2022

Conditional Access demystified whitepaper

Protecting your users Identity stored in Azure Active Directory and Company data stored in Microsoft 365 and other SaaS apps.

WHITEPAPER ON IMPLEMENTING AZURE ACTIVE DIRECTORY CONDITIONAL ACCESS

KENNETH VAN SURKSUM | WWW.VANSURKSUM.COM | ITGRATION

VERSION 1.4 | DECEMBER 2022

Contents

About: Kenneth van Surksum

Kenneth van Surksum works as a modern workplace consultant at Itgration and is specialized in building modern workplace solutions on top of Microsoft 365. Kenneth also works with public and private cloud solutions based on Azure and System Center. Kenneth is a Microsoft MVP for Enterprise Mobility.

Kenneth is co-founder of the Workplace Ninja User Group Netherlands and organizes (virtual) community meetings on a regular basis.

Kenneth loves to speak about technical topics related to his daily work. Kenneth is Microsoft Certified Trainer (MCT) and has multiple certifications, he has received the MVP and VMware vExpert award multiple times.

If you want to reach out to me, please use on of the options below:

Twitter: https://twitter.com/kennethvs
LinkedIn: https://www.linkedin.com/in/kennethvansurksum/
Email: kenneth@itgration.nl

Changelog:

Name & Version	Released	Comment
Conditional Access demystified-v1.0	August 2019	Initial version based on my series of blogposts on this topic
Conditional Access demystified-v1.1	May 2020	Updated cheat sheet, implementation workflow, documentation spreadsheet,
Conditional Access demystified-v1.2	February 2021	Big update, going from 30 to 77 pages including updated reference sheets. Describing some more scenario's and providing a set of default conditional access policies for anyone to use as a starting point.
Conditional Access demystified -v1.3	October 2021	Updated content and integration of several Conditional Access related blog articles
Conditional Access demystified – v1.4	November 2022	Updated content where needed. Added lots of new functionality, like Authentication Strength, granular Guest user options, Template policies, and making the policies available on Github to download and import.

1 Introduction

In July 2016 Microsoft made Conditional Access generally available as a feature of Azure Active Directory (AzureAD). Since that time, I had a love and hate relationship with this functionality of Azure AD. Mainly because it is difficult to test scenarios and some changes can have a really high impact. I even experienced being locked out of accessing the Azure portal during one of my tests.

1.1 Why this whitepaper?

There is already some good documentation from Microsoft and many blogposts by fellow bloggers detailing Conditional Access scenarios, but not really a one-stop shopping overview. With this whitepaper I hope to achieve this.

The paper combines several blogposts I wrote on this subject, where needed I will make references to those blogposts. I did my best to provide some structure in this paper, but it might be that you recognize that it exists of multiple blogposts tied together. I will do my upper best to correct that in new and future versions of this paper.

I will try to describe everything that I find important, and lessons learned while implementing Conditional Access in my own tenants and at customers. I will not go into much detail on creating individual Conditional Access policies, since that is both well documented by Microsoft and described by well-known bloggers on this subject like Peter van der Woude, Per Larsen, Alex Fields, Daniel Chronlund, Peter Daalmans, Thomas Naunheim, Christian Decker, John Craddock and many more.

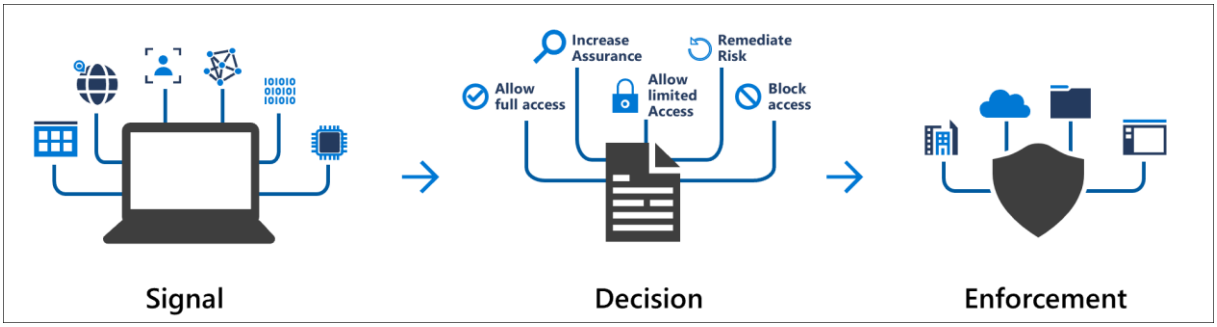
Microsoft is continuously adding functionality to Conditional Access, first functionality is added in a preview from which can be recognized by the (preview) tag in the name of the feature or Conditional Access policy and later it will eventually be released. The best way to keep up to date is by monitoring the Azure Updates webpage and the message center, where available, in preview and in development features of Azure Active Directory are shared.

2 What is Conditional Access?

Organizations are moving services which they traditionally hosted in their on-premises environment to the cloud. Because of this traditional network security in the form of Firewalls and other equipment no longer offers full protection when it comes to protecting company data.

Identity has become the new perimeter and is becoming the new attack surface for bad actors. So besides protecting our traditional assets which are either hosted on-premises or within an IaaS environment we also must protect the identity. Azure AD Conditional Access is one of the available methods to protect your identity.

Microsoft describes Conditional Access as following: "*With Conditional Access, you can implement automated access control decisions for accessing your cloud apps that are based on conditions.*" and "*Conditional Access policies are enforced **after** the first-factor authentication has been completed. Therefore, Conditional Access is **not** intended as a first line defense for scenarios like denial-of-service (DoS) attacks, but can utilize signals from these events (e.g. the sign-in risk level, location of the request, and so on) to determine access.*"



The way I see it, the best way to explain what Conditional Access does, is by making the comparison to a firewall. A firewall determines what traffic can access your resources, under what circumstances and Conditional Access sort of does the same. Conditional Access describes under what circumstances users can access (by acquiring an Access Token) your cloud data or applications. Keep in mind though that Conditional Access polices are enforced **after** the first authentication has taken place.

With cloud applications in Azure AD, Microsoft references to applications which use Azure Active Directory (Azure AD) for authentication (and sometimes also for authorization). Azure Active Directory (Azure AD) is Microsoft's cloud-based identity and access management service. So, if you access a SaaS based application through Azure AD you can use Conditional Access as your "access firewall". Microsoft's SaaS offerings like Office 365, Dynamics, the Azure Portal all use Azure AD as its authentication and authorization mechanism.

This also means that if you can access the SaaS application in some other way, you can bypass Conditional Access making the solution less effective. For example, you store only the credentials of the application in Azure AD, allowing you to use SSO to login to the applications and making use of Conditional Access, but when accessing the SaaS application directly you can also provide your userid and password, which allows you to bypass the Conditional Access policies. So, it is important that you modify the SaaS identity provider to only allow Azure Active Directory signins, and not accept local logins anymore (except for maybe a breakglass account in case of emergency).

If you have SaaS apps which have a federation with your ADFS infrastructure, you can use claim rules in ADFS which provide similar functionality as Conditional Access, if you want to make use of Conditional Access you need to modify the federation to use Azure AD instead of ADFS.

2.1 Licensing

Conditional Access is a feature which is part of an Azure AD Premium P1 and P2 license which you can buy individually or is part of a suite license like Enterprise Mobility and Security (EM+S) E3/E5 or Microsoft 365 E3/E5. Conditional Access is also included as part of Microsoft 365 Business Premium licensing as well, since that license includes Azure AD Premium P1.

Some of the Conditional Access settings require that you have licensed other products, for example in order to use the sign-in risk condition you need to have Azure AD Identity Protection licensed. (Part of Azure AD premium P2). In order integrate Conditional Access with Microsoft Defender for Cloud Apps (MDA), you must have MDA licensed as well.

Keep in mind, that you can apply Conditional Access policies to users which are not licensed for Azure AD Premium P1 and P2, since once you buy only one Azure AD Premium P1/P2 license the whole Azure AD is put in that modus. Even though this technically works, you are in conflict with the licensing terms. Microsoft states that ***if you use/benefit from a specific service within Azure, you must be licensed for it***.

Therefore, I advise to use group-based licensing and make sure that these groups are used for conditional access configuration as well.

2.2 Security defaults

If you are not licensed to use Azure AD Conditional Access, you can enable Security Defaults on your Azure AD tenant.

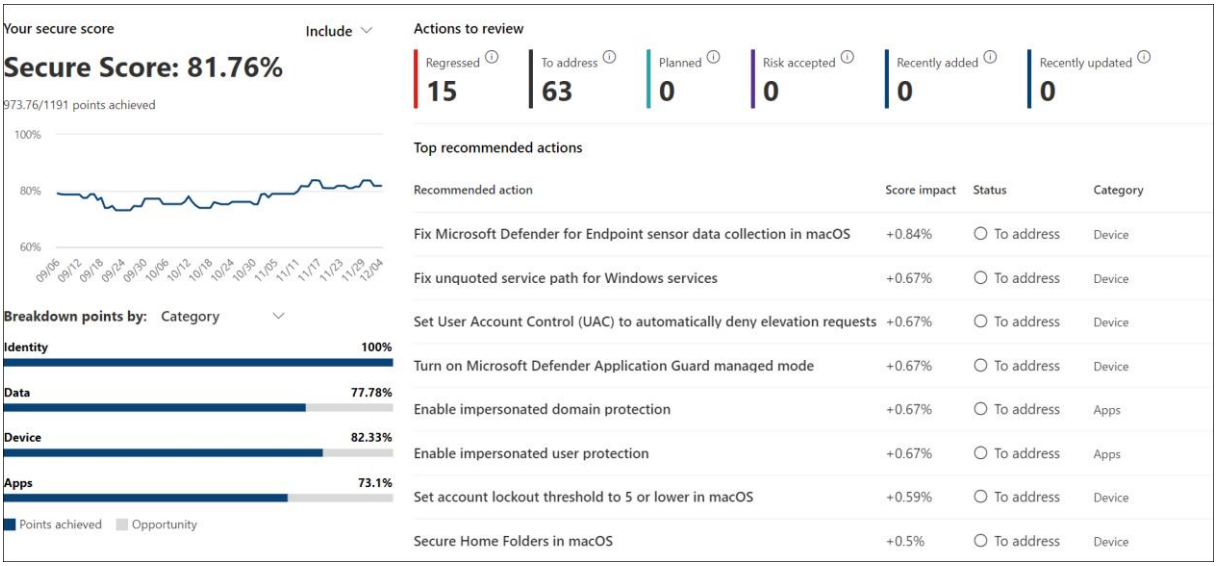
Microsoft explains the security defaults as following: "*Security defaults provide secure default settings that we manage on behalf of organizations to keep customers safe until they are ready to manage their own identity security story*." For now, when the security defaults are enabled the following security settings are enforced:

1. Requiring all users and admins to register for MFA.
2. Challenging users with MFA – mostly when they show up on a new device or app, but more often for critical roles and tasks.
3. Disabling authentication from legacy authentication clients, which can't do MFA.

Since security defaults are enabled for newly created tenants by default, they will provide a good security baseline for new customers, which is good news since many customers are still not using any form of MFA and have the "old" default option (which is nothing at all) enabled.

2.3 Secure score

Implementing Conditional Access policies can help with receiving points in Secure Score (https://security.microsoft.com/securescore). Secure Score provides a numerical summary of your security posture based on system configurations, user behavior and other security related measurements.

