there was an error connecting to the cloud password reset service. This error generally occurs when the on-premises service was unable to connect to the password-reset web service.
32002	ServiceBusError	This event indicates there was an error connecting to your tenant's Service Bus instance. This can happen if you're blocking outbound connections in your on-premises environment. Check your firewall to ensure that you allow

Code	Name or message	Description
		<p>connections over TCP 443 and to https://ssprdedicatedsbprodncu.servicebus.windows.net, and then try again. If you're still having problems, try disabling and then re-enabling password writeback.</p>
32003	InPutValidationError	<p>This event indicates that the input passed to our web service API was invalid. Try the operation again.</p>
32004	DecryptionError	<p>This event indicates that there was an error decrypting the password that arrived from the cloud. This might be due to a decryption key mismatch between the cloud service and your on-premises environment. To resolve this problem, disable and then re-enable password writeback in your on-premises environment.</p>
32005	ConfigurationError	<p>During onboarding, we save tenant-specific information in a configuration file in your on-premises environment. This event indicates that there was an error saving this file or that when the service was started, there was an error reading the file. To fix this problem, try disabling and then re-enabling password writeback to force a rewrite of the configuration file.</p>
32007	OnBoardingConfigUpdateError	<p>During onboarding, we send data from the cloud to the on-premises password-reset service. That data is then written to an in-memory file before it's sent to the sync service to be stored securely on disk. This event indicates that there's a problem with writing or updating that data in memory. To fix this problem, try disabling and then re-enabling password writeback to force a rewrite of this configuration file.</p>
32008	ValidationError	<p>This event indicates we received an invalid response from the password-reset web service. To fix this problem, try disabling and then re-enabling password writeback.</p>
32009	AuthTokenError	<p>This event indicates that we couldn't get an authorization token for the Hybrid Administrator account specified during Microsoft Entra Connect setup. This error can be caused by a bad username or password specified for the Hybrid Administrator account. This error can also occur if the Hybrid Administrator account specified is federated. To fix this problem, rerun the configuration with the correct username and password and ensure that the administrator is a managed (cloud-only or password-synchronized) account.</p>
32010	CryptoError	<p>This event indicates there was an error generating the password encryption key or decrypting a password that</p>

Code	Name or message	Description
		arrives from the cloud service. This error likely indicates a problem with your environment. Look at the details of your event log to learn more about how to resolve this problem. You can also try disabling and then re-enabling the password writeback service.
32011	OnBoardingServiceError	This event indicates that the on-premises service couldn't properly communicate with the password-reset web service to initiate the onboarding process. This can happen as a result of a firewall rule or if there's a problem getting an authentication token for your tenant. To fix this problem, ensure that you're not blocking outbound connections over TCP 443 and TCP 9350-9354 or to https://ssprdedicatedsbprodncu.servicebus.windows.net . Also ensure that the Microsoft Entra admin account you're using to onboard isn't federated.
32013	OffBoardingError	This event indicates that the on-premises service couldn't properly communicate with the password-reset web service to initiate the offboarding process. This can happen as a result of a firewall rule or if there's a problem getting an authorization token for your tenant. To fix this problem, ensure that you're not blocking outbound connections over 443 or to https://ssprdedicatedsbprodncu.servicebus.windows.net , and that the Microsoft Entra admin account you're using to offboard isn't federated.
32014	ServiceBusWarning	This event indicates that we had to retry to connect to your tenant's Service Bus instance. Under normal conditions, this shouldn't be a concern, but if you see this event many times, consider checking your network connection to Service Bus, especially if it's a high-latency or low-bandwidth connection.
32015	ReportServiceHealthError	In order to monitor the health of your password writeback service, we send heartbeat data to our password-reset web service every five minutes. This event indicates that there was an error when sending this health information back to the cloud web service. This health information doesn't include any personal data, and is purely a heartbeat and basic service statistics so that we can provide service status information in the cloud.
33001	ADUnKnownError	This event indicates that there was an unknown error returned by Active Directory. Check the Microsoft Entra Connect server event log for events from the ADSync source for more information.

Code	Name or message	Description
33002	ADUserNotFoundError	This event indicates that the user who is trying to reset or change a password wasn't found in the on-premises directory. This error can occur when the user is deleted on-premises but not in the cloud. This error can also occur if there's a problem with sync. Check your sync logs and the last few sync run details for more information.
33003	ADMutliMatchError	When a password reset or change request originates from the cloud, we use the cloud anchor specified during the setup process of Microsoft Entra Connect to determine how to link that request back to a user in your on-premises environment. This event indicates that we found two users in your on-premises directory with the same cloud anchor attribute. Check your sync logs and the last few sync run details for more information.
33004	ADPermissionsError	This event indicates that the Active Directory Management Agent (ADMA) service account doesn't have the appropriate permissions on the account in question to set a new password. Ensure that the ADMA account in the user's forest has reset password permissions on all objects in the forest. For more information on how to set the permissions, see Step 4: Set up the appropriate Active Directory permissions. This error could also occur when the user's attribute AdminCount is set to 1.
33005	ADUserAccountDisabled	This event indicates that we attempted to reset or change a password for an account that was disabled on-premises. Enable the account and try the operation again.
33006	ADUserAccountLockedOut	This event indicates that we attempted to reset or change a password for an account that was locked out on-premises. Lockouts can occur when a user tried a change or reset password operation too many times in a short period. Unlock the account and try the operation again.
33007	ADUserIncorrectPassword	This event indicates that the user specified an incorrect current password when performing a password change operation. Specify the correct current password and try again.
33008	ADPasswordPolicyError	This event occurs when the password writeback service attempts to set a password on your local directory that doesn't meet the password age, history, complexity, or filtering requirements of the domain. If you have a minimum password age and have recently changed the password within that window of time, you're

Code	Name or message	Description
		not able to change the password again until it reaches the specified age in your domain. For testing purposes, the minimum age should be set to 0.
		If you have password history requirements enabled, then you must select a password that hasn't been used in the last N times, where N is the password history setting. If you do select a password that's used in the last N times, then you see a failure in this case. For testing purposes, the password history should be set to 0.
		If you have password complexity requirements, all of them are enforced when the user attempts to change or reset a password.
		If you have password filters enabled and a user selects a password that doesn't meet the filtering criteria, then the reset or change operation fails.
33009	ADConfigurationError	This event indicates there was a problem writing a password back to your on-premises directory because of a configuration issue with Active Directory. Check the Microsoft Entra Connect machine's Application event log for messages from the ADSync service for more information on which error occurred.

Organizational unit characters reserved from password writeback

The following table lists reserved characters that prevent password writeback. If these characters appear in your on-premises organizational unit (OU) structure, password writeback may fail with event ID 33001.

[Expand table](#)

Reserved character	Description	Hex value
	space or # character at the beginning of a string	
	space character at the end of a string	
,	comma	0x2C
+	plus sign	0x2B

Reserved character	Description	Hex value
"	quotation mark	0x22
\	backslash	0x5C
<	left angle bracket	0x3C
>	right angle bracket	0x3E
;	semicolon	0x3B
LF	line feed	0x0A
CR	carriage return	0x0D
=	equal sign	0x3D
/	forward slash	0x2F

Microsoft Entra forums

If you have general questions about Microsoft Entra ID and self-service password reset, you can ask the community for assistance on the [Microsoft Q&A question page for Microsoft Entra ID](#). Members of the community include engineers, product managers, MVPs, and fellow IT professionals.

Contact Microsoft support

If you can't find the answer to a problem, our support teams are always available to assist you further.

To properly assist you, we ask that you provide as much detail as possible when opening a case. These details include the following:

- **General description of the error:** What is the error? What was the behavior that was noticed? How can we reproduce the error? Provide as much detail as possible.
- **Page:** What page were you on when you noticed the error? Include the URL if you're able to and a screenshot of the page.
- **Support code:** What was the support code that was generated when the user saw the error?
 - To find this code, reproduce the error, then select the **Support code** link at the bottom of the screen and send the support engineer the GUID that results.

The screenshot shows a Microsoft Online Password Reset page. At the top, there's a navigation bar with a back arrow, forward arrow, refresh button, the URL <https://passwordreset.microsoftonline.com>, and a guest sign-in button. Below the header, the Microsoft logo is displayed. The main content area has a heading "Get back into your account" and a sub-heading "Who are you?". A note below says "To recover your account, begin by entering your user ID and the characters in the picture or audio below." There is a "User ID:" label with an input field, an example text "Example: user@contoso.onmicrosoft.com or user@contoso.com", and a CAPTCHA challenge consisting of a pink box containing the letters "6R SQ" and a speaker icon with a volume dial. Below the CAPTCHA is a text input field with the placeholder "Enter the characters in the picture or the words in the audio." At the bottom, there are "Next" and "Cancel" buttons, and a "Support code" link which is highlighted with a red box.

- If you're on a page without a support code at the bottom, select F12 and search for the SID and CID and send those two results to the support engineer.
- **Date, time, and time zone:** Include the precise date and time *with the time zone* that the error occurred.
- **User ID:** Who was the user who saw the error? An example is *user@contoso.com*.
 - Is this a federated user?
 - Is this a pass-through authentication user?
 - Is this a password-hash-synchronized user?
 - Is this a cloud-only user?
- **Licensing:** Does the user have a Microsoft Entra ID license assigned?
- **Application event log:** If you're using password writeback and the error is in your on-premises infrastructure, include a zipped copy of your Application event log from the Microsoft Entra Connect server.

Next steps

To learn more about SSPR, see [How it works: Microsoft Entra self-service password reset](#) or [How does self-service password reset writeback work in Microsoft Entra ID?](#).

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Self-service password reset frequently asked questions

FAQ

The following are some frequently asked questions (FAQ) for all things related to self-service password reset.

If you have a general question about Microsoft Entra ID and self-service password reset (SSPR) that's not answered here, you can ask the community for assistance. You can do this on the [Microsoft Q&A question page for Microsoft Entra ID](#). Members of the community include engineers, product managers, MVPs, and fellow IT professionals.

This FAQ is split into the following sections:

- Questions about password reset registration
- Questions about password reset
- Questions about password change
- Questions about password management reports
- Questions about password writeback

Password reset registration

Can my users register their own password reset data?

Yes. As long as password reset is enabled and they're licensed, users can go to the password reset registration portal (<https://aka.ms/ssprsetup>) to register their authentication information. Users can also register through the Access Panel (<https://myapps.microsoft.com>). To register through the Access Panel, they need to select their profile picture, select **Profile**, and then select the **Register for password reset** option.

If you enable [combined registration](#), users can register for both SSPR and Microsoft Entra multifactor authentication at the same time.

If I enable password reset for a group and then decide to enable it for everyone are my users

required re-register?

No. Users who populated authentication data aren't required to re-register.

Can I define password reset data on behalf of my users?

Yes, you can do so with Microsoft Entra Connect, PowerShell, the [Microsoft Entra admin center](#), or the [Microsoft 365 admin center](#). For more information, see [Data used by Microsoft Entra self-service password reset](#).

Can I synchronize data for security questions from on-premises?

No, this isn't possible today.

Can my users register data in such a way that other users can't see this data?

Yes. When users register data by using the password reset registration portal, the data is saved into private authentication fields that are visible only to [Privileged Authentication Administrators](#) and the user.

Do my users have to be registered before they can use password reset?

No. If you define enough authentication information on their behalf, users don't have to register. Password reset works as long as you properly format the data stored in the appropriate fields in the directory.

Can I synchronize or set the authentication phone, authentication email, or alternate authentication phone fields on behalf of my users?

The fields that are able to be set by a [Privileged Authentication Administrator](#) are defined in the article [SSPR Data requirements](#).

How does the registration portal determine which options to show my users?

The password reset registration portal shows only the options that you enabled for your users. These options are found under the **User Password Reset Policy** section of your directory's **Configure** tab. For example, if you don't enable security questions, then users aren't able to register for that option.

When is a user considered registered?

A user is considered registered for SSPR when they registered at least the **Number of methods required to reset** a password that you set in the [Microsoft Entra admin center](#).

Password reset

Do you prevent users from multiple attempts to reset a password in a short period of time?

Yes, there are security features built into password reset to protect it from misuse.

Users can attempt to validate their information (such as their phone number), but if they're unable to prove their identity five times within a 24-hour period, they're locked out for 24 hours.

Users can try to validate a phone number, auth app, send a text message, or validate security questions and answers only five times within an hour before they're locked out for 24 hours.

Users can send an email a maximum of 10 times within a 10-minute period before they're locked out for 24 hours.

The counters are reset once a user resets their password.

How long should I wait to receive an email, text message, or phone call from password reset?

Emails, text messages, and phone calls should arrive in under a minute. The normal case is 5 to 20 seconds. If you don't receive the notification in this time frame:

- Check your junk folder.
- Check that the number or email being contacted is the one you expect.
- Check that the authentication data in the directory is correctly formatted, for example, +1 4255551234 or *user@contoso.com*.