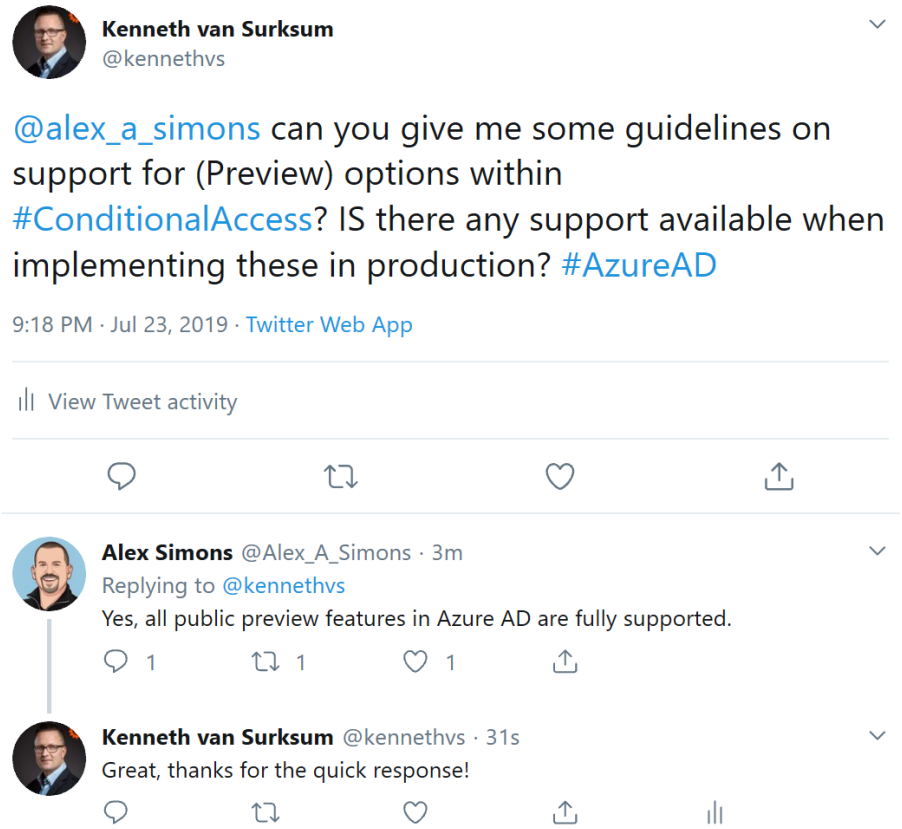


More information: Microsoft Secure Score

Conditional access provides functionality, which is part of a secure identity infrastructure, more actions than just implementing conditional access are needed. See this article for more information on the steps needed to secure your identity infrastructure: **Five steps to securing your identity infrastructure**.

2.4 What's up with the preview label?

Sometimes you will see the label "Preview" added to certain Conditional Access functionality. which made me wonder if enabling these policies is supported in production environment. I reached out to Alex Simons, who is Corporate Vice President PM for Microsoft's Identity Division, providing the following answer. "Yes, all public preview features in Azure AD are fully supported".



Microsoft also has an official statement on everything that is under the preview status, and what you can expect from a support point of view. You can find that information here: [Supplemental Terms of Use for Microsoft Azure Previews]

2.5 Legacy or basic authentication

When using Basic/Legacy Authentication application sends a username and password with every request to Exchange Online which either forwards the credentials towards Azure AD or a federated authentication provider like Active Directory Federation Services (ADFS). The problem with Basic/Legacy authentication is that it is vulnerable to brute force or password spray attacks. When legacy authentication is used, it is also possible to bypass Conditional Access.

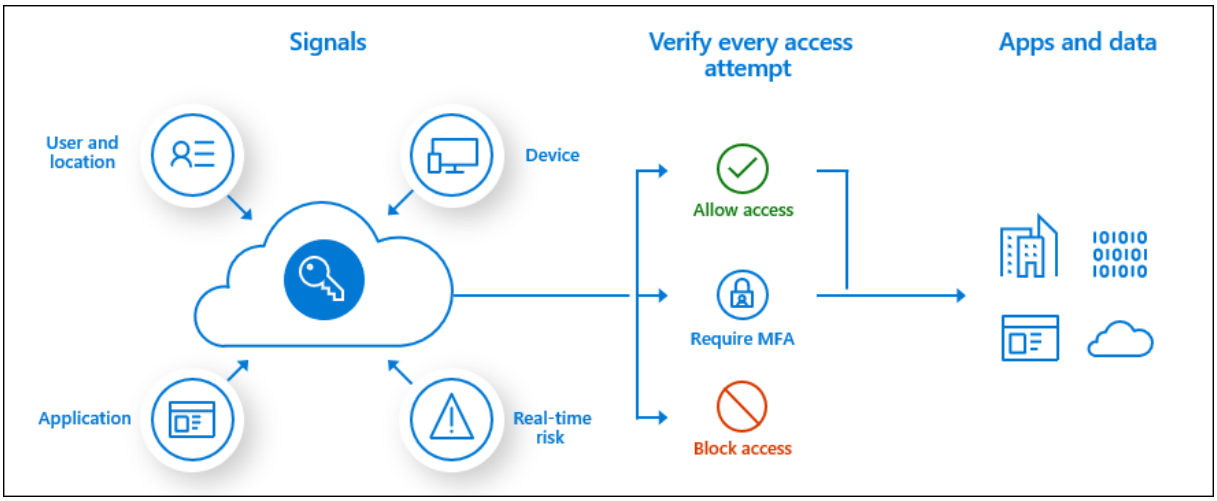
Modern Authentication is based on OAuth 2.0 and the Active Directory Authentication Library (ADAL) providing token-based authentication. OAuth 2.0 in this case is the protocol being used, and ADAL is used to authenticate against Azure AD.

On September 23, 2021, the Exchange Team announced that **effective October 1st, 2022** basic authentication, regardless of usage will be permanently disabled in all tenants. At time of writing, Microsoft is now busy disabling Legacy Authentication for Exchange Online in every Azure AD tenant, see the following article for more information: Basic Authentication Deprecation in Exchange Online – September 2022 Update - Microsoft Community Hub

I've written an extensive article on how to that which you can read here: Microsoft is going to disable basic/legacy authentication for Exchange Online. What does that actually mean and does that impact me?

3 How does Conditional Access work?

Conditional Access, when configured is evaluated whenever an Authentication or re-authentication request is made to Azure AD Conditional Access. Users can either sign-in **interactive**, where users provide an authentication factor such as a password and a response through for example MFA or biometric. Or users can sign-in **non-interactive**, where no interaction with the user takes place, for example when the user starts an application, like for example Outlook.



Microsoft explains Conditional Access in the following way.

Conditional Access consists of access scenario's called Conditional Access policies. An Conditional Access policy follows the following pattern:

When this happens	Then do this

"When this happens" defines the reason for triggering your policy. This reason is characterized by a group of conditions that have been satisfied. With "**Then do this**" you define how users can access your cloud apps.

Technically this is translated to Conditions (When this happens) and Access controls (Then do this)

Conditional access policy	
Conditions	Access controls

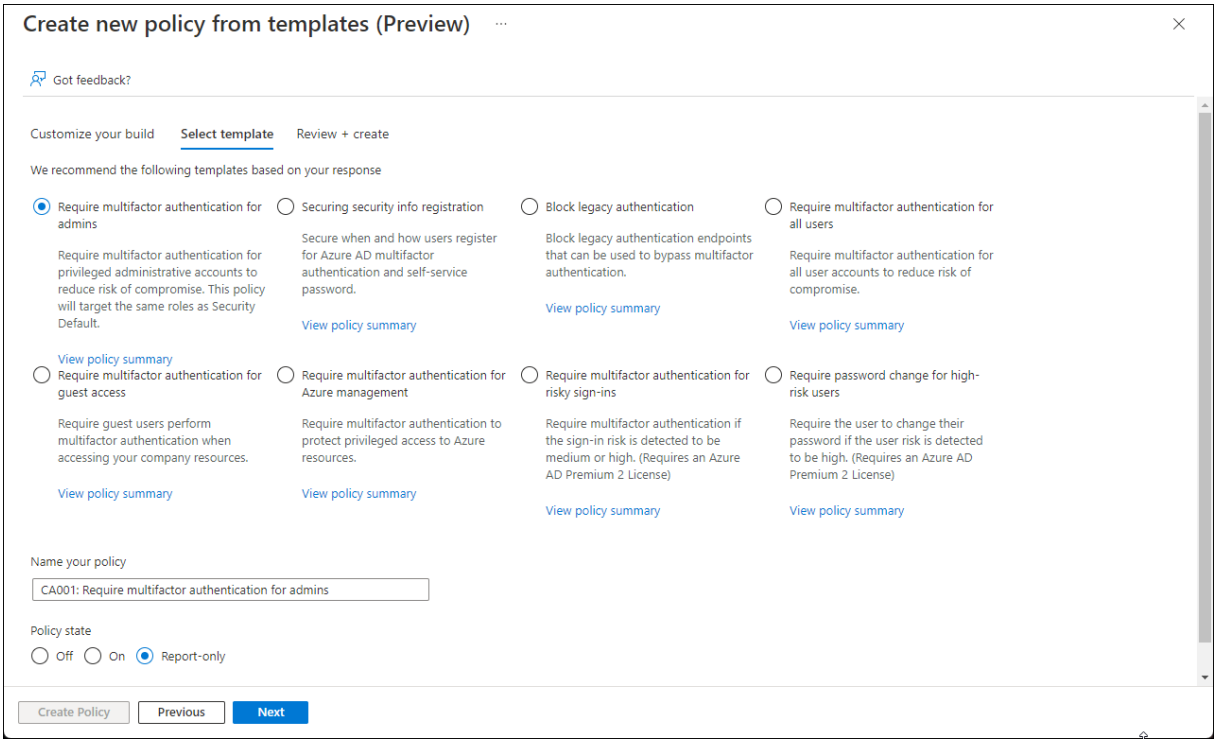
Microsoft provides some examples on their website for the most used Conditional Access policies, which you can use for reference. The following Conditional Access policies have been described:

1. Require MFA for administrators
2. Securing security info registration
3. Block Legacy Authentication
4. Require multifactor authentication for guest access
5. Require MFA for all users
6. Require MFA for Azure Management

7. Use application enforced restrictions for unmanaged devices
8. Require multifactor authentication for Intune device enrollments
9. Block access by location
10. Block access
11. Require an authentication strength for external users

3.1 Conditional Access templates

Microsoft currently provides (in Preview) the option to create their recommended policies by using the option to create a new policy from a template. Currently 14 policy templates are provides. You can use the templates by going to the Azure portal -> Azure Active Directory -> Security -> Conditional Access -> Create new policy from template.



From there you can select a template category, which is either Identities or Devices.

For Identities the following templates are available:

- **Require multifactor authentication for admins**

Require multifactor authentication for privileged administrative accounts to reduce risk of compromise. This policy will target the same roles as Security Default.
Suggested policy name: CA001: Require multifactor authentication for admins

- **Securing security info registration**

Secure when and how users register for Azure AD multifactor authentication and self-service password.
Suggested policy name: CA002: Securing security info registration

- **Block legacy authentication**

Block legacy authentication endpoints that can be used to bypass multifactor authentication.
Suggested policy name: CA003: Block legacy authentication

- **Require multifactor authentication for all users**

Require multifactor authentication for all user accounts to reduce risk of compromise.
Suggested policy name: CA004: Require multifactor authentication for all users

- **Require multifactor authentication for guest access**

Require guest users perform multifactor authentication when accessing your company resources.
Suggested policy name: CA005: Require multifactor authentication for guest access

- **Require multifactor authentication for Azure management**

Require multifactor authentication to protect privileged access to Azure resources.
Suggested policy name: CA006: Require multifactor authentication for Azure management

- **Require multifactor authentication for risky sign-ins**

Require multifactor authentication if the sign-in risk is detected to be medium or high. (Requires an Azure AD Premium 2 License)
Suggested policy name: CA007: Require multifactor authentication for risky sign-ins

- **Require password change for high-risk users**

Require the user to change their password if the user risk is detected to be high. (Requires an Azure AD Premium 2 License)
Suggested policy name: CA008: Require password change for high-risk users

For Devices the following templates are available:

- **Require compliant or hybrid Azure AD joined device for admins**

Require privileged administrators to only access resources when using a compliant or hybrid Azure AD joined device.
Suggested policy name: CA009: Require compliant or hybrid Azure AD joined device for admins

- **Block access for unknown or unsupported device platform**

Users will be blocked from accessing company resources when the device type is unknown or unsupported.
Suggested policy name: CA010: Block access for unknown or unsupported device platform

- **No persistent browser session**

Protect user access on unmanaged devices by preventing browser sessions from remaining signed in after the browser is closed and setting a sign-in frequency to 1 hour.
Suggested policy name: CA011: No persistent browser session

- **Require approved client apps and app protection**

To prevent data loss, organizations can restrict access to approved modern auth client apps with Intune app protection.
Suggested policy name: CA012: Require approved client apps and app protection

- **Require compliant or hybrid Azure AD joined device or multifactor authentication for all users**

Protect access to company resources by requiring users to use a managed device or perform multifactor authentication. (macOS or Windows only)
Suggested policy name: CA013: Require compliant or hybrid Azure AD joined device or multifactor authentication for all users

- **Use application enforced restrictions for unmanaged devices**

Block or limit access to SharePoint, OneDrive, and Exchange content from unmanaged devices.
Suggested policy name: CA014: Use application enforced restrictions for unmanaged devices

The templates give you the option to quick start your Conditional Access implementation. Be very careful implementing these policy directly though, they lack for example a best practise to exclude your break glass accounts, giving you no escape in case the CA policies lock you out of your environment. Best would be to implement the policies created from the templates in Report-Only mode for a while, and monitoring its applicability for a while before enabling them.

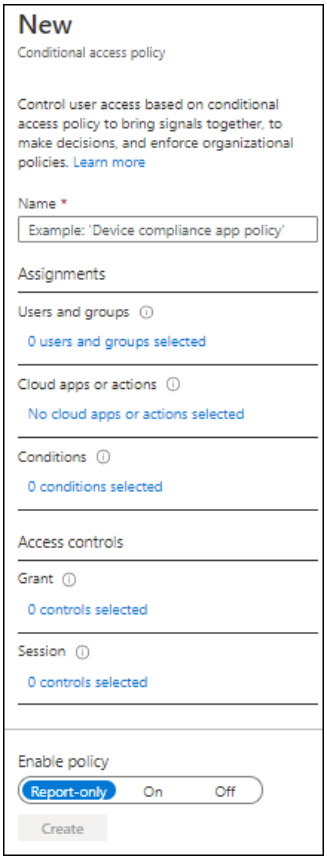
Note: at time of writing Microsoft announced a change in the way that they will deliver the Conditional Access templates. Even though the amount of policies will not change, they will be categorized differently in the portal, using the following scenario's:

- Secure foundation
- Zero Trust
- Remote work
- Protect administrators
- Emerging threats

See also: https://learn.microsoft.com/en-us/azure/active-directory/conditional-access/concept-conditional-access-policy-common. One new nice added feature is that they also allow you to export the Conditional Access policy into JSON format.

3.2 Custom Conditional Access policies

Besides some of the common policies optionally created by using the templates, customers can also create their own "custom" Conditional Access policies. The figure on the right shows how a new Conditional Access policy are grouped into sections. The Conditions (When this happens) are grouped as assignments, and Access controls (Then do this) are grouped as Access controls.

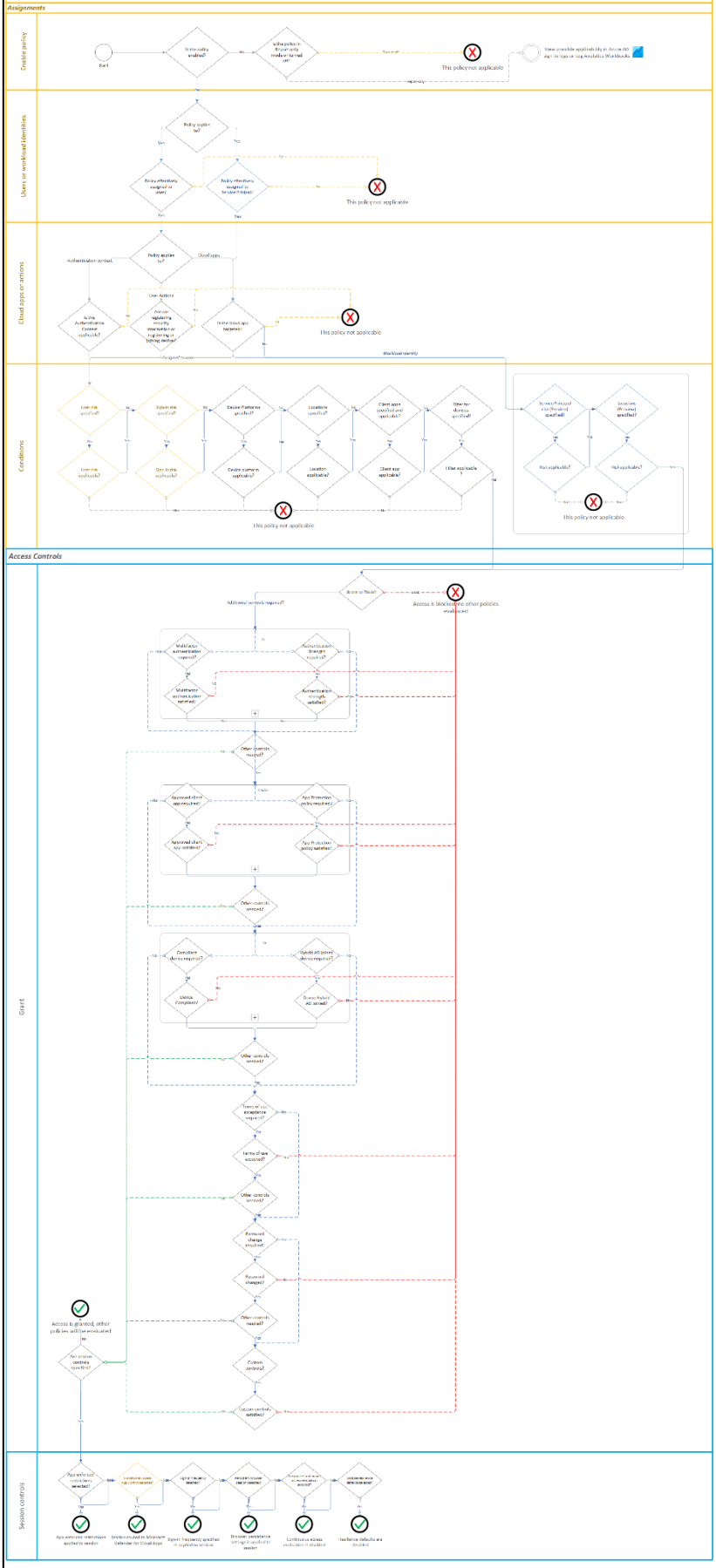


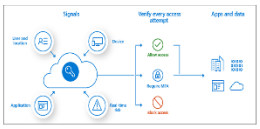
The Microsoft description covers Conditional Access from a high-level overview, practically Conditional Access is a little more complex as explained in the following flowchart or cheat sheet.

The flowchart takes you through all the steps of a Conditional Access policy, when there are multiple Conditional Access policies the steps are repeated for every policy which is applicable.

You can download this cheat sheet as PDF from the following location: [Conditional Access Workflow – v1.4.pdf]

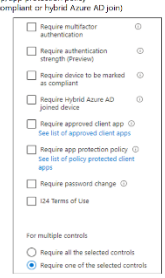
Conditional Access troubleshooting flowchart





Rules:
All policies are enforced in two phases:
In the first phase, all policies are evaluated and all access controls that aren't satisfied are collected.
In the second phase, you are prompted to satisfy the requirements you haven't met.
If one of the policies blocks access, you are blocked and not prompted to satisfy other policy controls. If none of the policies blocks you, you are prompted to satisfy either one or all selected policy controls in the following order. (see picture on the right)

1. Multi-factor authentication
2. Approved client app/app protection policy
3. Managed device (compliant or Hybrid Azure AD join)
4. Terms of use
5. Custom controls



External MFA providers and terms of use come next. All assignments are logically ANDed. If you have more than one assignment configured, all assignments must be satisfied to trigger a policy.

Block access thrumps all other configuration settings

Date: December 2022 | Version 1.4 | Author: Kenneth van Surksum | www.vansurksum.com

Based on the Conditional Access Workflow Cheat sheet, we can translate the most common used conditional access policies to the following formula:

Conditional Access = Access to <provided> Clouds Apps except <provided> Cloud apps by <provided> users and/or <provided> roles and/or <provided> groups except <provided> users and/or <provided> groups using <provided> User Risk and/or <provided> Sign-in Risk and/or <provided> Device Platform except <provided> Device Platform from <provided> Location except <provided> Location using <provided> Client apps with <provided> device state, except <provided> device state Grants, Grants but <provided requirement must be fulfilled> and/or applies Session controls or Blocks access.

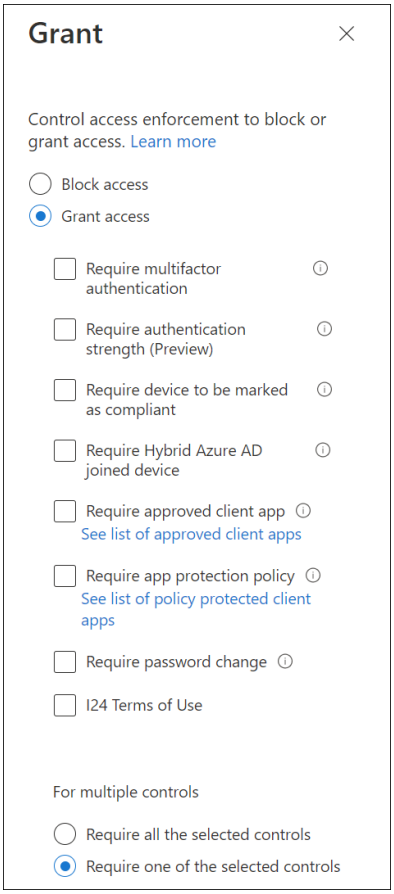
When multiple Conditional Access policies apply for a user when accessing a cloud app, **all the policies must grant access** before the user can access the cloud app.

Some important rules are:

1. All policies are enforced in two phases:

    - In the first phase, all policies are evaluated and all access controls that are not satisfied are collected.
    - In the second phase, you are prompted to satisfy the requirements you have not met.
    - If one of the policies blocks access, you are blocked and not prompted to satisfy other policy controls. If none of the policies blocks you, you are prompted to satisfy other policy controls, for which you require ALL or ONE of the selected controls, depending on the chosen option under "For multiple controls." If ONE is selected, the following order is used while evaluating.