What languages are supported by password reset?

The password reset UI, text messages, and voice calls are localized in the same languages that are supported in Microsoft 365.

What parts of the password reset experience get branded when I set the organizational branding items in my directory's configure tab?

The password reset portal shows your organization's logo and allows you to configure the "Contact your administrator" link to point to a custom email or URL. Any email that's sent by password reset includes your organization's logo, colors, and name in the body of the email, and is customized from the settings for that particular name.

How can I educate my users about where to go to reset their passwords?

Try some of the suggestions in our [SSPR deployment](#) article.

Can I use this page from a mobile device?

Yes, this page works on mobile devices.

Do you support unlocking local Active Directory accounts when users reset their passwords?

Yes. When a user resets their password, if password writeback is deployed through Microsoft Entra Connect, that user's account is automatically unlocked when they reset their password.

How can I integrate password reset directly into my user's desktop sign-in experience?

If you're a Microsoft Entra ID P1 or P2 customer, you can install Microsoft Identity Manager at no additional cost and deploy the on-premises password reset solution.

Can I set different security questions for different locales?

No, this isn't possible today.

How many questions can I configure for the security questions authentication option?

You can configure up to 20 custom security questions in the [Microsoft Entra admin center](#).

How long can security questions be?

Security questions can be 3 to 200 characters long.

How long can the answers to security questions be?

Answers can be 3 to 40 characters long.

Are duplicate answers to security questions rejected?

Yes, we reject duplicate answers to security questions.

Can a user register the same security question more than once?

No. After a user registers a particular question, they can't register for that question a second time.

Is it possible to set a minimum limit of security questions for registration and reset?

Yes, one limit can be set for registration and another for reset. Three to five security questions can be required for registration, and three to five questions can be required for reset.

I configured my policy to require users to use security questions for reset, but the Azure administrators seem to be configured differently.**

This is the expected behavior. Microsoft enforces a strong default two-gate password reset policy for any Azure administrator role. This prevents administrators from using security questions. You can find more information about this policy in the [Password policies and restrictions in Microsoft Entra ID](#) article.

If a user has registered more than the maximum number of questions required to reset, how are the security questions selected during reset?

N number of security questions are selected at random out of the total number of questions a user has registered for, where N is the amount that is set for the **Number of questions required to reset** option. For example, if a user has registered five security questions, but only three are required to reset a password, three of the five questions are randomly selected and are presented at reset. To prevent question hammering, if the user gets the answers to the questions wrong the selection process starts over.

How long are the email and text message one-time passcodes valid?

The session lifetime for password reset is 15 minutes. From the start of the password reset operation, the user has 15 minutes to reset their password. The one-time passcodes are valid for 5 minutes during the password reset session.

Can I block users from resetting their password?

Yes, if you use a group to enable SSPR, you can remove an individual user from the group that allows users to reset their password. If the user is a [Privileged Authentication Administrator](#), they can reset their password and this can't be disabled.

Password change

Where should my users go to change their passwords?

Users can change their passwords anywhere they see their profile picture or icon, like in the upper-right corner of their [Office 365](#) portal or [Access Panel](#) experiences. Users can change their passwords from the [Access Panel Profile page](#). Users can also be asked to change their passwords automatically at the Microsoft Entra sign-in page if their passwords are expired. Finally, users can browse to the [Microsoft Entra password change portal](#) directly if they want to change their passwords.

Can my users be notified in the Office portal when their on-premises password expires?

Yes, this is possible today if you use Active Directory Federation Services (AD FS). If you use AD FS, follow the instructions in the [Sending password policy claims with AD FS](#) article. If you use password hash synchronization, this isn't possible today. We don't sync password policies from on-premises directories, so it's not possible for us to post expiration notifications to cloud experiences. In either case, it's also possible to [notify users whose passwords are about to expire through PowerShell](#).

Can I block users from changing their password?

For cloud-only users, password changes can't be blocked. For on-premises users, you can set the **User can't change password** option to selected. The selected users can't change their password.

Password management reports

How long does it take for data to show up on the password management reports?

Data should appear on the password management reports in 5 to 10 minutes. In some instances, it might take up to an hour to appear.

How can I filter the password management reports?

To filter the password management reports, select the small magnifying glass to the extreme right of the column labels, near the top of the report. For more comprehensive filtering, you can download the report to Excel and create a pivot table.

What is the maximum number of events that are stored in the password management reports?

Up to 75,000 password reset or password reset registration events are stored in the password management reports, spanning back as far as 30 days. We're working to expand this number to include more events.

How far back do the password management reports go?

The password management reports show operations that occurred within the last 30 days. For now, if you need to archive this data, you can download the reports periodically and save them in a separate location.

Is there a maximum number of rows that can appear on the password management reports?

Yes. A maximum of 75,000 rows can appear on either of the password management reports, whether they're shown in the UI or are downloaded.

Is there an API to access the password reset or registration reporting data?

Yes, you can get this info from the [Authentication Methods Activity report](#) or the [API to get password reset activity](#). You can also use the [audit logs API](#) and filter by SSPR events.

Password writeback

How does password writeback work behind the scenes?

See the article [How password writeback works](#) for an explanation of what happens when you enable password writeback and how data flows through the system back into your on-premises environment.

How long does password writeback take to work? Is there a synchronization delay like there is with password hash sync?

Password writeback is instant. It's a synchronous pipeline that works differently than password hash synchronization. Password writeback allows users to get real-time feedback about the success of their password reset or change operation. The average time for a successful writeback of a password is under 500 ms.

If my on-premises account is disabled, how is my cloud account and access affected?

If your on-premises ID is disabled, your cloud ID and access will also be disabled at the next sync interval through Microsoft Entra Connect. By default, this sync is every 30 minutes.

If my on-premises account is constrained by an on-premises Active Directory password policy, does SSPR obey this policy when I change my password?

Yes, SSPR relies on and abides by the on-premises Active Directory password policy. This policy includes the typical Active Directory domain password policy, and any defined, fine-grained password policies that are targeted to a user.

What types of accounts does password writeback work for?

Password writeback works for user accounts that are synchronized from on-premises Active Directory to Microsoft Entra ID, including federated, password hash synchronized, and Pass-Through Authentication Users.

Does password writeback enforce my domain's password policies?

Yes. Password writeback enforces password age, history, complexity, filters, and any other restriction you might put in place on passwords in your local domain.

Is password writeback secure? How can I be sure I won't get hacked?

Yes, password writeback is secure. To read more about the multiple layers of security implemented by the password writeback service, check out the [Password writeback security](#) section in the [Password writeback overview](#) article.

Next steps

- [How do I complete a successful rollout of SSPR?](#)

- [Reset or change your password ↗](#)
 - [Register for self-service password reset ↗](#)
 - [Do you have a licensing question?](#)
 - [What data is used by SSPR and what data should you populate for your users?](#)
 - [What authentication methods are available to users?](#)
 - [What are the policy options with SSPR?](#)
 - [What is password writeback and why do I care about it?](#)
 - [How do I report on activity in SSPR?](#)
 - [What are all of the options in SSPR and what do they mean?](#)
 - [I think something is broken. How do I troubleshoot SSPR?](#)
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Frequently asked questions about Microsoft Entra multifactor authentication

FAQ

This FAQ answers common questions about Microsoft Entra multifactor authentication and using the multifactor authentication service. It's broken down into questions about the service in general, billing models, user experiences, and troubleshooting.

Important

In September 2022, Microsoft announced deprecation of Multifactor Authentication Server. Beginning September 30, 2024, Multifactor Authentication Server deployments will no longer service multifactor authentication requests, which could cause authentications to fail for your organization. To ensure uninterrupted authentication services and to remain in a supported state, organizations should [migrate their users' authentication data](#) to the cloud-based Microsoft Entra multifactor authentication service by using the latest Migration Utility included in the most recent [MFA Server update](#). For more information, see [MFA Server Migration](#).

General

How does Azure Multifactor Authentication Server handle user data?

With Multifactor Authentication Server, user data is only stored on the on-premises servers. No persistent user data is stored in the cloud. When the user performs two-step verification, Multifactor Authentication Server sends data to the Microsoft Entra multifactor authentication cloud service for authentication. Communication between Multifactor Authentication Server and the multifactor authentication cloud service uses Secure Sockets Layer (SSL) or Transport Layer Security (TLS) over port 443 outbound.

When authentication requests are sent to the cloud service, data is collected for authentication and usage reports. The following data fields are included in two-step verification logs:

- **Unique ID** (either user name or on-premises Multifactor Authentication Server ID)
- **First and Last Name** (optional)
- **Email Address** (optional)
- **Phone Number** (when using a voice call or text message authentication)
- **Device Token** (when using mobile app authentication)
- **Authentication Mode**
- **Authentication Result**
- **Multifactor Authentication Server Name**
- **Multifactor Authentication Server IP**
- **Client IP** (if available)

The optional fields can be configured in Multifactor Authentication Server.

The verification result (success or denial), and the reason if it was denied, is stored with the authentication data. This data is available in authentication and usage reports.

For more information, see [Data residency and customer data for Microsoft Entra multifactor authentication](#).

What short codes are used for sending text messages to my users?

In the United States, we use the following short codes:

- 97671
- 69829
- 51789
- 99399

In Canada, we use the following short codes:

- 759731
- 673801

There's no guarantee of consistent text message or voice-based multifactor authentication prompt delivery by the same number. In the interest of our users, we may add or remove short codes at any time as we make route adjustments to improve text message deliverability.

We don't support short codes for countries or regions besides the United States and Canada.

Does Microsoft Entra multifactor authentication throttle user sign-ins?

Yes, in certain cases that typically involve repeated authentication requests in a short time window, Microsoft Entra multifactor authentication throttles user sign-in attempts to protect telecommunication networks, mitigate MFA fatigue-style attacks and protect its own systems for the benefit of all customers.

Although we don't share specific throttling limits, they're based around reasonable usage.

Is my organization charged for sending the phone calls and text messages that are used for authentication?

No, you're not charged for individual phone calls placed or text messages sent to users through Microsoft Entra multifactor authentication. If you use a per-authentication MFA provider, you're billed for each authentication, but not for the method used.

Your users might be charged for the phone calls or text messages they receive, according to their personal phone service.

Does the per-user billing model charge me for all enabled users, or just the ones that performed two-step verification?

Billing is based on the number of users configured to use multifactor authentication, regardless of whether they performed two-step verification that month.

How does multifactor authentication billing work?

When you create a per-user or per-authentication MFA provider, your organization's Azure subscription is billed monthly based on usage. This billing model is similar to how Azure bills for usage of virtual machines and Web Apps.

When you purchase a subscription for Microsoft Entra multifactor authentication, your organization only pays the annual license fee for each user. MFA licenses and Microsoft

365, Microsoft Entra ID P1 or P2, or Enterprise Mobility + Security bundles are billed this way.

For more information, see [How to get Microsoft Entra multifactor authentication](#).

Is there a free version of Microsoft Entra multifactor authentication?

Security defaults can be enabled in the Microsoft Entra ID Free tier. With security defaults, all users are enabled for multifactor authentication using the Microsoft Authenticator app. There's no ability to use text message or phone verification with security defaults, just the Microsoft Authenticator app.

For more information, see [What are security defaults?](#)

Can my organization switch between per-user and per-authentication consumption billing models at any time?

If your organization purchases MFA as a standalone service with consumption-based billing, you choose a billing model when you create an MFA provider. You can't change the billing model after an MFA provider is created.

If your MFA provider is *not* linked to a Microsoft Entra tenant, or you link the new MFA provider to a different Microsoft Entra tenant, user settings, and configuration options aren't transferred. Also, existing MFA Servers need to be reactivated using activation credentials generated through the new MFA Provider. Reactivating the MFA Servers to link them to the new MFA Provider doesn't impact phone call and text message authentication, but mobile app notifications stop working for all users until they reactivate the mobile app.

Learn more about MFA providers in [Getting started with an Azure multifactor authentication provider](#).

Can my organization switch between consumption-based billing and subscriptions (a license-based model) at any time?

In some instances, yes.

If your directory has a *per-user* Microsoft Entra multifactor authentication provider, you can add MFA licenses. Users with licenses aren't counted in the per-user consumption-based billing. Users without licenses can still be enabled for MFA through the MFA provider. If you purchase and assign licenses for all your users configured to use multifactor authentication, you can delete the Microsoft Entra multifactor authentication provider. You can always create another per-user MFA provider if you have more users than licenses in the future.

If your directory has a *per-authentication* Microsoft Entra multifactor authentication provider, you're always billed for each authentication, as long as the MFA provider is linked to your subscription. You can assign MFA licenses to users, but you'll still be billed for every two-step verification request, whether it comes from someone with an MFA license assigned or not.