    1. Multi-factor authentication
    2. Device to be marked as compliant
    3. Hybrid Azure AD joined device
    4. Approved client app
    5. App protection policy
    6. Password change
    7. Terms of use
    8. Custom controls



When using custom controls, your users are redirected to a compatible service to satisfy authentication requirements outside of Azure Active Directory. Custom controls have some limitation though, since they only work after a password has been entered and they cannot be used as MFA for step-up authentication for some scenario's. Therefore current external solutions using Custom Controls will be supported but in the meantime Microsoft is also working on a new implementation where partner provided authentication factors work alongside built-in factors including registration, usage, MFA claims, setup-authentication, reporting and logging.

2. All assignments are logically ANDed. If you have more than one assignment configured, all assignments must be satisfied to trigger a policy.
3. If one assigned Conditional Access policy blocks access, all other configuration is ignored and access is blocked. Therefore, be extra careful when creating block access policies.

A Conditional Access policy is built from the following components:

- General
- Assignments
- Access controls

The different components are further described in the following chapter.

4 Structure of a Conditional Access policy explained

In this chapter we are going to explain how a Conditional Access policy is setup, there are some general settings like name on whether the policy is turned on, in report-only mode or off. And we have the Assignments and Access controls.

4.1 General

The conditional access policy must have a unique name, use a name which gives an idea of what the policy is doing under what circumstances.

Microsoft recommends the following naming structure for your Conditional Access policies

Within the name the following information should appear.

- A Sequence Number
- The cloud app(s) it applies to
- The response
- Who it applies to?
- When it applies (if applicable)

My personal recommendation is to further extend this suggested naming convention and device the policies into categories and version them.

For sequence numbers I use categories with a 3 digit follow up number:

- CAPxxx = Conditional Access Prerequisite
- CAUxxx = Conditional Access User
- CADxxx = Conditional Access Device
- CALxxx = Conditional Access Location
- CACxxx = Conditional Access Custom

And I also version the Conditional Access policy, allowing me to release new versions and test those on a few users first.

An example of my proposed naming convention would be:

CAD007-O365: Session set Sign-in Frequency for Apps for All users when Modern Auth Clients and Non-Compliant-v1.0

Policies can either be enabled (On), disabled (Off) or defined in Report-only mode which can help to determine whether the policy is working as supposed to. More on Report-only mode in chapter 6, Implementing Conditional Access.

4.2 Assignments

Assignments define the "When this happens" part of the Conditional Access rule and consists of the following conditions.

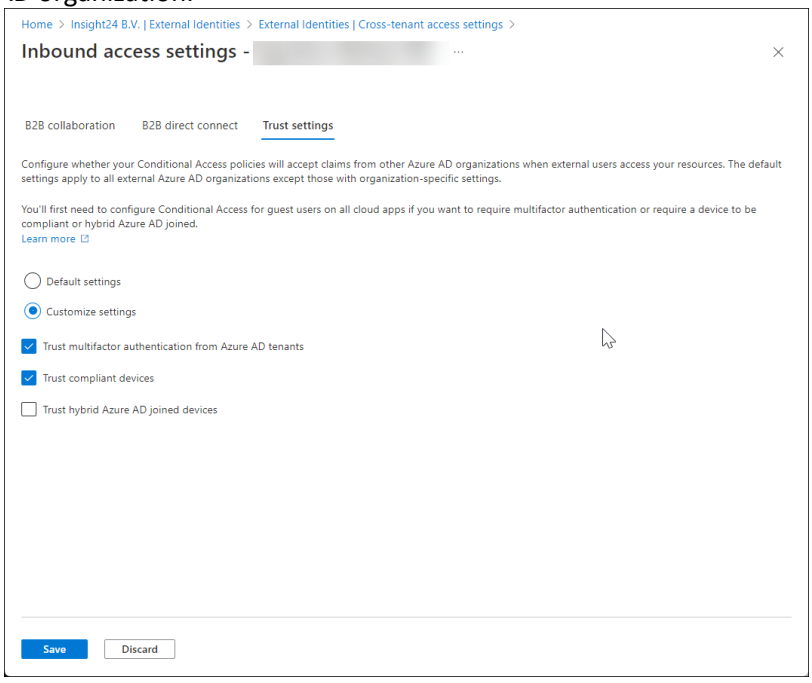
4.2.1 Users

In the users section you can specify to either apply the policy to Users and groups, or apply the policy to Workload identities. A workload identity is an identity that allows an application or service principal access to resources, sometimes in the context of a user. Conditional Access policies can be applied to single tenant service principals that have been registered in your tenant. Third party SaaS and multi-tenanted apps are out of scope. Managed identities aren't covered by policy.

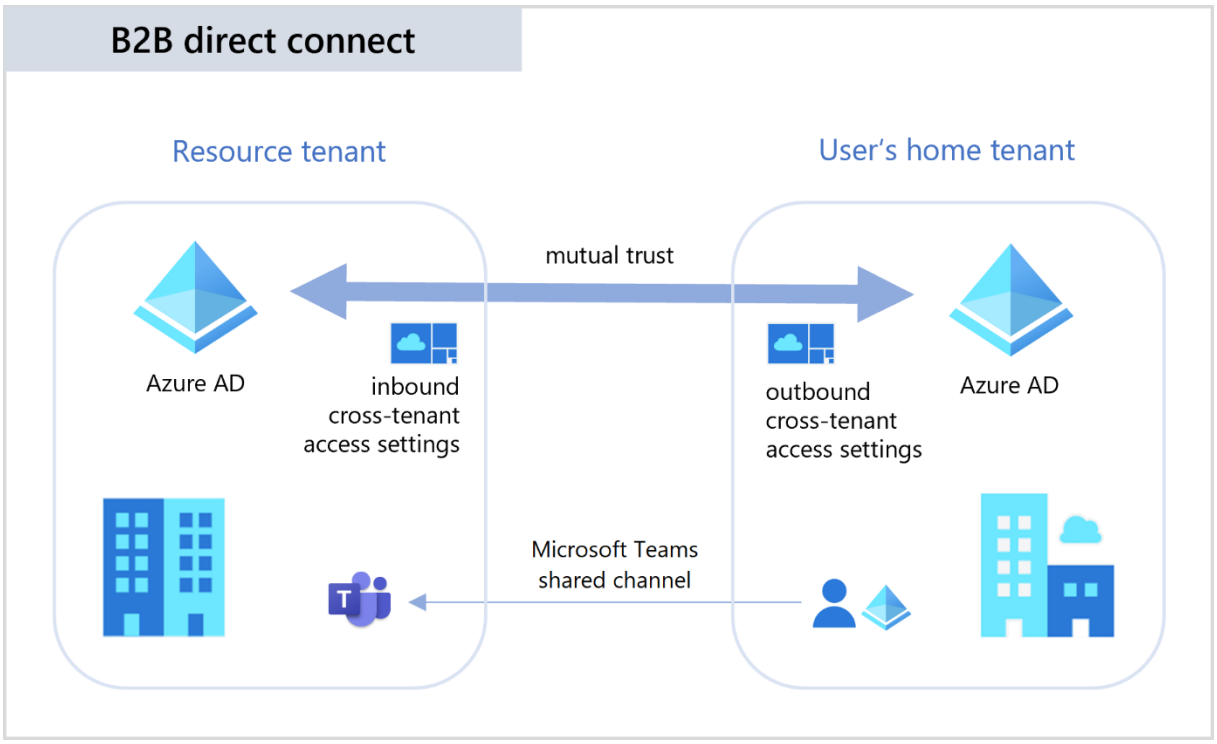
When applying the policy to users and group, you can specify whether you want to assign the policy to all users, or to a selection. Within the selection you can specify Guest or external users, Azure AD directory roles or specific users and groups. That same selection is also applicable if you want to exclude from the policy.

4.2.1.1 Guest or external users

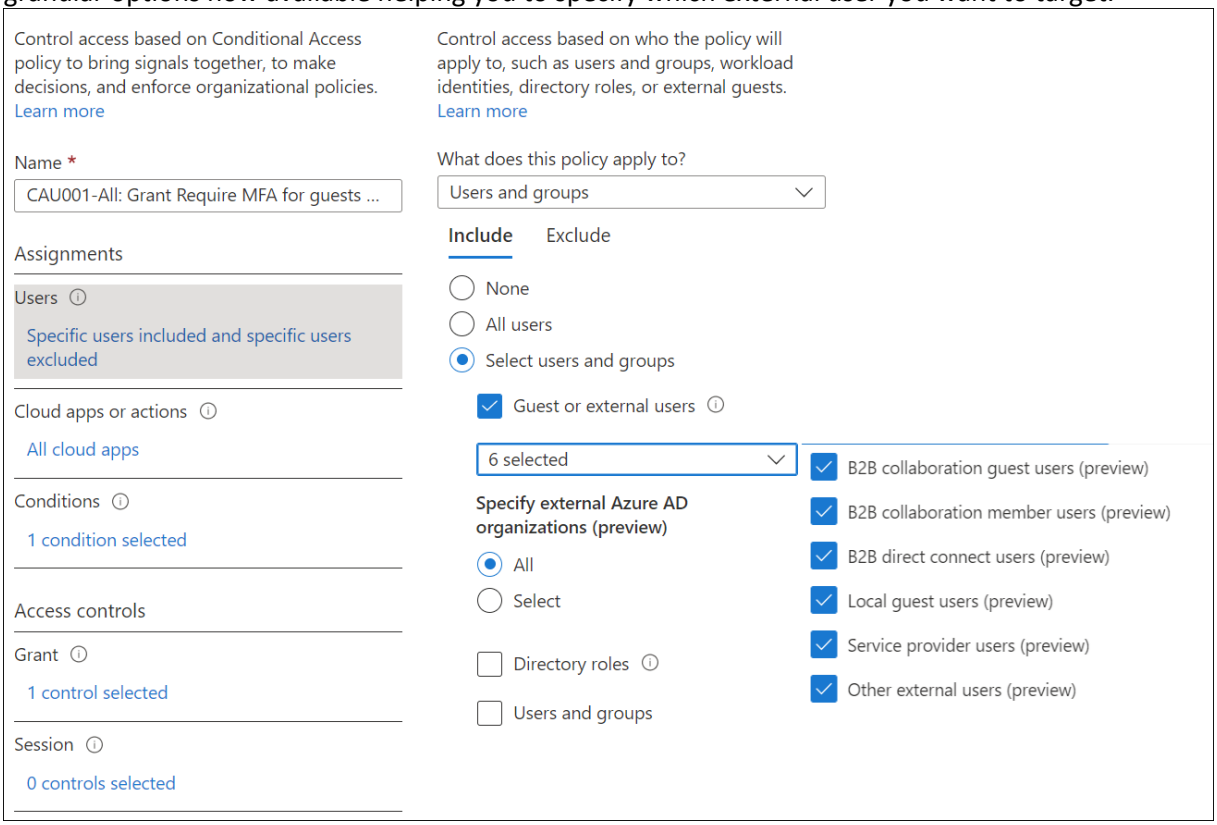
In July 2022, Microsoft made cross-tenant access settings for external collaboration generally available. With cross-tenant access settings we now have the option to setup a mutual "trust" with external Azure AD organizations, called B2B direct connect. While defining this "Trust" we can also decide whether we want to "trust" already performed MFA from a user in the other Azure AD organization, or if we want to "trust" compliant devices and/or Hybrid Azure AD joined devices from the other Azure AD organization.



The B2B direct connect makes the new "Shared channel" functionality in Microsoft Teams possible.



The addition of B2B direct connect introduced a new challenge when it comes to assigning Conditional Access policies to these new type of external users, which is addressed by the more granular options now available helping you to specify which external user you want to target.