Does my organization have to use and synchronize identities to use Microsoft Entra multifactor authentication?

If your organization uses a consumption-based billing model, Microsoft Entra ID is optional, but not required. If your MFA provider isn't linked to a Microsoft Entra tenant, you can only deploy Azure Multifactor Authentication Server on-premises.

Microsoft Entra ID is required for the license model because licenses are added to the Microsoft Entra tenant when you purchase and assign them to users in the directory.

Manage and support user accounts

What should I tell my users to do if they don't receive a response on their phone?

Have your users attempt up to five times in 5 minutes to get a phone call or text message for authentication. Microsoft uses multiple providers for delivering calls and text messages. If this approach doesn't work, open a support case to troubleshoot further.

Third-party security apps may also block the verification code text message or phone call. If using a third-party security app, try disabling the protection, then request another MFA verification code be sent.

If the prior steps don't work, check if users are configured for more than one verification method. Try signing in again, but select a different verification method on the sign-in page.

For more information, see the [end-user troubleshooting guide](#).

What should I do if one of my users can't get in to their account?

You can reset the user's account by making them go through the registration process again. Learn more about [managing user and device settings with Microsoft Entra multifactor authentication in the cloud](#).

What should I do if one of my users loses a phone that is using app passwords?

To prevent unauthorized access, delete all the user's app passwords. After the user has a replacement device, they can recreate the passwords. Learn more about [managing user and device settings with Microsoft Entra multifactor authentication in the cloud](#).

What if a user can't sign in to nonbrowser apps?

If your organization still uses legacy clients, and you [allowed the use of app passwords](#), then your users can't sign in to these legacy clients with their username and password. Instead, they need to [set up app passwords](#). Your users must clear (delete) their sign-in information, restart the app, and then sign in with their username and *app password* instead of their regular password.

If your organization doesn't have legacy clients, you shouldn't allow your users to create app passwords.

Note

Modern authentication for Office 2013 clients

App passwords are only necessary for apps that don't support modern authentication. Office 2013 clients support modern authentication protocols, but need to be configured. Modern authentication is available to any customer running the March 2015 or later update for Office 2013. For more information, see the blog post [Updated Office 365 modern authentication](#).

My users say that sometimes they don't receive the text message or the verification times out.

Delivery of text messages isn't guaranteed because uncontrollable factors might affect the reliability of the service. These factors include the destination country or region, the mobile phone carrier, and the signal strength.

Third-party security apps may also block the verification code text message or phone call. If using a third-party security app, try disabling the protection, then request another MFA verification code be sent.

If your users often have problems with reliably receiving text messages, tell them to use the Microsoft Authenticator app or phone call method instead. The Microsoft Authenticator can receive notifications both over cellular and Wi-Fi connections. In addition, the mobile app can generate verification codes even when the device has no signal at all. The Microsoft Authenticator app is available for [Android ↗](#), [iOS ↗](#), and [Windows Phone ↗](#).

Can I change the amount of time my users have to enter the verification code from a text message before the system times out?

In some cases, yes.

For one-way SMS with MFA Server v7.0 or higher, you can configure the timeout setting by setting a registry key. After the MFA cloud service sends the text message, the verification code (or one-time passcode) is returned to the MFA Server. The MFA Server stores the code in memory for 300 seconds by default. If the user doesn't enter the code before the 300 seconds have passed, their authentication is denied. Use these steps to change the default timeout setting:

1. Go to `HKLM\Software\Wow6432Node\Positive Networks\PhoneFactor`.
2. Create a **DWORD** registry key called `pfsvc_pendingSmsTimeoutSeconds` and set the time in seconds that you want the MFA Server to store one-time passcodes.

💡 Tip

If you have multiple MFA Servers, only the one that processed the original authentication request knows the verification code that was sent to the user. When the user enters the code, the authentication request to validate it must be sent to

the same server. If the code validation is sent to a different server, the authentication is denied.

If users don't respond to the SMS within the defined timeout period, their authentication is denied.

For one-way SMS with Microsoft Entra multifactor authentication in the cloud (including the AD FS adapter or the Network Policy Server extension), you can't configure the timeout setting. Microsoft Entra ID stores the verification code for 180 seconds.

Can I use hardware tokens with Multifactor Authentication Server?

If you're using Multifactor Authentication Server, you can import third-party Open Authentication (OATH) time-based, one-time password (TOTP) tokens, and then use them for two-step verification.

You can use ActiveIdentity tokens that are OATH TOTP tokens if you put the secret key in a CSV file and import to Multifactor Authentication Server. You can use OATH tokens with Active Directory Federation Services (ADFS), Internet Information Server (IIS) forms-based authentication, and Remote Authentication Dial-In User Service (RADIUS) as long as the client system can accept the user input.

You can import third-party OATH TOTP tokens with the following formats:

- Portable Symmetric Key Container (PSKC)
- CSV if the file contains a serial number, a secret key in Base 32 format, and a time interval

Can I use Multifactor Authentication Server to secure Terminal Services?

Yes, but if you're using Windows Server 2012 R2 or later, you can only secure Terminal Services by using Remote Desktop Gateway (RD Gateway).

Security changes in Windows Server 2012 R2 changed how Multifactor Authentication Server connects to the Local Security Authority (LSA) security package in Windows Server 2012 and earlier versions. For versions of Terminal Services in Windows Server 2012 or earlier, you can [secure an application with Windows Authentication](#). If you're using Windows Server 2012 R2, you need RD Gateway.

I configured Caller ID in MFA Server, but my users still receive multifactor authentication calls from an anonymous caller.

When multifactor authentication calls are placed through the public telephone network, sometimes they're routed through a carrier that doesn't support caller ID. Because of this carrier behavior, caller ID isn't guaranteed, even though the multifactor authentication system always sends it.

Why are my users being prompted to register their security information?

There are several reasons that users could be prompted to register their security information:

- The user has been enabled for MFA by their administrator in Microsoft Entra ID, but doesn't have security information registered for their account yet.
- The user has been enabled for self-service password reset in Microsoft Entra ID. The security information will help them reset their password in the future if they ever forget it.
- The user accessed an application that has a Conditional Access policy to require MFA and hasn't previously registered for MFA.
- The user is registering a device with Microsoft Entra ID (including Microsoft Entra join), and your organization requires MFA for device registration, but the user hasn't previously registered for MFA.
- The user is generating Windows Hello for Business in Windows 10 (which requires MFA) and hasn't previously registered for MFA.
- The organization has created and enabled an MFA Registration policy that has been applied to the user.
- The user previously registered for MFA, but chose a verification method that an administrator has since disabled. The user must therefore go through MFA registration again to select a new default verification method.

Errors

What should users do if they see an "Authentication request isn't for an activated

account" error message when using mobile app notifications?

Ask the user to complete the following procedure to remove their account from the Microsoft Authenticator, then add it again:

1. Go to [their account profile](#) and sign in with an organizational account.
2. Select **Additional Security Verification**.
3. Remove the existing account from the Microsoft Authenticator app.
4. Select **Configure**, and then follow the instructions to reconfigure the Microsoft Authenticator.

What should users do if they see a 0x800434D4L error message when signing in to a nonbrowser application?

The *0x800434D4L* error occurs when you try to sign in to a nonbrowser application, installed on a local computer, that doesn't work with accounts that require two-step verification.

A workaround for this error is to have separate user accounts for admin-related and nonadmin operations. Later, you can link mailboxes between your admin account and nonadmin account so that you can sign in to Outlook by using your nonadmin account. For more details about this solution, learn how to [give an administrator the ability to open and view the contents of a user's mailbox](#).

What are the possible reasons why a user fails, with the error code "LsaLogonUser failed with NTSTATUS -1073741715 for MFA Server"?

Error 1073741715 = Status Logon Failure -> The attempted logon is invalid. This is due to either a bad username or authentication.

A plausible reason for this error: If the primary credentials entered are correct, there might be a mismatch between the supported NTLM version on the MFA server and the domain controller. MFA Server supports only NTLMv1 (LmCompatibilityLevel=1 through 4) and not NTLMv2 (LmCompatibilityLevel=5).

Next steps

If your question isn't answered here, the following support options are available:

- Search the [Microsoft Support Knowledge Base](#) for solutions to common technical issues.
 - Search for and browse technical questions and answers from the community, or ask your own question in the [Microsoft Entra Q&A](#).
 - Contact Microsoft professional through [Multifactor Authentication Server support](#). When contacting us, it's helpful if you can include as much information about your issue as possible. Information you can supply includes the page where you saw the error, the specific error code, the specific session ID, and the ID of the user who saw the error.
-

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#)

Azure PowerShell documentation

Official product documentation for Azure PowerShell. Azure PowerShell is a collection of modules for managing Azure resources from PowerShell.

About Azure PowerShell

OVERVIEW

[Get started](#)

[What is Azure PowerShell?](#)

[Support lifecycle](#)

TRAINING

[Automate Azure tasks from PowerShell](#)

[Choose the best Azure command line tools](#)

Installation

DOWNLOAD

[Overview](#)

[Install - Windows](#)

[Install - Linux](#)

[Install - macOS](#)

[Run in Azure Cloud Shell ↗](#)

[Run in a Docker container](#)

What's new

WHAT'S NEW

[Overview](#)

[Release notes](#)

[Az 14.0.0 migration guide](#)

[Upcoming breaking changes](#)

[Impact of MFA in automation](#)

Azure PowerShell reference



[Cmdlet reference](#)

Identity and authentication



[Authentication methods](#)

[Create a service principal](#)

[Credential contexts](#)

Concepts



[Manage subscriptions](#)

[Manage Azure resources with Invoke-AzRestMethod](#)

[Filter cmdlet results](#)

[Format output](#)

[PowerShell jobs](#)



[Create virtual machines](#)

Configuration

HOW-TO GUIDE

[Configure global settings](#)

[Intelligent command completion](#)

[Use the Az PowerShell module behind a proxy](#)

Deploy

DEPLOY

[Deploy resource manager templates](#)

[Export resource manager templates](#)

[Deploy private resource manager templates](#)

Samples

SAMPLE

[Azure App Service](#)

[SQL databases](#)

[Cosmos DB](#)

[Samples repo](#)

Migrate from AzureRM

OVERVIEW

[Introducing the Az PowerShell module](#)

[Changes between AzureRM and Az](#)

HOW-TO GUIDE

Migration steps

QUICKSTART

[Automatically migrate PowerShell scripts](#)

Help & support

OVERVIEW

[Report product issues ↗](#)

[Troubleshoot](#)

[Get help from the community ↗](#)

[Follow Azure PowerShell on X ↗](#)

[Azure Tools Blog ↗](#)

Microsoft Entra authentication methods API overview

Article • 04/22/2025

Namespace: microsoft.graph

[Authentication methods](#) are the ways that users authenticate in Microsoft Entra ID.

Authentication methods in Microsoft Entra ID include password and phone (for example, SMS and voice calls), which are manageable in Microsoft Graph beta endpoint today, among many others such as FIDO2 security keys and the Microsoft Authenticator app. Authentication methods are used in primary, second-factor, and step-up authentication, and also in the self-service password reset (SSPR) process.

The authentication method APIs are used to manage a user's authentication methods. For example:

- You can add a phone number to a user. The user can then use that phone number for SMS and voice call authentication if they're enabled to use it by policy.
 - You can update that number, or delete it from the user.
 - You can enable or disable the number for SMS sign-in.
- You can retrieve details of a user's FIDO2 Security Key, and delete it if the user has lost the key.
- You can retrieve details of a user's Microsoft Authenticator registration, and delete it if the user has lost the phone.
- You can retrieve details of a user's Windows Hello for Business registration, and delete it if the user has lost the device.
- You can add an email address to a user. The user can then use that email as part of the Self-Service Password Reset (SSPR) process.
 - You can update that email, or delete it from the user.

The ability for a user to use an authentication method is governed by the [authentication method policy](#) for the tenant. For example, only users in the R&D department might be enabled to use the FIDO2 method while all users might be enabled to use Microsoft Authenticator.

We don't recommend using the authentication methods APIs for scenarios where you need to iterate over your entire user population for auditing or security check purposes. For these types of scenarios, we recommend using the [authentication method registration and usage reporting APIs](#) (available on the `beta` endpoint only).

 Note

Requests to the authentication methods APIs time-out after 60 seconds.

What authentication methods can be managed in Microsoft Graph?

 Expand table

Authentication method	Description	Examples
emailAuthenticationMethod	A user can use an email address as part of the Self-Service Password Reset (SSPR) process.	See a user's authentication email address. Add, update, or remove an email address to a user.
fido2AuthenticationMethod	A user can use a FIDO2 security key to sign-in to Microsoft Entra ID.	Delete a lost FIDO2 security key.
microsoftAuthenticatorAuthenticationMethod	A user can use Microsoft Authenticator app to sign-in or perform multi-factor authentication to Microsoft Entra ID	Delete a Microsoft Authenticator authentication method.
passwordAuthenticationMethod	A password is currently the default primary authentication method in Microsoft Entra ID.	Reset a user's password
phoneAuthenticationMethod	A user can use a phone to authenticate using SMS or voice calls as allowed by policy.	See a user's authentication phone numbers. Add, update, or remove a phone number for a user. Enable or disable a primary mobile phone for SMS sign-in.
platformCredentialAuthenticationMethod	A user can use a platform credential to sign in to Microsoft Entra ID.	Delete a lost platform credential authentication method.
softwareOathAuthenticationMethod	Allow users to perform multifactor authentication using an application that supports the OATH TOTP	Get and delete a software OATH token assigned to a user.

Authentication method	Description	Examples
	specification and provides a one-time code.	
temporaryAccessPassAuthenticationMethod	A time-limited passcode that serves as a strong credential and allows onboarding of passwordless credentials.	Create and manage a customized time-limited passcode for a given user to use for strong authentication or recovery.
windowsHelloForBusinessAuthenticationMethod	Windows Hello for Business is a passwordless sign-in method on Windows devices.	See devices where a user has enabled Windows Hello for Business sign-in. Delete a Windows Hello for Business credential.

The following authentication methods are not yet supported in Microsoft Graph v1.0.

[Expand table](#)

Authentication method	Description	Examples
Default method	Represents the method the user has selected as default for performing multi-factor authentication.	Change a user's default MFA method.
Hardware token	Allow users to perform multifactor authentication using a physical device that provides a one-time code.	Get a hardware token assigned to a user.
Security questions and answers	Allow users to validate their identity when performing a self-service password reset.	Delete a security question a user registered.
Authentication states	Manage a user's sign-in preferences and per-user MFA	See or set the MFA state for a user. See or set the system-preferred multifactor authentication (MFA) setting.

Require re-register multifactor authentication

To require users to set up a new multifactor authentication the next time they sign in, call the individual DELETE authentication method operations to delete each of the user's current authentication methods. After the user has no more methods, they're prompted to register the next time they sign in where strong authentication is required.

Tenant-level authentication method usage

You can monitor tenant-level authentication method registration and usage, including users registered or unregistered for MFA and passwordless authentication, and users registered or unregistered for SSPR by using the [Authentication methods usage report APIs](#).

Next steps

- Review the authentication method types and their various methods.
- Try the API in [Graph Explorer](#).

Microsoft Entra authentication strengths API overview

Article • 04/21/2023

Namespace: microsoft.graph

Authentication strengths allow administrators to require specific combinations of Microsoft Entra [authentication methods](#) to access a resource. Each authentication strength comprises one or more combinations of authentication methods, where each combination is one or more authentication methods. When the strength is applied to a scenario as a [grant control](#) in [Conditional Access](#), a user in scope of the policy is required to satisfy one of those allowed combinations at sign-in before they can access the resource. As part of Conditional Access, authentication strengths can also be paired with other Conditional Access controls such as user risk and location.

For example, an administrator can require users to authenticate using phishing-resistant authentication methods before they can access a sensitive resource. The administrator can also allow users to authenticate using less-secure multifactor authentication (MFA) combinations, such as password and SMS, for them to access non-sensitive applications.

This article introduces the Microsoft Graph APIs that allow administrators to programmatically manage authentication strengths.

Authentication strength policies

Authentication strength policies define the authentication strengths that are available for use in the tenant. Use the [authenticationStrengthPolicy](#) resource type and its associated methods to define and manage these policies. The policies include the following configurations:

- The name, identifier, and description of the policy.
- Authentication method combinations that are part of the policy.
- Whether the policy, when the authentication method requirements are satisfied, can be used to satisfy an MFA claim in the access token.

Microsoft Entra ID supports both built-in and custom authentication strength policies.

Microsoft has supplied the following three built-in policies:

- Multifactor authentication
- Passwordless multifactor authentication
- Phishing resistant multifactor authentication

You can only read built-in policies, but you can create up to 15 custom policies to suit your requirements.

Authentication method combinations

Core to a policy are the authentication method combinations. A combination consists of one or more authentication methods in a comma-separated list. Combinations are pre-defined and are used to define an authentication strength. These authentication methods are based the **authenticationMethodModes** flagged enumeration. Some example combinations include:

 Expand table

Example allowed combination	Description
fido2	The user must sign in using a FIDO2 security key to satisfy the authentication strength requirement.
password,microsoftAuthenticatorPush	The user must sign in using <i>both</i> password and Microsoft Authenticator push approval to satisfy the authentication strength requirement.
password,softwareOauth	The user must sign in using <i>both</i> password and software OATH token to satisfy the authentication strength requirement.

Microsoft Entra ID provides the predefined, read-only combinations using the following principles:

- Single factor authentication methods that can be used as first factors such as password and SMS.
- Combinations of password and a second factor that make a valid multifactor authentication combination ("something you have" and "something you know").
- Passwordless multifactor authenticators such as FIDO2 and x509 certificate authentication.

Built-in authentication strengths use these combinations, and combinations can be used in custom authentication strengths.

To view the details of the supported authentication methods and the allowed combinations, call the [List authenticationMethodModes](#) API.

The authentication combinations of built-in policies are read-only. To see all built-in policies and their configurations, call the [List authenticationStrengthPolicies](#) API.

To create a custom authentication strength policy, you must configure the authentication method combinations using the allowed combinations.

Combination configurations

You can apply further restrictions on certain authentication methods to control which instances of the method a user can use to authenticate. These kinds of restrictions are [combination configurations](#) and can also be part of an [authenticationStrengthPolicy](#) object.

A combination configuration may apply to one or more combinations that include the specific authentication method. Today, [FIDO2](#) is the only method that supports combination configurations.

For example, a custom policy allows the following combinations: `password`, `softwareOauth`, `fido2`, and `x509CertificateMultiFactor`. For this policy, you can restrict the FIDO2 security keys that the user can use to authenticate by configuring a combination configuration with specific Authenticator Attestation GUIDs (AAGUIDs).

An authentication strength policy has zero or more combination configurations.

Apply authentication strength policies in Conditional Access

After defining the authentication strength policy, you apply and enforce it for the protected resource using Microsoft Entra [conditional access policies](#).

In the Conditional Access [grant controls](#), configure the **authenticationStrength** relationship by assigning the [authenticationStrengthPolicy](#) object that should be associated with the conditional access policy. When a conditional access policy applies to a sign-in and that policy has an authentication strength grant control, the user will be required to use one of the allowed authentication method combinations to sign in. Authentication strength policies can also be enforced for guest users through both conditional access policies and [cross-tenant access inbound trust settings](#).

The **authenticationStrength** object corresponds to the 'Require authentication strength' control of the Conditional Access policy's UX on the Microsoft Entra admin center.

You can't configure authentication strengths and the multifactor authentication grant control on the same conditional access policy.

Next steps

- [Read more ↗](#) about authentication strengths.
- Try the API in [Graph Explorer](#).

Microsoft Entra authentication methods policies API overview

Article • 02/21/2025

Namespace: microsoft.graph

Authentication methods policies define [authentication methods](#) and the users that are allowed to use them to sign in and perform multifactor authentication (MFA) in Microsoft Entra ID. Authentication methods policies that can be managed in Microsoft Graph include FIDO2 Security Keys and Passwordless Phone Sign-in with Microsoft Authenticator app.

The authentication method policies APIs are used to manage policy settings. For example:

- Define the types of FIDO2 security keys that can be used in the Microsoft Entra tenant.
- Define the users or groups of users who are allowed to use FIDO2 Security Keys or Passwordless Phone Sign-in to sign in to Microsoft Entra ID.
- Define the users or groups of users who should be reminded to set up the Microsoft Authenticator for MFA using push notifications.

 Note

Requests to the authentication methods policies APIs time-out after 60 seconds.

What authentication methods policies can be managed in Microsoft Graph?

 Expand table

Authentication method policy	Description
emailauthenticationmethodconfiguration	Define users who can use email OTP on the Microsoft Entra tenant.
fido2authenticationmethodconfiguration	Define FIDO2 security key restrictions and users who can use them to sign in to Microsoft Entra ID.
microsoftauthenticatorauthenticationmethodconfiguration	Define users who can use Microsoft Authenticator on the Microsoft Entra tenant.
smsAuthenticationMethodConfiguration	Defines users who can use Text Message on the Microsoft Entra tenant.

Authentication method policy	Description
softwareOathAuthenticationMethodConfiguration	Defines users who can use a third-party software OATH authentication method.
temporaryaccesspassauthenticationmethodconfiguration	Defines users who can use Temporary Access Pass to sign in to Microsoft Entra ID.
voiceAuthenticationMethodConfiguration	Defines users or groups that are enabled to use the voice call authentication method.
x509CertificateAuthenticationMethodConfiguration	Defines users who can use X.509 certificate to sign in to Microsoft Entra ID.

Policies available for authentication methods registration campaign

[\[+\] Expand table](#)

Policy	Description
authenticationMethodsRegistrationCampaign	Define users who should be reminded to set up an authentication method (currently only supported for the Microsoft Authenticator).

Next steps

- Try the API in the [Graph Explorer](#).

Working with the authentication methods usage report API

Article • 12/31/2024

Namespace: microsoft.graph

Authentication methods activity reports provides information on the registration and usage of [authentication methods](#) in your tenant.

These reports provide information such as:

- How many users are registered for each authentication method
- How many users are registered for features such as multifactor authentication (MFA), Self-Service Password Reset (SSPR), and passwordless authentication.
- The failure rates of each authentication method

These reports are available on the Microsoft Entra portal through **Protection** tab group > **Authentication methods** tab > **Activity** tab under the *Monitoring* tab group.

Licenses

A Microsoft Entra ID P1 or P2 license is required to access authentication methods usage and insights reports. Microsoft Entra multifactor authentication and self-service password reset (SSPR) licensing information can be found on the [Microsoft Entra pricing site](#).

Available reports

The following reports are available through Microsoft Graph:

- Per-user report of the status of their authentication methods including the default methods, whether registered for MFA, SSPR, and a passwordless authentication method, and so on. For more information, see the [userRegistrationDetails resource type](#).
- Count of users registered, enabled, and capable of using MFA, SSPR, and passwordless authentication. For more information, see the [usersRegisteredByFeature resource type](#).
- Raw count of users registered for email, password, and phone authentication methods. For more information, see the [usersRegisteredByMethod resource type](#).

The following reports are available on the `beta` endpoint only:

- Users registered and capable of self-service password reset (SSPR) and Azure multifactor authentication (MFA). For more information, see the [credentialUserRegistrationCount resource type](#).

- SSPR usage activity. For more information, see the [userCredentialUsageDetails resource type](#).
- Tenant-level summary of user SSPR activity, including failure and successes. For more information, see the [credentialUsageSummary resource type](#).

Related content

- [Microsoft Entra authentication methods activity](#)

Microsoft Entra service limits and restrictions

Article • 01/31/2025

This article contains the usage constraints and other service limits for the Microsoft Entra ID, part of Microsoft Entra, service. If you're looking for the full set of Microsoft Azure service limits, see [Azure Subscription and Service Limits, Quotas, and Constraints](#).

Here are the usage constraints and other service limits for the Microsoft Entra service.

[+] [Expand table](#)

Category	Limit
Tenants	<ul style="list-style-type: none">A single user can belong to a maximum of 500 Microsoft Entra tenants as a member or a guest.Create a maximum of 200 tenants.Limit of 300 license-based subscriptions (such as Microsoft 365 subscriptions) per tenant
Domains	<ul style="list-style-type: none">You can add no more than 5,000 managed domain names.If you set up all of your domains for federation with on-premises Active Directory, you can add no more than 2,500 domain names in each tenant.
Resources	<ul style="list-style-type: none">By default, a maximum of 50,000 Microsoft Entra resources can be created in a single tenant by users of the Microsoft Entra ID Free edition. If you have at least one verified domain, the default Microsoft Entra service quota for your organization is extended to 300,000 Microsoft Entra resources. The Microsoft Entra service quota for organizations created by self-service sign-up remains 50,000 Microsoft Entra resources, even after you perform an internal admin takeover and the organization is converted to a managed tenant with at least one verified domain. This service limit is unrelated to the pricing tier limit of 500,000 resources on the Microsoft Entra pricing page. To go beyond the default quota, you must contact Microsoft Support.A non-admin user can create no more than 250 Microsoft Entra resources. Both active resources and deleted resources that are available to restore count toward this quota. Only deleted Microsoft Entra resources that were deleted fewer than 30 days ago are available to restore. Deleted Microsoft Entra resources that are no longer available to restore count toward this quota at a value of one-quarter for 30 days. If you have developers who are likely to repeatedly exceed this quota in the course of their regular duties, you can create and assign a custom role with permission to create a limitless number of app registrations.