Current Conditional Access policies targeting Guest or external users, have all options enabled, targeting all external Azure AD organizations.

Microsoft defines the following types of Guest or external users:

- **B2B collaboration guest users** - Most users who are commonly considered guests fall into this category. This B2B collaboration user has an account in an external Azure AD organization or an external identity provider (such as a social identity), and they have guest-level permissions in your organization. The user object created in your Azure AD directory has a UserType of Guest. This category includes B2B collaboration users who have been invited and who have used self-service sign-up.
- **B2B collaboration member users** - This B2B collaboration user has an account in an external Azure AD organization or an external identity provider (such as a social identity) and member-level access to resources in your organization. This scenario is common in organizations consisting of multiple tenants, where users are considered part of the larger organization and need member-level access to resources in the organization's other tenants. The user object created in the resource Azure AD directory has a UserType of Member.
- **B2B direct connect users** - External users who are able to access your resources via B2B direct connect, which is a mutual, two-way connection with another Azure AD organization that allows single sign-on access to certain Microsoft applications (currently, Microsoft Teams Connect shared channels). B2B direct connect users don't have a presence in your Azure AD organization, but are instead managed from within the application (for example, by the Teams shared channel owner).
- **Local guest users** - Local guest users have credentials that are managed in your directory. Before Azure AD B2B collaboration was available, it was common to collaborate with distributors, suppliers, vendors, and others by setting up internal credentials for them and designating them as guests by setting the user object UserType to Guest.
- **Service provider users** - Organizations that serve as cloud service providers for your organization (the isServiceProvider property in the Microsoft Graph partner-specific configuration is true).
- **Other external users** - Applies to any users who don't fall into the categories above, but who are not considered internal members of your organization, meaning they don't authenticate internally via Azure AD, and the user object created in the resource Azure AD directory does not have a UserType of Member.

The different types of guest or external users gives some nice insights in the options we have today to define guest or external users. So, while we would normally invite Guest users by adding them to a Microsoft 365 groups, we also have the option to designate those users as Members in our Azure AD. By doing so we can distinct between these type of users and target different Conditional Access policies against them. It's also interesting to see that you can designate Azure AD normal users as "Guest" user.

For B2B direct connect users, no accounts are created in your own Azure AD, just as for Service provider users which use Delegated Access Permissions (DAP)/ Granular Delegated Access Permissions (GDAP) to access your tenant in order to perform administrative activities.

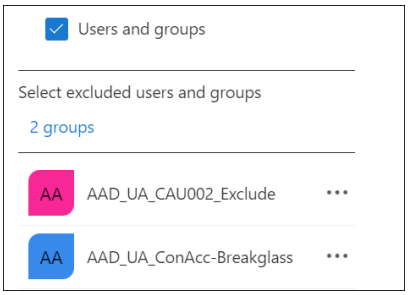
The "Other external users" can be more considered as a "catch all" just in case an external user doesn't meet any of the properties which define the other types of guest users.

4.2.1.2 Directory roles

Directory roles are especially interesting when using Azure Privileged Identity Management (PIM) where the Global Administrator role is assigned "temporarily" to a user instead of permanent. You

can for example have a more restricting Conditional Access policy applied while the user has activated the Global Administrator rights.

4.2.1.3 Users and groups



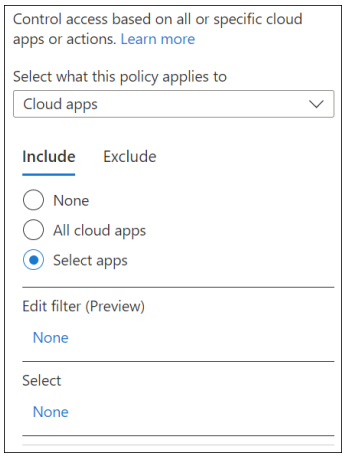
Make sure that when defining your users that you **always** exclude your break glass accounts, and create a unique group for each conditional access policy, using the sequence number in its name so that you can allow an exception on your policy.

Having this flexibility will help you, especially during initial rollout or migration to a new set of Conditional Access policies.

4.2.2 Cloud apps or actions

Withing Cloud apps or actions you can specify whether the policy applies to Cloud apps, User actions or Authentication Context.

4.2.2.1 Cloud Apps



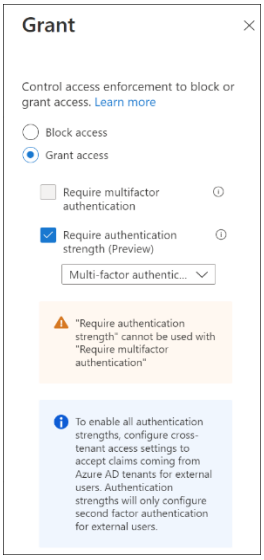
Microsoft defines a cloud app as a website, service or "endpoint protected by Azure AD Application Proxy". The supported cloud apps from Microsoft can be found in the following list: Microsoft cloud applications. Some of these cloud apps are Office 365, Office 365 Exchange Online, Office 365 SharePoint Online and Microsoft Azure Management (the Azure Portal). Besides the Microsoft applications you can also select any application which is configured as an Enterprise Application within your Azure AD.

The Office 365 app is a special application since it allows you to target all of the Office 365 services at once.

When specifying selected apps, you can either select the apps from a list, or use a filter. The filtering option is still in public preview and is described in more detail in "Conditional Access filters for Apps and Workload Identities"

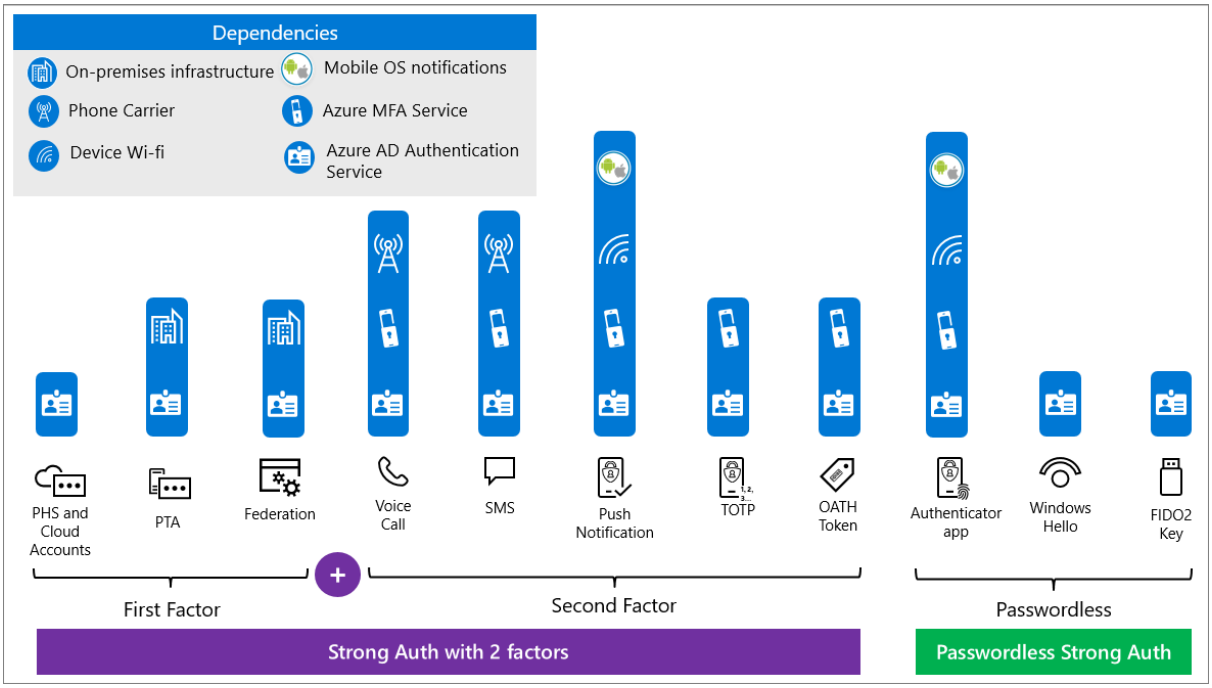
4.3 Authentication Strength

On October 19th, Alex Weinert the Director of Identity Security at Microsoft announced the public preview of authentication strength. Authentication Strength is a new Grant access control option available when you create or modify an existing Conditional Access policy.



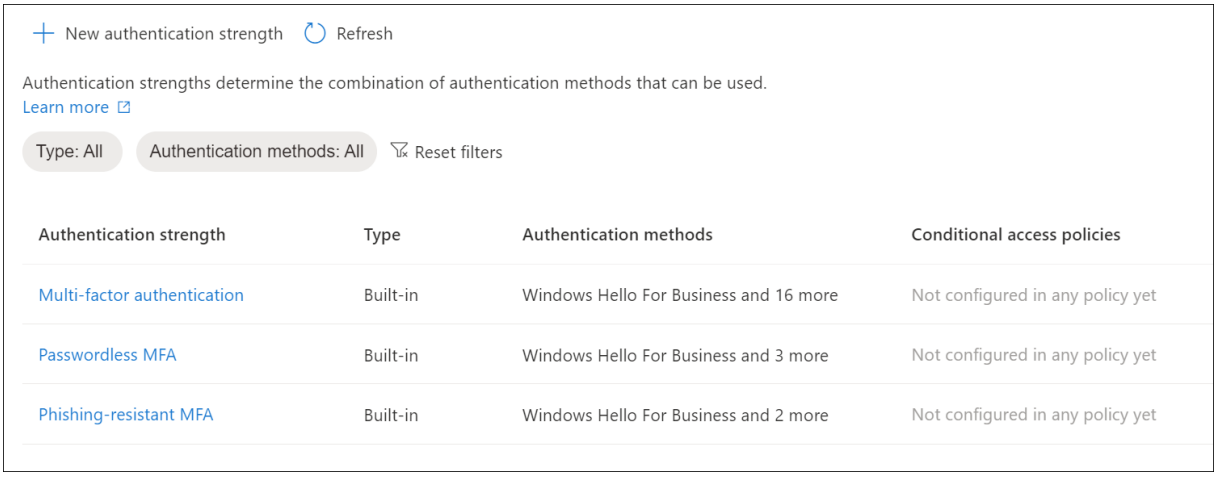
With Authentication Strength we have the option to distinct between the Multi Factor Authentication (MFA) method that can be used to fulfil the Access Control eventually granting access to the targeted resource app that you define in your conditional access policy.

Fact is today that not all MFA methods can be considered equally secure, and in my opinion customers should start moving away from these lesser secure authentication methods like SMS, phone call, but also only letting users allowing/denying an MFA request they receive in the authenticator app which gets exploited nowadays as well. If you want to know a bit more about this specific subject I want to suggest that your read this excellent article written by Jeffrey Appel: How to mitigate MFA fatigue and learn from the Uber breach for additional protection (jeffreyappel.nl)



As you can see from the screenshots below, you have the ability to choose between different Authentication strengths configurations. Three built-in configuration are provided by Microsoft, but you can also create your own. Let's go through the built-in ones first, which are:

- Multi-factor authentication
- Passwordless MFA
- Phishing-resistant MFA



4.3.1 Multi-factor authentication

Microsoft calls the Multi-factor authentication authentication strength a medium assurance authentication strength that includes multi-factor, for example password + SMS.