Category	Limit
	<ul style="list-style-type: none"> Resource limitations apply to all directory objects in a given Microsoft Entra tenant, including users, groups, applications, and service principals.
Schema extensions	<ul style="list-style-type: none"> String-type extensions can have a maximum of 256 characters. Binary-type extensions are limited to 256 bytes. Only 100 extension values, across <i>all</i> types and <i>all</i> applications, can be written to any single Microsoft Entra resource. Only User, Group, TenantDetail, Device, Application, and ServicePrincipal entities can be extended with string-type or binary-type single-valued attributes.
Applications	<ul style="list-style-type: none"> A maximum of 100 users and service principals can be owners of a single application. A user, group, or service principal can have a maximum of 1,500 app role assignments. The limitation is on the assigned service principal, user, or group across all app roles and not on the number of assignments of a single app role. This limit includes app role assignments where the resource service principal has been soft-deleted. A user can have credentials configured for a maximum of 48 apps using password-based single sign-on. This limit only applies for credentials configured when the user is directly assigned the app, not when the user is a member of a group that is assigned. A group can have credentials configured for a maximum of 48 apps using password-based single sign-on. See additional limits in Validation differences by supported account types.
Application manifest	<p>A maximum of 1,200 entries can be added to the application manifest.</p> <p>See additional limits in Validation differences by supported account types.</p>
Groups	<ul style="list-style-type: none"> A non-admin user can create a maximum of 250 groups in a Microsoft Entra organization. Any Microsoft Entra admin who can manage groups in the organization can also create an unlimited number of groups (up to the Microsoft Entra object limit). If you assign a role to a user to remove the limit for that user, assign a less privileged, built-in role such as User Administrator or Groups Administrator. A Microsoft Entra organization can have a maximum of 15,000 dynamic groups and dynamic administrative units combined. A maximum of 500 role-assignable groups can be created in a single Microsoft Entra organization (tenant). A maximum of 100 users can be owners of a single group. Any number of Microsoft Entra resources can be members of a single group.

Category	Limit
	<ul style="list-style-type: none"> • A user can be a member of any number of groups. When security groups are being used in combination with SharePoint Online, a user can be a part of 2,049 security groups in total. This includes both direct and indirect group memberships. When this limit is exceeded, authentication and search results become unpredictable. • Starting with Microsoft Entra Connect v2.0, the V2 endpoint is the default API. The number of members in a group that you can synchronize from your on-premises Active Directory to Microsoft Entra ID by using Microsoft Entra Connect is limited to 250,000 members. For more information, see Microsoft Entra Connect Sync V2. • When you select a list of groups, you can assign a group expiration policy to a maximum of 500 Microsoft 365 groups. There's no limit when the policy is applied to all Microsoft 365 groups.

At this time, the following scenarios are supported with nested groups:

- One group can be added as a member of another group, and you can achieve group nesting.
- Group membership claims. When an app is configured to receive group membership claims in the token, nested groups in which the signed-in user is a member are included.
- Conditional Access (when a Conditional Access policy has a group scope).
- Restricting access to self-serve password reset.
- Restricting which users can do Microsoft Entra join and device registration.

The following scenarios are *not* supported with nested groups:

- App role assignment, for both access and provisioning. Assigning groups to an app is supported, but any groups nested within the directly assigned group won't have access.
- Group-based licensing (assigning a license automatically to all members of a group).
- Microsoft 365 Groups.

Category	Limit
Application Proxy	<ul style="list-style-type: none"> A maximum of 500 transactions* per second per Application Proxy application. A maximum of 750 transactions per second for the Microsoft Entra organization.
	<p>*A transaction is defined as a single HTTP request and response for a unique resource. When clients are throttled, they receive a 429 response (too many requests). Transaction metrics are collected on each connector and can be monitored using performance counters under the object name <code>Microsoft Entra private network connector</code>.</p>
Access Panel	There's no limit to the number of applications per user that can be displayed in the Access Panel, regardless of the number of assigned licenses.
Reports	A maximum of 1,000 rows can be viewed or downloaded in any report. Any additional data is truncated.
Administrative units	<ul style="list-style-type: none"> A Microsoft Entra resource can be a member of no more than 30 administrative units. A maximum of 100 restricted management administrative units in a tenant. A Microsoft Entra organization can have a maximum of 15,000 dynamic membership groups and dynamic administrative units combined.
Microsoft Entra roles and permissions	<ul style="list-style-type: none"> A maximum of 100 Microsoft Entra custom roles can be created in a Microsoft Entra organization. A maximum of 150 Microsoft Entra custom role assignments for a single principal at any scope. A maximum of 100 Microsoft Entra built-in role assignments for a single principal at non-tenant scope (such as an administrative unit or Microsoft Entra object). There's no limit to Microsoft Entra built-in role assignments at tenant scope. For more information, see Assign Microsoft Entra roles. A group can't be added as a group owner. A user's ability to read other users' tenant information can be restricted only by the Microsoft Entra organization-wide switch to disable all non-admin users' access to all tenant information (not recommended). For more information, see To restrict the default permissions for member users. It might take up to 15 minutes or you might have to sign out and sign back in before admin role membership additions and revocations take effect.
Conditional Access Policies	A maximum of 195 policies can be created in a single Microsoft Entra organization (tenant).

Category	Limit
Terms of use	You can add no more than 40 terms to a single Microsoft Entra organization (tenant).
Multitenant organizations	<ul style="list-style-type: none">A maximum of 100 active tenants, including the owner tenant. The owner tenant can add more than 100 pending tenants, but they won't be able to join the multitenant organization if the limit is exceeded. This limit is applied at the time a pending tenant joins a multitenant organization. <p>This limit is specific to the number of tenants in a multitenant organization. It doesn't apply to cross-tenant synchronization by itself.</p>

Related content

- [Configure group claims for applications by using Microsoft Entra ID](#)
- [Sign up for Azure as an organization](#)
- [How Azure subscriptions are associated with Microsoft Entra ID](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Microsoft Entra feature availability

Article • 03/04/2025

The following tables list Microsoft Entra feature availability in Azure Government.

Microsoft Entra ID

[Expand table](#)

Service	Feature	Availability
Authentication, single sign-on, and MFA	Cloud authentication (Pass-through authentication, password hash synchronization)	✓
	Federated authentication (Active Directory Federation Services or federation with other identity providers)	✓
	Single sign-on (SSO) unlimited	✓
	Multifactor authentication (MFA)	✓
	Passwordless (Windows Hello for Business, Microsoft Authenticator, FIDO2 security key integrations)	✓
	Certificate-based authentication	✓
	Service-level agreement	✓
Applications access	SaaS apps with modern authentication (Microsoft Entra application gallery apps, SAML, and OAUTH 2.0)	✓
	Group assignment to applications	✓
	Cloud app discovery (Microsoft Defender for Cloud Apps)	✓
	Application Proxy for on-premises, header-based, and Integrated Windows Authentication	✓
	Secure hybrid access partnerships (Kerberos, NTLM, LDAP, RDP, and SSH authentication)	✓

Service	Feature	Availability
Authorization and Conditional Access	Role-based access control (RBAC)	
	Conditional Access	
	SharePoint limited access	
	Session lifetime management	
Identity Protection	ID Protection (vulnerabilities and risky accounts)	See Microsoft Entra ID Protection below.
	ID Protection (risk events investigation, SIEM connectivity)	See Microsoft Entra ID Protection below.
Administration and hybrid identity	User and group management	
	Advanced group management (Dynamic groups, naming policies, expiration, default classification)	
	Directory synchronization—Microsoft Entra Connect (sync and cloud sync)	
	Microsoft Entra Connect Health reporting	
	Delegated administration—built-in roles	
	Global password protection and management – cloud-only users	
	Global password protection and management – custom banned passwords, users synchronized from on-premises Active Directory	
End-user self-service	Microsoft Identity Manager user client access license (CAL)	
	Application launch portal (My Apps)	
	User application collections in My Apps	
	Self-service account management portal (My Account)	
	Self-service password change for cloud users	
Self-service password reset/change/unlock	Self-service password reset/change/unlock	

Service	Feature	Availability
	with on-premises write-back	
	Self-service sign-in activity search and reporting	✓
	Self-service group management (My Groups)	✓
	Self-service entitlement management (My Access)	✓
Identity governance	Automated user provisioning to apps	✓
	Automated group provisioning to apps	✓
	HR-driven provisioning	Partial. See HR-provisioning apps .
	Terms of use	✓
	Access reviews	✓
	Entitlement management	✓
	Privileged Identity Management (PIM)	✓
	Lifecycle workflows, in Microsoft Entra ID Governance	✓
Event logging and reporting	Basic security and usage reports	✓
	Advanced security and usage reports	✓
	ID Protection: vulnerabilities and risky accounts	✓
	ID Protection: risk events investigation, SIEM connectivity	✓
Frontline workers	SMS sign-in	✓
	Shared device sign-out	Enterprise state roaming for Windows 10 devices isn't available.
	Delegated user management portal (My Staff)	✗

Microsoft Entra ID Protection

[\[\] Expand table](#)

Risk Detection	Availability
Leaked credentials (MACE)	✓
Microsoft Entra threat intelligence	✗
Anonymous IP address	✓
Atypical travel	✓
Anomalous Token	✓
Token Issuer Anomaly	✓
Malware linked IP address	✓
Suspicious browser	✓
Unfamiliar sign-in properties	✓
Admin confirmed user compromised	✓
Malicious IP address	✓
Suspicious inbox manipulation rules	✓
Password spray	✓
Impossible travel	✓
New country	✓
Activity from anonymous IP address	✓
Suspicious inbox forwarding	✓
Additional risk detected	✓

HR provisioning apps

[\[\] Expand table](#)

HR-provisioning app	Availability
Workday to Microsoft Entra user provisioning	✓
Workday Writeback	✓

HR-provisioning app	Availability
SuccessFactors to Microsoft Entra user provisioning	✓
SuccessFactors to Writeback	✓
API-driven inbound provisioning	✗
Provisioning agent configuration and registration with Gov cloud tenant	Works with special undocumented command-line invocation: <pre>AADConnectProvisioningAgent.Installer.exe ENVIRONMENTNAME=AzureUSGovernment</pre>

Other Microsoft Entra products

[Microsoft Entra ID Governance](#) is available in the US Government community cloud (GCC), GCC-High, and Department of Defense cloud environments. [Microsoft Entra Workload Identities Premium edition](#) is available in the US government clouds. [Microsoft Entra Permissions Management](#) is not available in the US government or US national clouds.

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#)

Microsoft Entra releases and announcements

Article • 04/30/2025

This article provides information about the latest releases and change announcements across the Microsoft Entra family of products over the last six months (updated monthly). If you're looking for information that's older than six months, see: [Archive for What's new in Microsoft Entra](#).

Get notified about when to revisit this page for updates by copying and pasting this URL:

`https://learn.microsoft.com/api/search/rss?search=%22Release+notes+-+Azure+Active+Directory%22&locale=en-us` into your  feed reader.

April 2025

Public Preview - Conditional Access Optimization Agent in Microsoft Entra

Type: New feature

Service category: Conditional Access

Product capability: Identity Security & Protection

[Conditional Access Optimization Agent in Microsoft Entra](#) monitors for new users or apps not covered by existing policies, identifies necessary updates to close security gaps, and recommends quick fixes for identity teams to apply with a single selection. For more information, see: [Microsoft Entra Conditional Access optimization agent](#).

Public Preview - Microsoft Entra ID Governance: Suggested access packages in My Access

Type: New feature

Service category: Entitlement Management

Product capability: Entitlement Management

In December 2024, we introduced a new feature in My Access: a curated list of suggested access packages. Users view the most relevant access packages, based on their peers' access packages and previous assignments, without scrolling through a long list. By May 2025, suggestions will be enabled by default and we'll introduce a new card in the Microsoft Entra

Admin Center Entitlement Management control configurations for admins to see My Access settings. We recommend admins turn on the peer-based insights for suggested access packages via this setting. For more information, see: [Suggested access packages in My Access \(Preview\)](#).

Public Preview - Conditional Access What If evaluation API

Type: New feature

Service category: Conditional Access

Product capability: Access Control

Conditional Access What If evaluation API – Leverage the What If tool using the Microsoft Graph API to programmatically evaluate the applicability of conditional access policies in your tenant on user and service principal sign-ins. For more information, see: [conditionalAccessRoot: evaluate](#).

Public Preview - Manage refresh tokens for mover and leaver scenarios with Lifecycle Workflows

Type: New feature

Service category: Lifecycle Workflows

Product capability: Identity Governance

Now customers can configure a Lifecycle workflows task to automatically revoke access tokens when employees move within, or leave, the organization. For more information, see: [Revoke all refresh tokens for user \(Preview\)](#).

General Availability - Use managed identities as credentials in Microsoft Entra apps

Type: New feature

Service category: Managed identities for Azure resources

Product capability: Identity Security & Protection

You can now use managed identities as federated credentials for Microsoft Entra apps, enabling secure, secret-less authentication in both single- and multi-tenant scenarios. This eliminates the need to store and manage client secrets or certificates when using Microsoft Entra app to access Azure resources across tenants. This capability aligns with Microsoft's

Secure Future Initiative [🔗](#) pillar of protecting identities and secrets across systems. Learn how to configure this capability in the [official documentation](#).

Plan for change - Roll out of Application Based Authentication on Microsoft Entra Connect Sync

Type: Plan for change

Service category: Microsoft Entra Connect

Product capability: Microsoft Entra Connect

What is changing

Microsoft Entra Connect creates and uses a [Microsoft Entra Connector account](#) to authenticate and sync identities from Active Directory to Microsoft Entra ID. The account uses a locally stored password to authenticate with Microsoft Entra ID. To enhance the security of the Microsoft Entra Connect application sync process, we will, in the coming week roll out support for "Application based Authentication" (ABA), which uses a Microsoft Entra ID application based identity and Oauth 2.0 client credential flow to authenticate with Microsoft Entra ID. To enable this, Microsoft Entra Connect will create a single tenant 3rd party application in customer's Microsoft Entra ID tenant, register a certificate as the credential for the application, and authorize the application to perform on-premises directory synchronization

The Microsoft Entra Connect Sync .msi installation file for this change will be exclusively available in the Microsoft Entra admin center within the [Microsoft Entra Connect pane](#) [🔗](#).

Check our [version history page](#) in the next week for more details of the change.

March 2025

Microsoft Entra Permissions Management end of sale and retirement

Type: Plan for change

Service category: Other

Product capability: Permissions Management

Effective April 1, 2025, Microsoft Entra Permissions Management (MEPM) will no longer be available for sale to new Enterprise Agreement or direct customers. Additionally, starting May

1, it will not be available for sale to new CSP customers. Effective October 1, 2025, we will retire Microsoft Entra Permissions Management and discontinue support of this product.

Existing customers will retain access to this product until September 30, 2025, with ongoing support for current functionalities. We have partnered with Delinea to provide an alternative solution, [Privilege Control for Cloud Entitlements \(PCCE\)](#), that offers similar capabilities to those provided by Microsoft Entra Permissions Management. The decision to phase out Microsoft Entra Permissions Management was done after deep consideration of our innovation portfolio and how we can focus on delivering the best innovations aligned to our differentiating areas and partner with the ecosystem on adjacencies. We remain committed to delivering top-tier solutions across the Microsoft Entra portfolio. For more information, see: [Important change announcement: Microsoft Entra Permissions Management end of sale and retirement](#).

Public Preview - Track and investigate identity activities with linkable identifiers in Microsoft Entra

Type: New feature

Service category: Authentications (Logins)

Product capability: User Authentication

Microsoft will standardize the linkable token identifiers, and expose them in both Microsoft Entra and workflow audit logs. This allows customers to join the logs to track, and investigate, any malicious activity. Currently linkable identifiers are available in Microsoft Entra sign in logs, Exchange Online audit logs, and MSGraph Activity logs.

For more information, see: [Track and investigate identity activities with linkable identifiers in Microsoft Entra \(preview\)](#).

General Availability- Conditional Access reauthentication policy

Type: New feature

Service category: Conditional Access

Product capability: Identity Security & Protection

Require reauthentication every time can be used for scenarios where you want to require a fresh authentication, every time a user performs specific actions like accessing sensitive

applications, securing resources behind VPN, or Securing privileged role elevation in PIM. For more information, see: [Require reauthentication every time](#).

General Availability- Custom Attributes support for Microsoft Entra Domain Services

Type: New feature

Service category: Microsoft Entra Domain Services

Product capability: Microsoft Entra Domain Services

Custom Attributes for Microsoft Entra Domain Services is now Generally Available. This capability allows customers to use Custom Attributes in their managed domains. Legacy applications often rely on custom attributes created in the past to store information, categorize objects, or enforce fine-grained access control over resources. For example, these applications might use custom attributes to store an employee ID in their directory and rely on these attributes in their application LDAP calls. Modifying legacy applications can be costly and risky, and customers might lack the necessary skills or knowledge to make these changes. Microsoft Entra Domain Services now supports custom attributes, enabling customers to migrate their legacy applications to the Azure cloud without modification. It also provides support to synchronize custom attributes from Microsoft Entra ID, allowing customers to benefit from Microsoft Entra ID services in the cloud. For more information, see: [Custom attributes for Microsoft Entra Domain Services](#).

Public Preview - Conditional Access Per-Policy Reporting

Type: New feature

Service category: Conditional Access

Product capability: Identity Security & Protection

Conditional Access Per-Policy Reporting enables admins to easily evaluate the impact of enabled and report-only Conditional Access policies on their organization, without using Log Analytics. This feature surfaces a graph for each policy in the Microsoft Entra Admin Center, visualizing the policy's impact on the tenant's past sign-ins. For more information, see: [Policy impact \(Preview\)](#).

Public Preview - Limit creation or promotion of multitenant apps

Type: New feature

Service category: Directory Management

Product capability: Developer Experience

A new feature has been added to the [App Management Policy Framework](#) that allows restriction on creation or promotion of multitenant applications, providing administrators with greater control over their app environments.

Administrators can now configure tenant default or custom app policy using the new '[audiences](#)' restriction to block new app creation if the signInAudience value provided in the app isn't permitted by the policy. In addition, existing apps can be restricted from changing their signInAudience if the target value isn't permitted by the policy. These policy changes are applied during app creation or update operations, offering control over application deployment and usage. For more information, see: [audiencesConfiguration resource type](#).

General Availability - Download Microsoft Entra Connect Sync on the Microsoft Entra admin center

Type: Plan for change

Service category: Microsoft Entra Connect

Product capability: Identity Governance

The Microsoft Entra Connect Sync .msi installation files are also available on Microsoft Entra admin center within the [Microsoft Entra Connect pane](#). As part of this change, we'll stop uploading new installation files on the [Microsoft Download Center](#).

General Availability - New Microsoft-managed Conditional Access policies designed to limit device code flow and legacy authentication flows

Type: Changed feature

Service category: Conditional Access

Product capability: Access Control

As part of our ongoing commitment to enhance security and protect our customers from evolving cyber threats, we're rolling out two new Microsoft-managed Conditional Access policies designed to limit device code flow and legacy authentication flows. These policies are aligned to the secure by default principle of our broader [Secure Future Initiative](#), which aims to provide robust security measures to safeguard your organization by default.

Deprecated - Upgrade your Microsoft Entra Connect Sync version to avoid impact on the Sync Wizard

Type: Deprecated

Service category: Microsoft Entra Connect

Product capability: Microsoft Entra Connect

As announced in the Microsoft Entra What's New [Blog](#) and in Microsoft 365 Center communications, customers should upgrade their connect sync versions to at least [2.4.18.0](#) for commercial clouds and [2.4.21.0](#) for non-commercial clouds before April 7, 2025. A breaking change on the Connect Sync Wizard will affect all requests that require authentication such as schema refresh, configuration of staging mode, and user sign in changes. For more information, see: [Minimum versions](#).

February 2025

General Availability - Authentication methods migration wizard

Type: New feature

Service category: MFA

Product capability: User Authentication

The authentication methods migration guide in the Microsoft Entra Admin Center lets you automatically migrate method management from the [legacy MFA and SSPR policies](#) to the [converged authentication methods policy](#). In 2023, it was announced that the ability to manage authentication methods in the legacy MFA and SSPR policies would be retired in September 2025. Until now, organizations had to manually migrate methods themselves by using [the migration toggle](#) in the converged policy. Now, you can migrate in just a few selections by using the migration guide. The guide evaluates what your organization currently has enabled in both legacy policies, and generates a recommended converged policy configuration for you to review and edit as needed. From there, confirm the configuration, and we set it up for you and mark your migration as complete. For more information, see: [How to migrate MFA and SSPR policy settings to the Authentication methods policy for Microsoft Entra ID](#).

Public Preview - Enhanced user management in Admin Center UX

Type: New feature

Service category: User Management

Product capability: User Management

Admins are now able to multi-select and edit users at once through the Microsoft Entra Admin Center. With this new capability, admins can bulk edit user properties, add users to groups, edit account status, and more. This UX enhancement will significantly improve efficiency for user management tasks in the Microsoft Entra admin center. For more information, see: [Add or update a user's profile information and settings in the Microsoft Entra admin center.](#)

Public Preview – QR code authentication, a simple and fast authentication method for Frontline Workers

Type: New feature

Service category: Authentications (Logins)

Product capability: User Authentication

We're thrilled to announce public preview of QR code authentication in Microsoft Entra ID, providing an efficient and simple authentication method for frontline workers.

You see a new authentication method 'QR code' in Microsoft Entra ID Authentication method Policies. You can enable and add QR code for your frontline workers via Microsoft Entra ID, My Staff, or MS Graph APIs. All users in your tenant see a new link 'Sign in with QR code' on navigating to <https://login.microsoftonline.com> > 'Sign-in options' > 'Sign in to an organization' page. This new link is visible only on mobile devices (Android/iOS/iPadOS). Users can use this auth method only if you add and provide a QR code to them. QR code auth is also available in BlueFletch and Jamf. MHS QR code auth support is generally available by early March.

The feature has a 'preview' tag until it's generally available. For more information, see: [Authentication methods in Microsoft Entra ID - QR code authentication method \(Preview\).](#)

Public Preview - Custom SAML/WS-Fed External Identity Provider Support in Microsoft Entra External ID

Type: New feature

Service category: B2C - Consumer Identity Management

Product capability: B2B/B2C

By setting up federation with a custom-configured identity provider that supports the SAML 2.0 or WS-Fed protocol, you enable your users to sign up and sign in to your applications using their existing accounts from the federated external provider.

This feature also includes domain-based federation, so a user who enters an email address on the sign-in page that matches a predefined domain in any of the external identity providers will be redirected to authenticate with that identity provider.

For more information, see: [Custom SAML/WS-Fed identity providers \(preview\)](#).

Public Preview - External Auth Methods support for system preferred MFA

Type: New feature

Service category: MFA

Product capability: 3rd Party Integration

Support for external auth methods as a supported method begins rolling out at the beginning of March 2025. When this is live in a tenant where system preferred is enabled and users are in scope of an external auth methods policy, those users will be prompted for their external authentication method if their most secure registered method is Microsoft Authenticator notification. External Authentication Method will appear as third in the list of most secure methods. If the user has a Temporary Access Pass (TAP) or Passkey (FIDO2) device registered, they'll be prompted for those. In addition, users in the scope of an external auth methods policy will have the ability to delete all registered second factor methods from their account, even if the method being deleted is specified as the default sign in method or is system preferred. For more information, see: [System-preferred multifactor authentication - Authentication methods policy](#).

General Availability - Granular Microsoft Graph permissions for Lifecycle workflows

Type: New feature

Service category: Lifecycle Workflows

Product capability: Identity Governance

Now new, lesser privileged permissions can be used for managing specific read and write actions in Lifecycle workflows scenarios. The following granular permissions were introduced in Microsoft Graph:

- LifecycleWorkflows-Workflow.ReadBasic.All
- LifecycleWorkflows-Workflow.Read.All
- LifecycleWorkflows-Workflow.ReadWrite.All
- LifecycleWorkflows-Workflow.Activate
- LifecycleWorkflows-Reports.Read.All
- LifecycleWorkflows-CustomExt.Read.All
- LifecycleWorkflows-CustomExt.ReadWrite.All

For more information, see: [Microsoft Graph permissions reference](#).

January 2025

Public Preview - Manage Lifecycle Workflows with Microsoft Security CoPilot in Microsoft Entra

Type: New feature

Service category: Lifecycle Workflows

Product capability: Identity Governance

Customers can now manage, and customize, Lifecycle Workflows using natural language with Microsoft Security CoPilot. Our Lifecycle Workflows (LCW) Copilot solution provides step-by-step guidance to perform key workflow configuration and execution tasks using natural language. It allows customers to quickly get rich insights to help monitor, and troubleshoot, workflows for compliance. For more information, see: [Manage employee lifecycle using Microsoft Security Copilot \(Preview\)](#).

General Availability - Microsoft Entra PowerShell

Type: New feature

Service category: MS Graph

Product capability: Developer Experience

Manage and automate Microsoft Entra resources programmatically with the scenario-focused Microsoft Entra PowerShell module. For more information, see: [Microsoft Entra PowerShell module now generally available ↗](#).

General Availability - Improving visibility into downstream tenant sign-ins

Type: New feature

Service category: Reporting

Product capability: Monitoring & Reporting

Microsoft Security wants to ensure that all customers are aware of how to notice when a partner is accessing a downstream tenant's resources. Interactive sign-in logs currently provide a list of sign in events, but there's no clear indication of which logins are from partners accessing downstream tenant resources. For example, when reviewing the logs, you might see a series of events, but without any additional context, it's difficult to tell whether these logins are from a partner accessing another tenant's data.