The Multi-factor authentication authentication strength when used allows the following methods:

- Windows Hello for Business
- FIDO2 Security Key
- Certificate Based Authentication (Multi-factor)
- Microsoft Authenticator (Phone Sign-in)
- 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
- Password + Voice
- Federated Multi-Factor
- Federated Single Factor + Microsoft Authenticator (Push Notification)
- Federated Single Factor + Software OATH token
- Federated Single Factor + Hardware OATH token
- Federated Single Factor + SMS
- Federated Single Factor + Voice

4.3.2 Passwordless MFA

Microsoft calls the Passwordless MFA authentication strength a high assurance authentication strength that includes methods with Cryptographic keys, for example FIDO2 security key.

The Passwordless MFA authentication strength when used allows the following methods:

- Windows Hello for Business
- FIDO2 Security Key
- Certificate Based Authentication (Multi-Factor)
- Microsoft Authenticator (Phone Sign-in)

4.3.3 Phishing-resistant MFA

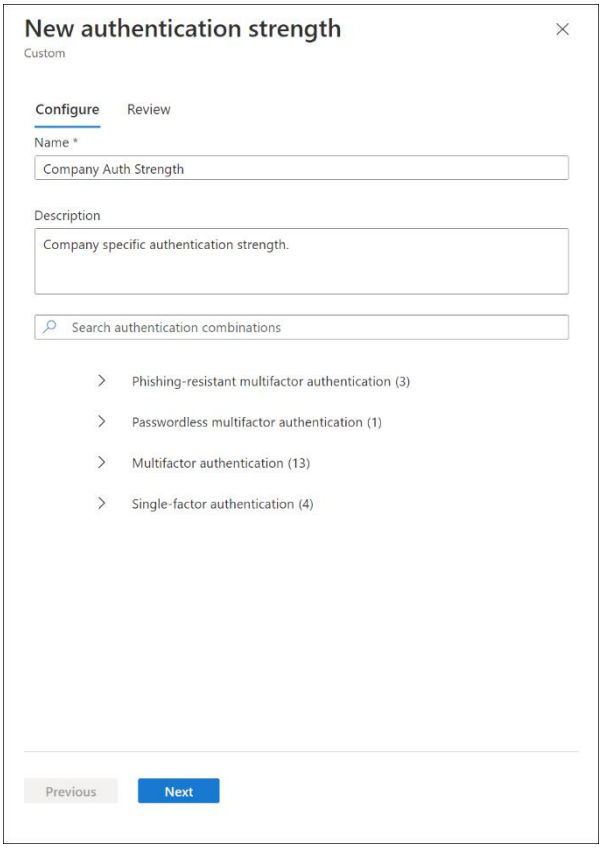
Microsoft calls the Phishing-resistant MFA authentication strength a method that is phishing resistant that includes methods like FIDO2 and Windows Hello for Business

The Phishing resistant MFA authentication strength when used allows the following methods:

- Windows Hello for Business
- FIDO2 Security Key
- Certificate Based Authentication (Multi-Factor)

4.3.4 Creating your own Authentication Strength Method

You can also define your own Authentication Strength method(s), by clicking on "+ Authentication Strength" which will start the New authentication strength wizard.



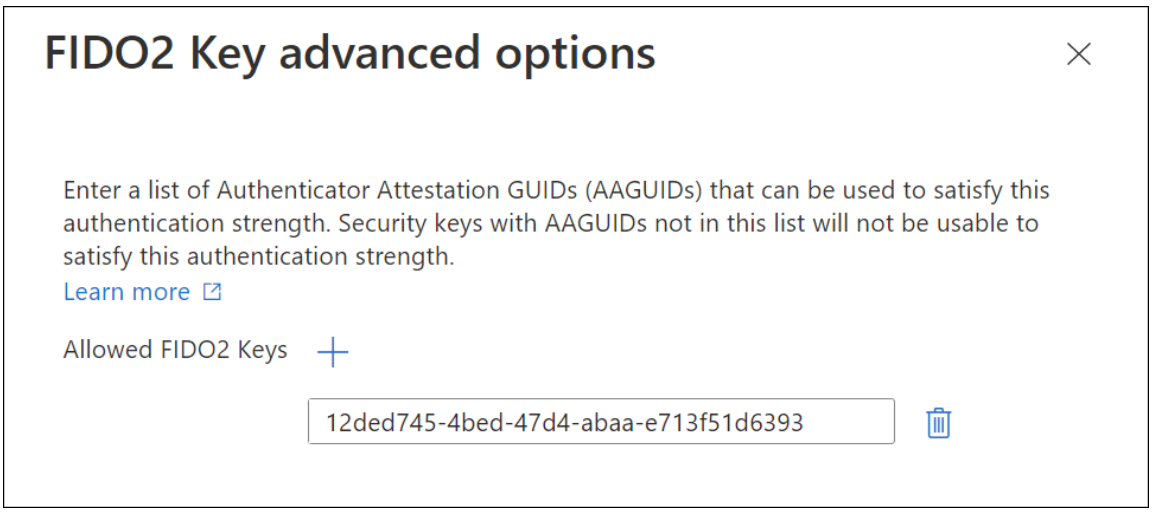
In the custom authentication strength wizard you can choose methods from the following categories:

- Phishing-resistant multifactor authentication
    - Windows Hello for Business

    - FIDO2 Security Key
    - Certificate Based Authentication (Multi-factor)
- Passwordless multifactor authentication
    - Microsoft Authenticator (Phone Sign-in
- Multifactor authentication
    - 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
    - Password + Voice
    - Federated Multi-Factor
    - Federated Single Factor + Microsoft Authenticator (Push Notification)
    - Federated Single Factor + Software OATH token
    - Federated Single Factor + Hardware OATH token
    - Federated Single Factor + SMS
    - Federated Single Factor + Voice
- Single-factor authentication
    - Certificate Based Authentication (Single Factor)
    - SMS
    - Password
    - Federated Single-Factor

4.3.4.1 Configure specific allowed FIDO2 keys

For the FIDO2 Security key you can even specify the allowed FIDO2 keys by specifying their Authenticator Attestation GUIDs (AAGUIDs). In the example below I'm configuring the Feitan AllinPass Fido 2 as allowed FIDO2 method



4.3.5 Authentication Strength use cases

The Authentication Strength option allows for all kinds of new scenario's which can be accomplished using Conditional Access.

Personally I would start with making sure that Administrative accounts can only sign in using Password-less MFA options, removing legacy MFA factors like SMS and Phone voice call if still in use

and you are not able to turn those off on the global level. Another interesting option is to combine Authentication Strength with Authentication Context, so that you can require a specific MFA method, when a SharePoint site with a specific sensitivity label gets accessed, or when a certain session control policy is triggered in Microsoft Defender for Cloud Apps (MDA).

https://www.vansurksum.com/2021/06/23/a-first-look-at-azure-ad-conditional-access-authentication-context/

We could also leverage authentication strength when user risk or sign-in risk as part of Azure AD identity protection is High. Another option is to use Authentication Strength in combination with cross-tenant settings as configured in Azure AD.

4.3.6 Authentication Strength in action

Before you start exploring Authentication Strength, make sure to read the known issues, which are documented here: Overview of Azure Active Directory authentication strength (preview) - Known issues.

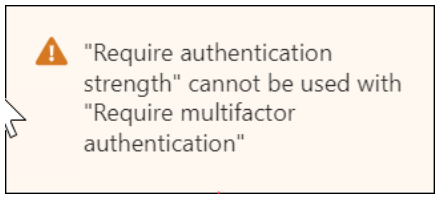
In my case, I tried building the following scenario:

See what happens if an Administrator role eligible user logs in with an authentication method, which isn't supported as an authentication method after elevating to its administrator role using Privileged Identity Management (PIM).

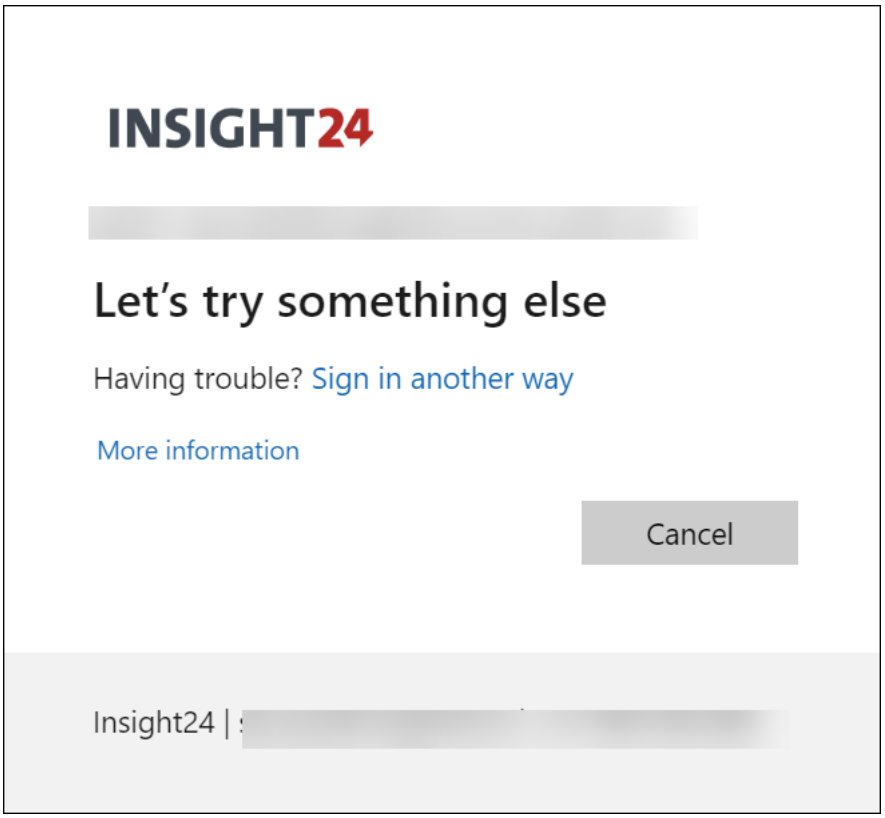
While building this solution I learned the following:

1. If for example SMS is turned off in the Global Settings, it's not available for the user to use if it's included in the Authentication Strength policy.