Here's a list of steps that one can take to clarify which logins are associated with partner tenants:

1. Take note of the "ServiceProvider" value in the CrossTenantAccessType column:

- This filter can be applied to refine the log data. When activated, it immediately isolates events related to partner logins.

2. Utilize the "Home Tenant ID" and "Resource Tenant ID" Columns:

- These two columns identify logins coming from the partner's tenant to a downstream tenant.

After seeing a partner logging into a downstream tenant's resources, an important follow-up activity to perform is to validate the activities that might have occurred in the downstream environment. Some examples of logs to look at are Microsoft Entra Audit logs for Microsoft Entra ID events, Microsoft 365 Unified Audit Log (UAL) for Microsoft 365 and Microsoft Entra ID events, and/or the Azure Monitor activity log for Azure events. By following these steps, you're able to clearly identify when a partner is logging into a downstream tenant's resources and subsequent activity in the environment, enhancing your ability to manage and monitor cross-tenant access efficiently.

To increase visibility into the aforementioned columns, Microsoft Entra will begin enabling these columns to display by default when loading the sign-in logs UX starting on March 7, 2025.

Public Preview - Auditing administrator events in Microsoft Entra Connect

Type: New feature

Service category: Microsoft Entra Connect

Product capability: Microsoft Entra Connect

We have released a new version of Microsoft Entra Connect, version 2.4.129.0, that supports the logging of the changes an administrator makes on the Connect Sync Wizard and PowerShell. For more information, see: [Auditing administrator events in Microsoft Entra Connect Sync \(Public Preview\)](#).

Where supported, we'll also autoupgrade customers to this version of Microsoft Entra Connect in February 2025. For customers who wish to be autoupdated, [ensure that you have auto-upgrade configured](#).

For upgrade-related guidance, see [Microsoft Entra Connect: Upgrade from a previous version to the latest](#).

Public Preview - Flexible Federated Identity Credentials

Type: New feature

Service category: Authentications (Logins)

Product capability: Developer Experience

Flexible Federated Identity Credentials extend the existing Federated Identity Credential model by providing the ability to use wildcard matching against certain claims. Currently available for GitHub, GitLab, and Terraform Cloud scenarios, this functionality can be used to lower the total number of FICs required to manage similar scenarios. For more information, see: [Flexible federated identity credentials \(preview\)](#).

General Availability - Real-time Password Spray Detection in Microsoft Entra ID Protection

Type: New feature

Service category: Identity Protection

Product capability: Identity Security & Protection

Traditionally, password spray attacks are detected post breach or as part of hunting activity. Now, we've enhanced Microsoft Entra ID Protection to detect password spray attacks in real-

time before the attacker ever obtains a token. This reduces remediation from hours to seconds by interrupting attacks during the sign-in flow.

Risk-based Conditional Access can automatically respond to this new signal by raising session risk, immediately challenging the sign-in attempt, and stopping password spray attempts in their tracks. This cutting-edge detection, now Generally Available, works alongside existing detections for advanced attacks such as Adversary-in-the-Middle (AitM) phishing and token theft, to ensure comprehensive coverage against modern attacks. For more information, see: [What is Microsoft Entra ID Protection?](#)

General Availability - Protected actions for hard deletions

Type: New feature

Service category: Other

Product capability: Identity Security & Protection

Customers can now configure Conditional Access policies to protect against early hard deletions. Protected action for hard deletion protects hard deletion of users, Microsoft 365 groups, and applications. For more information, see: [What are protected actions in Microsoft Entra ID?](#).

Public Preview - Elevate Access events are now exportable via Microsoft Entra Audit Logs

Type: New feature

Service category: RBAC

Product capability: Monitoring & Reporting

This feature enables administrators to export and stream Elevate Access events to both first-party and third-party SIEM solutions via Microsoft Entra Audit logs. It enhances detection and improves logging capabilities, allowing visibility into who in their tenant has utilized Elevate Access. For more information on how to use the feature, see: [View elevate access log entries](#).

Deprecated - Action Required by February 1, 2025: Azure AD Graph retirement

Type: Deprecated

Service category: Azure AD Graph

Product capability: Developer Experience

The Azure AD Graph API service was [deprecated] in 2020. [Retirement of the Azure AD Graph API service](#) began in September 2024, and the next phase of this retirement starts February 1, 2025. This phase will impact new and existing applications unless action is taken. The latest updates on Azure AD Graph retirement can be found here: [Take action by February 1: Azure AD Graph is retiring](#).

Starting from February 1, both new and existing applications will be prevented from calling Azure AD Graph APIs, unless they're configured for an extension. You might not see impact right away, as we're rolling out this change in stages across tenants. We anticipate full deployment of this change around the end of February, and by the end of March for national cloud deployments.

If you haven't already, it's now urgent to review the applications on your tenant to see which ones depend on Azure AD Graph API access, and mitigate or migrate these before the February 1 cutoff date. For applications that haven't migrated to Microsoft Graph APIs, [an extension](#) can be set to allow the application access to Azure AD Graph through June 30, 2025.

Microsoft Entra Recommendations are the best tool to identify applications that are using Azure AD Graph APIs in your tenant and require action. Reference this blog post: Action required: [Azure AD Graph API retirement](#) for step by step guidance.

General Availability - Microsoft Entra Connect Version 2.4.129.0

Type: Changed feature

Service category: Microsoft Entra Connect

Product capability: Microsoft Entra Connect

On January 15, 2025, we released Microsoft Entra Connect Sync Version 2.4.129.0 which supports auditing administrator events. More details are available in the [release notes](#). We'll automatically upgrade eligible customers to this latest version of Microsoft Entra Connect in February 2025. For customers who wish to be auto-upgraded, [ensure that you have auto-upgrade configured](#).

Deprecated - Take action to avoid impact when legacy MSOnline and AzureAD PowerShell modules retire

Type: Deprecated

Service category: Legacy MSOnline and AzureAD PowerShell modules

Product capability: Developer Experience

As announced in Microsoft Entra [change announcements](#) and in the Microsoft Entra [Blog](#), the MSOnline, and Microsoft Azure AD PowerShell modules (for Microsoft Entra ID) retired on March 30, 2024.

The retirement for MSOnline PowerShell module starts in early April 2025, and ends in late May 2025. If you're using MSOnline PowerShell, you must take action by March 30, 2025 to avoid impact after the retirement by migrating any use of MSOnline to [Microsoft Graph PowerShell SDK](#) or [Microsoft Entra PowerShell](#).

Key points

- MSOnline PowerShell will retire, and stop working, between early April 2025 and late May 2025
- AzureAD PowerShell will no longer be supported after March 30, 2025, but its retirement will happen in early July 2025. This postponement is to allow you time to finish the MSOnline PowerShell migration
- To ensure customer readiness for MSOnline PowerShell retirement, a series of temporary outage tests will occur for all tenants between January 2025 and March 2025.

For more information, see: [Action required: MSOnline and AzureAD PowerShell retirement - 2025 info and resources](#).

December 2024

General Availability - What's new in Microsoft Entra

Type: New feature

Service category: Reporting

Product capability: Monitoring & Reporting

What's new in Microsoft Entra offers a comprehensive view of Microsoft Entra product updates including product roadmap (like Public Previews and recent GAs), and change announcements (like deprecations, breaking changes, feature changes and Microsoft-managed policies). It's a one stop shop for Microsoft Entra admins to discover the product updates.

Public Preview - Microsoft Entra ID Governance: Approvers can revoke access in MyAccess

Type: New feature

Service category: Entitlement Management

Product capability: Entitlement Management

For Microsoft Entra ID Governance users, approvers of access package requests can now revoke their decision in MyAccess. Only the person who took the approve action is able to revoke access. To opt into this feature, admins can go to the [Identity Governance settings page](#), and enable the feature. For more information, see: [What is the My Access portal?](#).

General Availability - Expansion of SSPR Policy Audit Logging

Type: New feature

Service category: Self Service Password Reset

Product capability: Monitoring & Reporting

Starting Mid-January, we are improving the audit logs for changes made to the SSPR Policy.

With this improvement, any change to the SSPR policy configuration, including enablement or disablement, will result in an audit log entry that includes details about the change made. Additionally, both the previous values and current values from the change will be recorded within the audit log. This additional information can be found by selecting an audit log entry and selecting the Modified Properties tab within the entry.

These changes are rolled out in phases:

- Phase 1 includes logging for the Authentication Methods, Registration, Notifications, and Customization configuration settings.
- Phase 2 includes logging for the On-premises integration configuration settings.

This change occurs automatically, so admins take no action. For more information and details regarding this change, see: [Microsoft Entra audit log categories and activities](#).

General Availability - Update Profile Photo in MyAccount

Type: New feature

Service category: My Profile/Account

Product capability: End User Experiences

Users can now update their profile photo directly from their MyAccount portal. This change exposes a new edit button on the profile photo section of the user's account.

In some environments, it's necessary to prevent users from making this change. Global Administrators can manage this using a tenant-wide policy with Microsoft Graph API, following the guidance in the [Manage user profile photo settings in Microsoft 365](#) document.

General Availability - Temporary Access Pass (TAP) support for internal guest users

Type: New feature

Service category: MFA

Product capability: Identity Security & Protection

Microsoft Entra ID now supports issuing Temporary Access Passes (TAP) to internal guest users. TAPs can be issued to internal guests just like normal members, through the Microsoft Entra ID Admin Center, or natively through Microsoft Graph. With this enhancement, internal guests can now seamlessly onboard, and recover, their accounts with time-bound temporary credentials.

For more information, see: [Configure Temporary Access Pass to register passwordless authentication methods](#).

Public Preview - Microsoft Entra ID Governance: access package request suggestions

Type: New feature

Service category: Entitlement Management

Product capability: Entitlement Management

Opt-In As communicated [earlier](#), we're excited to introduce a new feature in [My Access](#): a curated list of suggested access packages. This capability allows users to quickly view the most relevant access packages (based off their peers' access packages and previous requests) without scrolling through a long list. In December you can [enable the preview in the Opt-in Preview Features for Identity Governance](#). From January, this setting is enabled by default.

Public Preview - Security Copilot embedded in Microsoft Entra

Type: New feature

Service category: Other

Product capability: Identity Security & Protection

We've announced the public preview of Microsoft Security Copilot embedded in the Microsoft Entra admin Center. This integration brings all identity skills previously made generally available for the Security Copilot standalone experience in April 2024, along with new identity capabilities for admins and security analysts to use directly within the Microsoft Entra admin center. We've also added brand new skills to help improve identity-related risk investigation. In December, we broaden the scope even further to include a set of skills specifically for App Risk Management in both standalone and embedded experiences of Security Copilot and Microsoft Entra. These capabilities allow identity admins and security analysts to better identify, understand, and remediate the risks impacting applications and workload identities registered in Microsoft Entra.

With Security Copilot now embedded in Microsoft Entra, identity admins get AI-driven, natural-language summaries of identity context and insights tailored for handling security incidents, equipping them to better protect against identity compromise. The embedded experience also accelerates troubleshooting tasks like resolving identity-related risks and sign-in issues, without ever leaving the admin center.

Public Preview - Security Copilot in Microsoft Entra: App Risk skills

Type: New feature

Service category: Other

Product capability: Identity Security & Protection

Identity admins and security analysts managing Microsoft Entra ID registered apps can identify and understand risks through natural language prompts. Security Copilot has links to the Microsoft Entra Admin Center for admins to take needed remediation actions. For more information, see: [Assess application risks using Microsoft Security Copilot in Microsoft Entra](#).

Public Preview - Provision custom security attributes from HR sources

Type: New feature

Service category: Provisioning

Product capability: Inbound to Entra ID

With this feature, customers can automatically provision "*custom security attributes*" in Microsoft Entra ID from authoritative HR sources. Supported authoritative sources include: Workday, SAP SuccessFactors, and any HR system integrated using API-driven provisioning.

Public Preview - Sign in with Apple

Type: New feature

Service category: B2C - Consumer Identity Management

Product capability: Extensibility

This new feature adds Apple to our list of preconfigured social identity providers. As the first social identity provider implemented on the eSTS platform, it introduces a "*Sign in with Apple*" button to the sign-in options, allowing users to access applications with their Apple accounts. For more information, see: [Add Apple as an identity provider \(preview\)](#).

General Availability - Microsoft Entra External ID Custom URL Domains

Type: New feature

Service category: Authentications (Logins)

Product capability: Identity Lifecycle Management

This feature allows users to customize their Microsoft default sign in authentication endpoint with their own brand names. Custom URL Domains help users to change Ext ID endpoint < tenant-name >.ciamlogin.com to login.contoso.com.