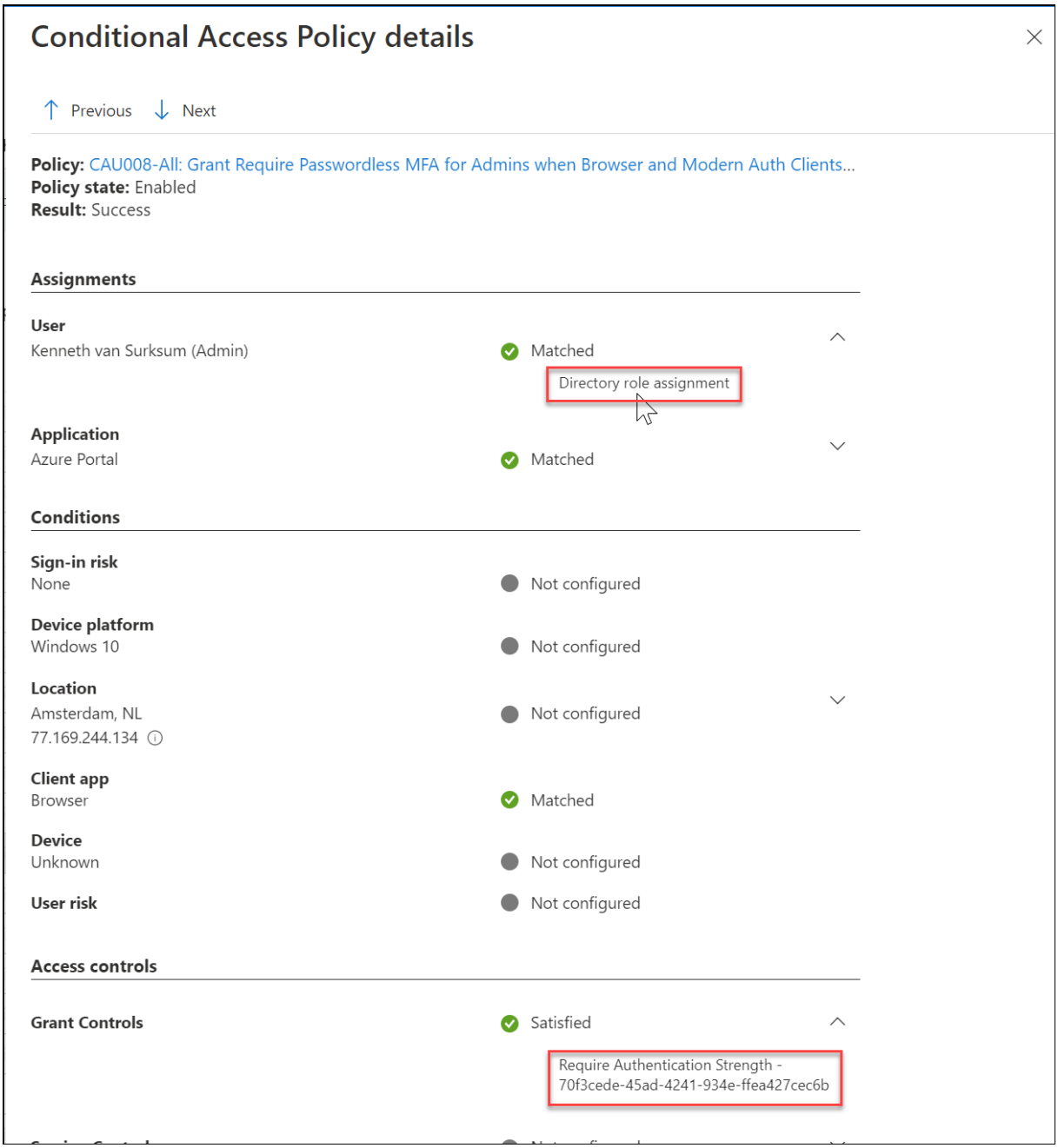
2. It's not possible to combine Multi Factor Authentication and Authentication Strength as grant controls in your Conditional Access policy, this is also noted when you create the policy.



3. You cannot combine Authentication Strength as a grant access control, while still having applicable Conditional Access policies using the "Multi Factor Authentication" access control. All applicable policies must use the Authentication Strength method, if one of them is using MFA you will receive a message similar like the one below.



4. When having multiple Conditional Access policies setting Authentication Strength, but with different authentication strengths set, the outcome can become unreliable. Say for example you have One CA policy with the "Multi-factor Authentication" authentication strength and another with the "Passwordless MFA" authentication strength, and both are applicable to the scenario, the outcome of the applied authentication strength can differ on a per scenario basis, making the use case unreliable.
5. I also had issues with using the Authentication App, when making the Passwordless MFA default option available to the CA policy requiring MFA for the Admin roles. In this case after elevating my rights, I was prompted for MFA but couldn't use the Phone sign-in which I configured on my admin account in the Microsoft Authenticator App, resulting in the "Let's try something else" method. I could use another method (my configured FIDO2 security key) successfully luckily. The only way for me to make this scenario work was to make the a new Authentication Strength policy and include the Password + Microsoft Authenticator (Push Notification) option as well.



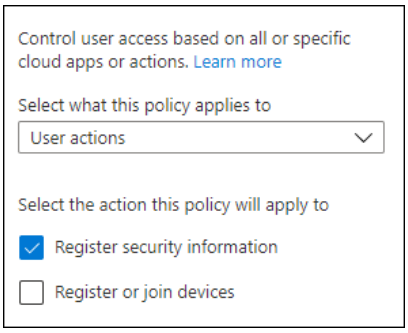
4.3.7 Authentication Strength Conclusion

Authentication Strength is a welcome addition to the Conditional Access Grant access control options already available and will eventually replace the current "Require MFA" option. Most ideally you want to configure your Conditional Access policies in such a way that all MFA required policies are enforced in the most secure way. Authentication Strength can help to setup a phased approach in order to move your users to these new secure ways of accessing your company apps and data, and provide you with options to enforce stronger authentication methods in certain use cases.

Be careful though with the current caveats causing authentication loops or , and thoroughly test your modified/new Conditional Access policies before deploying this new grant control.

Conditional Access filters for Apps and Workload Identities

4.3.7.1 User actions



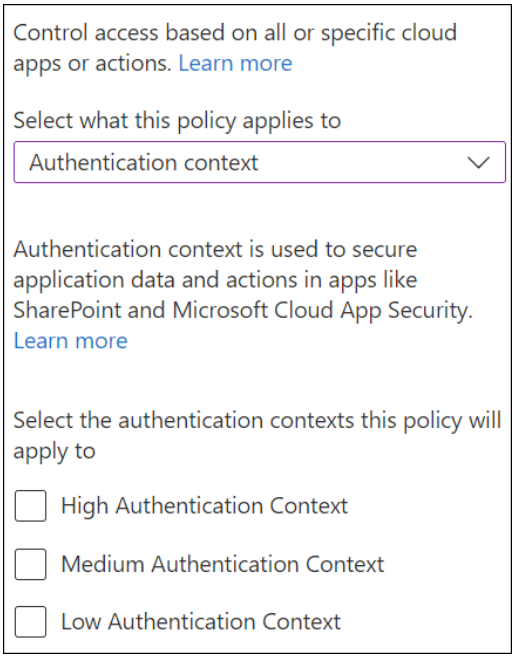
User actions refer to tasks a user can perform. For now, there are two user actions available:

- "Register security information", which requires the user to register security information needed to start using MFA. More information on that here: Combined security information registration for Azure Active Directory overview.
- Register or join devices, which is a newly introduced Conditional Access policy allowing to enforce Conditional Access during device join or registration (like asking for MFA). More Information here: Using Conditional Access to provide more granularity when registering or joining devices

Actions are a nice addition since they can help to make sure that prerequisites, like MFA are met, before enabling or applying other conditional access policies.

Keep in mind that if you do not use the User actions, you must select as least one cloud application for the Conditional access policy to work.

4.3.7.2 Authentication Context



Conditional Access Authentication Context can trigger a Conditional Access policy when sensitive content is accessed. With this new functionality new scenarios become available; some examples are:

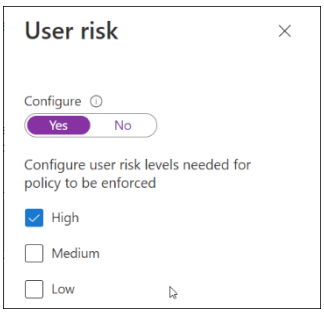
- You can require MFA when a SharePoint site with a certain sensitivity label gets accessed on an unmanaged device.
- You can block the session from a Conditional Access policy when a SharePoint site with a certain sensitivity label gets accessed on an unmanaged device.
- You can require that Terms of Use must be accepted first before access to a SharePoint site with a certain sensitivity label gets accessed.
- You can set the sign-in frequency to a low value when a SharePoint site with a certain sensitivity label gets accessed.
- You can trigger the Authentication Context from within a session control policy defined in MDA, for example to trigger MFA as defined in the Conditional Access policy. You could for example trigger MFA when a document with a certain sensitivity label gets downloaded or require the user to accept a Terms of Use first.
- App developers can leverage the authentication context in their own apps

More about Authentication Context at "Azure AD Conditional Access authentication context" later in this paper.

4.3.8 Conditions

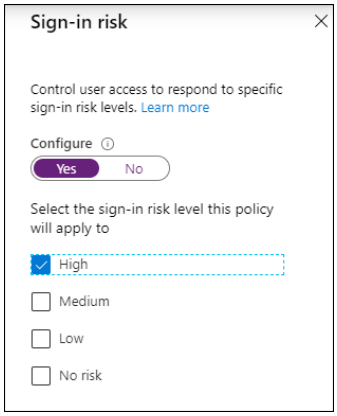
There are many conditions you can use, and Microsoft is sometimes adding even more available conditions depending on new functionality becoming available in Azure AD. Below are the current conditions described.

4.3.8.1 User Risk (additional license needed)



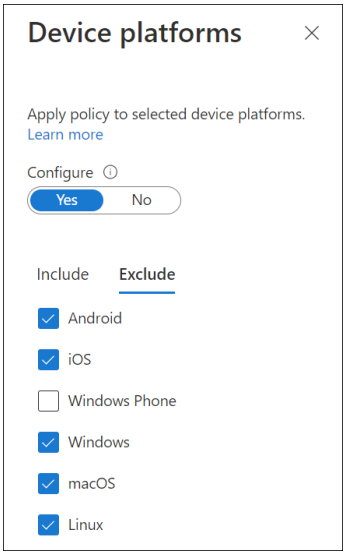
If you have licensed Azure Active Directory Identity Protection as part of Azure AD Premium P2 you can use this condition as a criterion to determine to which situation the conditional access policy will apply. Azure Active Directory Identity Protection will generate a so called "User risk level" and based on the level (High, Medium, and Low) you can make the conditional access policy applicable. More information about this scenario here: Conditional Access: Sign-in risk-based Conditional Access and in an article I wrote about the subject here: "Azure AD Identity Protection deep dive"

4.3.8.2 Sign-in Risk (additional license needed)



If you have licensed Azure Active Directory Identity Protection as part of Azure AD Premium P2 you can use this condition as a criteria to determine to which situation the conditional access policy will apply. Azure Active Directory Identity Protection will generate a so called "sign-in risk level" and based on the level (High, Medium, Low and No Risk) you can make the conditional access policy applicable. More information about this scenario here: Conditional Access: Sign-in risk-based Conditional Access and in an article I wrote about the subject here: "Azure AD Identity Protection deep dive"

4.3.8.3 Device Platforms



In the device platform condition, you can specify for which device platforms the policy is applicable. You can either include or optionally exclude device platforms from the condition. The following device platforms are available to select:

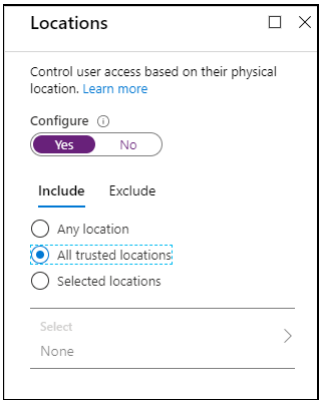
- Android
- iOS
- Windows Phone
- Windows
- macOS
- Linux

You can also include All platforms, where you also include the platforms not in the list above (unsupported platforms, like for example Linux) and then exclude a certain supported platform from the list above. You can use the device platform condition in the case that you want to restrict access to cloud apps from managed devices, but also if you need to create several conditional access policies when you want to implement a feature which is not supported on all device platforms.