General Availability - Privileged Identity Management integration in Azure Role Based Access Control

Type: New feature

Service category: RBAC

Product capability: Access Control

Privileged Identity Management (PIM) capabilities are now integrated into the Azure Role Based Access Control (Azure RBAC) UI. Before this integration, RBAC admins could only manage standing access (active permanent role assignments) from the Azure RBAC UI. With this integration, just-in-time access and timebound access, which are functionalities supported

by PIM, are now brought into the Azure RBAC UI for customers with either a P2, or Identity Governance, license.

RBAC admins can create assignments of type eligible and timebound duration from the Azure RBAC add role assignment flow, see the list of different states of role assignment in a single view, as well as convert the type and duration of their role assignments from the Azure RBAC UI. In addition, end users now see all their role assignments of different state straight from the Azure RBAC UI landing page, from where they can also activate their eligible role assignments. For more information, see: [List role assignments at a scope](#).

General Availability - Dedicated new 1st party resource application to enable Active Directory to Microsoft Entra ID sync using Microsoft Entra Connect Sync or Cloud Sync

Type: Changed feature

Service category: Provisioning

Product capability: Directory

As part of ongoing security hardening, Microsoft deployed Microsoft Entra AD Synchronization Service, a dedicated first-party application to enable the synchronization between Active Directory and Microsoft Entra ID. This new application, with Application ID `6bf85cfa-ac8a-4be5-b5de-425a0d0dc016`, was provisioned in customer tenants that use Microsoft Entra Connect Sync or the Microsoft Entra Cloud Sync service.

November 2024

Public Preview - Universal Continuous Access Evaluation

Type: New feature

Service category: Provisioning

Product capability: Network Access

Continuous Access Evaluation (CAE) revokes, and revalidates, network access in near real-time whenever Microsoft Entra ID detects changes to the identity. For more information, see: [Universal Continuous Access Evaluation \(Preview\)](#).

Public Preview - Microsoft Entra new store for certificate-based authentication

Type: New feature

Service category: Authentications (Logins)

Product capability: User Authentication

Microsoft Entra ID has a new scalable PKI (Public Key Infrastructure) based CA (Certificate Authorities) store with higher limits for the number of CAs and the size of each CA file. PKI based CA store allows CAs within each different PKI to be in its own container object allowing administrators to move away from one flat list of CAs to more efficient PKI container based CAs. PKI-based CA store now supports up to 250CAs, 8KB size for each CA and also supports issuers hints attribute for each CA. Administrators can also upload the entire PKI and all the CAs using the "Upload CBA PKI" feature or create a PKI container and upload CAs individually. For more information, see: [Step 1: Configure the certificate authorities with PKI-based trust store \(Preview\)](#).

Public Preview - Updating profile photo in MyAccount

Type: New feature

Service category: My Profile/Account

Product capability: End User Experiences

On November 13, 2024, users received the ability to update their profile photo directly from their [MyAccount](#) portal. This change exposes a new edit button on the profile photo section of the user's account.

In some environments, it's necessary to prevent users from making this change. Global Administrators can manage this using a tenant-wide policy with Microsoft Graph API, following the guidance in the [Manage user profile photo settings in Microsoft 365](#) document.

General Availability - Microsoft Entra Health Monitoring, Health Metrics Feature

Type: New feature

Service category: Reporting

Product capability: Monitoring & Reporting

Microsoft Entra health monitoring, available from the Health pane, includes a set of low-latency pre-computed health metrics that can be used to monitor the health of critical user scenarios

in your tenant. The first set of health scenarios includes MFA, CA-compliant devices, CA-managed devices, and SAML authentications. This set of monitor scenarios will grow over time. These health metrics are now released as general availability data streams with the public preview of an intelligent alerting capability. For more information, see: [What is Microsoft Entra Health?](#)

General Availability - Microsoft Entra Connect Sync Version 2.4.27.0

Type: Changed feature

Service category: Provisioning

Product capability: Identity Governance

On November 14, 2025, we released Microsoft Entra Connect Sync Version 2.4.27.0 that uses the OLE DB version 18.7.4 that further hardens our service. Upgrade to this latest version of connect sync to improve your security. More details are available in the [release notes](#).

Changed feature - expansion of WhatsApp as an MFA one-time passcode delivery channel for Microsoft Entra ID

Type: Changed feature

Service category: MFA

Product capability: User Authentication

In late 2023, Microsoft Entra ID started using WhatsApp as an alternate channel to deliver multifactor authentication (MFA) one-time passcodes to users in India and Indonesia. We saw improved deliverability, completion rates, and satisfaction when using the channel in both countries. The channel was temporarily disabled in India in early 2024. Starting early December 2024, we'll be re-enabling the channel in India, and expanding its use to more countries.

Starting December 2024, users in India, and other countries can start receiving MFA text messages via WhatsApp. Only users that are enabled to receive MFA text messages as an authentication method, and already have WhatsApp on their phone, get this experience. If a user with WhatsApp on their device is unreachable or doesn't have internet connectivity, we'll quickly fall back to the regular SMS channel. In addition, users receiving OTPs via WhatsApp for the first time will be notified of the change in behavior via SMS text message.

If you don't want your users to receive MFA text messages through WhatsApp, you can disable text messages as an authentication method in your organization or scope it down to only be

enabled for a subset of users. Note that we highly encourage organizations move to using more modern, secure methods like Microsoft Authenticator and passkeys in favor of telecom and messaging app methods. For more information, see: [Text message verification](#).

Retirement - MFA Fraud Alert will be retired on March 1st 2025

Type: Deprecated

Service category: MFA

Product capability: Identity Security & Protection

Microsoft Entra multifactor authentication (MFA) fraud alert allows end users to report MFA voice calls, and Microsoft Authenticator push requests, they didn't initiate as fraudulent.

Beginning March 1, 2025, MFA Fraud Alert will be retired in favor of the replacement feature

[Report Suspicious Activity](#) which allows end users to report fraudulent requests, and is also

integrated with [Identity Protection](#) for more comprehensive coverage and remediation.

To ensure users can continue reporting fraudulent MFA requests, organizations should migrate to using Report Suspicious Activity, and review how reported activity is remediated based on their Microsoft Entra licensing. For more information, see: [Configure Microsoft Entra multifactor authentication settings](#).

Public Preview - Microsoft Entra Health Monitoring, Alerts Feature

Type: Changed feature

Service category: Other

Product capability: Monitoring & Reporting

Intelligent alerts in Microsoft Entra health monitoring notify tenant admins, and security engineers, whenever a monitored scenario breaks from its typical pattern. Microsoft Entra's alerting capability watches the low-latency health signals of each scenario, and fires a notification if an anomaly is detected. The set of alert-ready health signals and scenarios will grow over time. This alerts feature is now available in Microsoft Entra Health as an API-only public preview release (UX release is scheduled for February 2025). For more information, see: [How to use Microsoft Entra Health monitoring alerts \(preview\)](#).

General Availability - Log analytics sign-in logs schema is in parity with MSGraph schema

Type: Plan for change

Service category: Authentications (Logins)

Product capability: Monitoring & Reporting

To maintain consistency in our core logging principles, we've addressed a legacy parity issue where the Azure Log Analytics sign-in logs schema didn't align with the MSGraph sign-in logs schema. The updates include fields such as ClientCredentialType, CreatedDateTime, ManagedServiceIdentity, NetworkLocationDetails, tokenProtectionStatus, SessionID, among others. These changes take effect in the first week of December 2024.

We believe this enhancement provides a more consistent logging experience. As always, you can perform pre-ingestion transformations to remove any unwanted data from your Azure Log Analytics storage workspaces. For guidance on how to perform these transformations, see:

[Data collection transformations in Azure Monitor](#).

Deprecated - MIM hybrid reporting agent

Type: Deprecated

Service category: Microsoft Identity Manager

Product capability: Monitoring & Reporting

The hybrid reporting agent, used to send a MIM Service event log to Microsoft Entra to surface in password reset and self-service group management reports, is deprecated. The recommended replacement is to use Azure Arc to send the event logs to Azure Monitor. For more information, see: [Microsoft Identity Manager 2016 reporting with Azure Monitor](#).

Phishing-resistant authentication in Microsoft Entra ID

Article • 03/04/2025

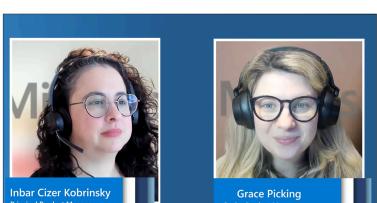
In this video series, we guide you through the basics of phishing-resistant authentication methods in Microsoft Entra ID. Multifactor authentication is one of the most effective controls to prevent an adversary from gaining access to sensitive information.

Watch the following videos, or visit the [Phishing Resistant Authentication in Microsoft Entra ID](#) video series, for guidance on how to configure phishing-resistant multifactor authentication methods.

To learn more, see [Windows Hello for Business](#), [certificate-based authentication](#), and passkeys formally [FIDO2](#) security keys. In addition, learn about [Microsoft Entra Conditional Access](#), which brings signals together to inform decision making, and enforces organizational policies.

[+] Expand table

Video title	Video
<p>Why phishing resistant authentication is important</p> <p>Join Alex Weinert, VP Identity and Network Access Security, in a discussion about the current threat landscape and why implementing phishing-resistant authentication should be a priority for your organization. (12:40)</p>	 Alex Weinert VP, Product Identity and Network Access Security
<p>Get started with phishing-resistant multifactor authentication</p> <p>Join Ehud Itshaki, Principal Product Manager, for a technical deep dive on how Microsoft Entra ID authentication methods achieve their phishing-resistant properties by implementing open standards. Learn about the meaning of phishing and real-time phishing-flows. (10:24)</p>	 Ehud Itshaki Principal Product Manager
<p>Phishing-resistant multifactor authentication methods available in Microsoft Entra ID</p> <p>Join Keith Brewer, Principal Product Manager, for an overview of the four phishing-resistant authentication methods available in Microsoft Entra ID. (5:13)</p>	 Keith Brewer Principal Product Manager

Video title	Video
<p>Windows Hello for Business and cloud Kerberos trust provisioning ↗</p> <p>Join Bailey Bercik, Senior Product Manager and Merill Fernando, Principal Product Manager to learn about Windows Hello for Business and how deployment is easier with the cloud Kerberos deployment model. (8:08)</p>	
<p>Configure Windows Hello for Business for passwordless authentication ↗</p> <p>Join Merill Fernando, Principal Product Manager, and Bailey Bercik, Senior Product Manager for a discussion about deploying the Windows Hello for Business cloud Kerberos model. (8:48)</p>	
<p>Configure Microsoft certificate-based authentication ↗</p> <p>Join Nick Wryter, Principal Product Manager, and Vimala Ranganathan, Principal Product Manager, to learn about certificate-based authentication (CBA) in Microsoft Entra ID, without the need to deploy, secure, and manage a federated identity provider, such as Active Directory Federated Services (ADFS). (6:53)</p>	
<p>Configure user experience in Microsoft Entra certificate-based authentication ↗</p> <p>Join Vimala Ranganathan, Principal Product Manager, for guidance on configuring Microsoft Entra certificate-based authentication (CBA), including the end-user experience. (4:55)</p>	
<p>Microsoft Entra Conditional Access authentication strength ↗</p> <p>Join Grace Picking, Senior Product Manager and Inbar Cizer Kobrinsky, Principal Product Manager, to learn about authentication strength and how it can help your organization enforce the use of phishing-resistant authentication. (13:24)</p>	
<p>Configure Conditional Access authentication strength policies ↗</p> <p>Join Inbar Cizer Kobrinsky, Principal Product Manager, and Grace Picking, Senior Product Manager for insights on how authentication strength works with Conditional Access. (6:53)</p>	
<p>Introduction to passkeys in Microsoft Entra ID ↗</p> <p>Join Product Manager Calvin Lui and Product Manager Mayur Santani as they discuss passkeys in Microsoft Entra ID and how to configure them. The Introduction to passkeys in Microsoft Entra ID ↗ video details passkeys for work. It describes how</p>	

Video title	Video
<p>passkeys can help organizations usher in a phishing resistant future (11:16).</p> <p>How to configure passkeys in Microsoft Entra ID ↗</p> <p>Join Product Manager Calvin Lui and Product Manager Mayur Santani as they discuss passkeys in Microsoft Entra ID and how to configure them. In the How to configure passkeys in Microsoft Entra ID ↗ video, Calvin and Mayur walk the user through how to set up passkeys and use them with an account. They walk you through registration and authentication experiences for passkeys in the Microsoft Authenticator app then passkeys on a FIDO2 security key (7:10).</p>	 <p>Mayur Santani Product Manager</p> <p>Calvin Lui Product Manager</p>

Feedback

Was this page helpful?

👍 Yes

👎 No

Provide product feedback ↗