Note: The device platform feature in Conditional Access is depending on user agent strings sent by the application or the web browser, which can easily be spoofed. This is something you must keep in

mind when designing your Conditional Access policies. See this article from Nicola Suter for more excellent information: Bypassing Conditional Access Device Platform Policies

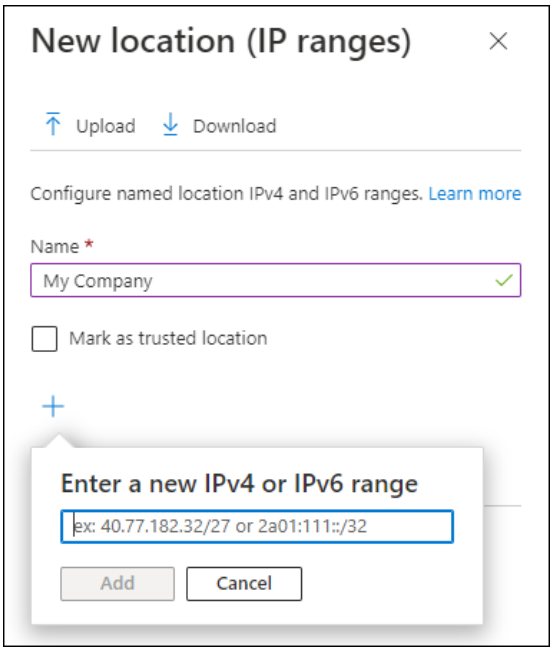
4.3.8.4 Locations



With locations you can specify conditions based on the network location the user is coming from, this is always the public IP address which is used on the internet and you cannot therefore use internal IP addresses to distinct in CA policies.

You can either include or exclude locations from the conditional access policy. Some use cases are that you want to restrict accessing the cloud app only from known locations (for example access to the Azure portal) or that you want to block access to a cloud app from a country or region for which you are sure your users will never use the cloud app service.

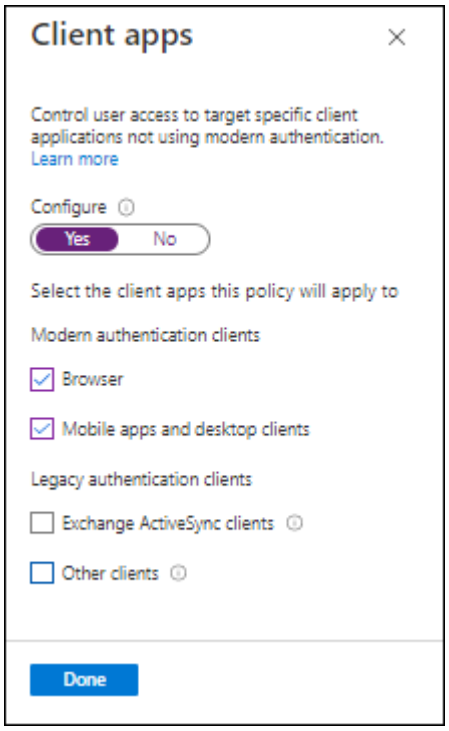
On December 7, 2022 Microsoft announced that it will bring IPv6 support to Azure Active Directory, this basically means that locations must be updated with IPv6 ranges as well, see: https://learn.microsoft.com/en-us/troubleshoot/azure/active-directory/azure-ad-ipv6-support



Note: Keep in mind that if within your company you provide guest network access but breakout to the internet using the same public IP address as your corporate devices, your guest network will fall under the same regime as your trusted network.

4.3.8.5 Client apps

Here you can specify the apps on the client for which the condition is applicable. These can be:



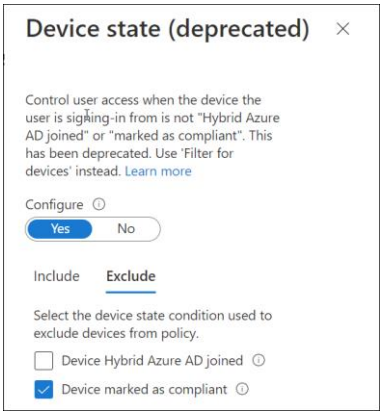
Modern Authentication Clients:

- **Browser** - apps accessed by a web browser on the client, see 4.2 for more information about restrictions.
- **Mobile Apps and desktop clients** - Applications supporting Modern Authentication

Legacy Authentication clients:

- **Exchange Active Sync clients** - apps which use Active Sync to connect to the cloud app - this option can only be selected if Exchange Online only is selected in the Cloud Apps selection. When Apply policy only to supported platforms is selected, only supported platforms like iOS, Android and Windows will be applicable.
- **Other clients** - apps which are not using "modern" authentication mechanisms, like IMAP, POP, SMTP etc... (for example, Outlook 2010). For more information about Modern and Legacy authentication see my article on that subject: "Microsoft is going to disable basic/legacy authentication for Exchange Online. What does that actually mean and does that impact me?"

4.3.8.6 Device state (deprecated)



When using the device state condition, you can exclude devices marked as compliant and devices which are Hybrid Azure AD joined (meaning Active Directory joined, and Azure AD registered) from the policy. Some scenarios are that you don't want the policy to apply to domain joined/azure ad registered devices, or that the managed device must report itself as compliant and if not, the policy will apply (block access for example until the device is compliant again)

Note: With Filter for devices now implemented, Microsoft recommends to move away from Device state and use Filter for devices instead. For new Conditional Access policies the device state is not available anymore.

4.3.8.7 Filter for devices

Filters for devices are available as conditions which you can use when creating your Conditional Access policies, with this functionality you can include or exclude devices based on filters using a rule expression. Combining include and exclude is not supported.

Microsoft provides the following examples where filters might provide a solution in their documentation:

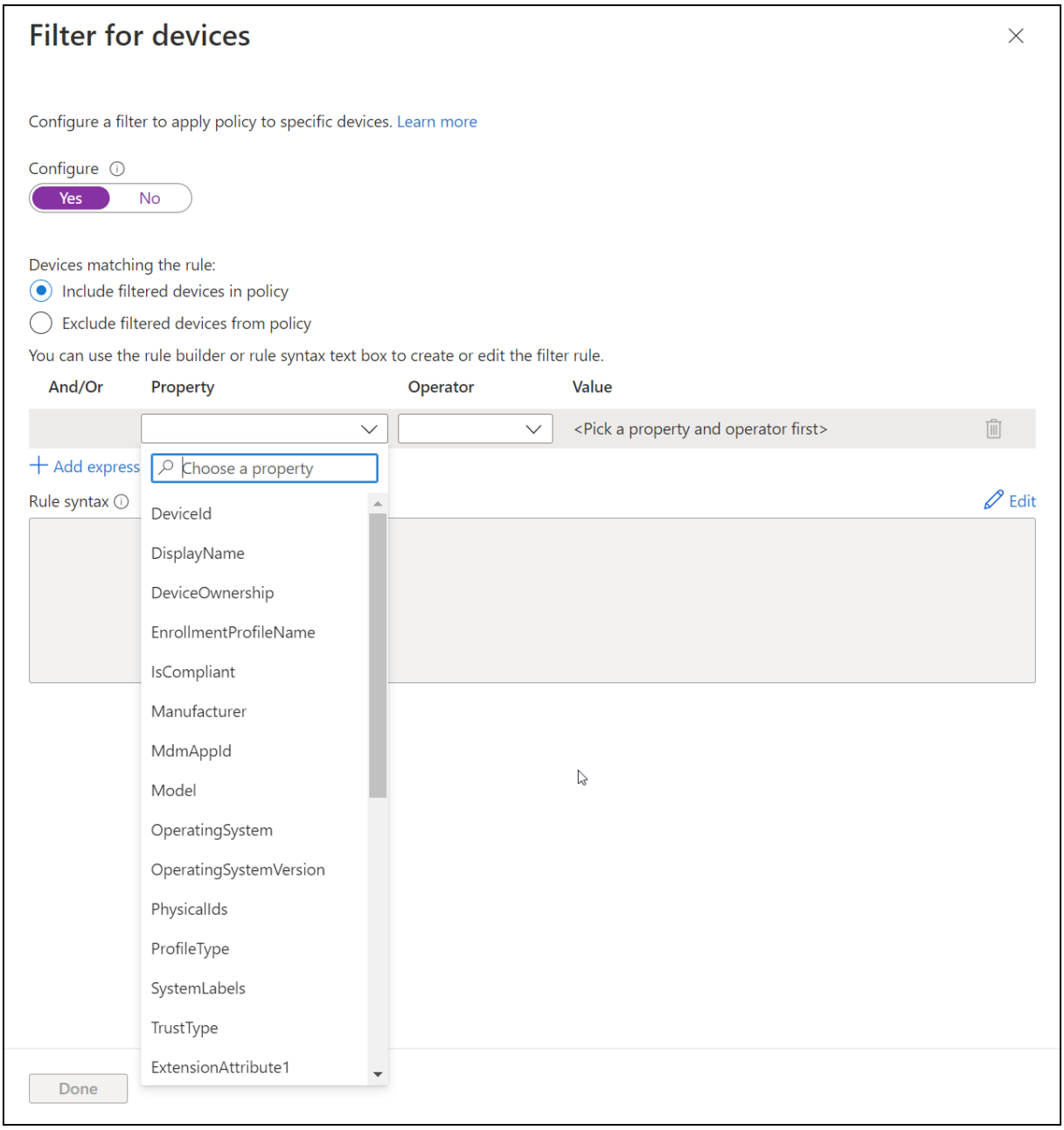
1. Give access to Azure Management for privileged users only, coming from privileged or secure admin workstations
2. Block access from devices running non supported Windows versions (like Windows 7, 8.1)
3. Do not require MFA for specific account (traditional AD service accounts) when used on specific devices, like Teams phones or Surface Hub devices

The documentation states that Device state (which allows you to exclude Compliant and/or Azure AD Hybrid joined devices) **and** Filters for devices cannot be used in one Conditional Access policy.
You can use the Compliancy and Azure AD Hybrid joined status in the Filter for devices as well though using the trustType and/or isCompliant properties, so basically this means that the Device State condition might disappear in the future to be replaced by the Filters for devices functionality.

Under the Filters for devices the following attributes can be used:
deviceId, displayName, manufacurer, mdmAppId, model, operatingSystem, operatingSystemVersion, physicalIds, profileType, systemLabels, trustType

Besides these attributes also custom attributes which can be set on the device in the form of extensionAttribute can be used, for this 15 custom fields can be used.

Each attribute supports operators, like Equals or Contains. But not every operator is supported for each attribute. The following operators are available: Equals, NotEquals, StartsWith, NotStartsWith, EndsWith, NotEndsWith, Contains, NotContains, In and NotIn



Create include or exclude filter

The following table, which I copied from the documentation give a very good idea of the available options for each attribute. The examples give an example of what's possible for each attribute.

Supported device attributes	Supported operators	Supported values	Example
deviceId	Equals, NotEquals, In, NotIn	A valid deviceId that is a GUID	(device.deviceid -eq "498c4de7-1aee-4ded-8d5d-000000000000")
displayName	Equals, NotEquals, StartsWith, NotStartsWith, EndsWith, NotEndsWith, Contains,	Any string	(device.displayName -contains "ABC")

	NotContains, In, NotIn		
manufacturer	Equals, NotEquals, StartsWith, NotStartsWith, EndsWith, NotEndsWith, Contains, NotContains, In, NotIn	Any string	(device.manufacturer -startsWith “Microsoft”)
mdmAppId	Equals, NotEquals, In, NotIn	A valid MDM application ID	(device.mdmAppId -in [“0000000a-0000-0000-c000-000000000000”]
model	Equals, NotEquals, StartsWith, NotStartsWith, EndsWith, NotEndsWith, Contains, NotContains, In, NotIn	Any string	(device.model -notContains “Surface”)
operatingSystem	Equals, NotEquals, StartsWith, NotStartsWith, EndsWith, NotEndsWith, Contains, NotContains, In, NotIn	A valid operating system (like Windows, iOS, or Android)	(device.operatingSystem -eq “Windows”)
operatingSystemVersion	Equals, NotEquals, StartsWith, NotStartsWith, EndsWith, NotEndsWith, Contains, NotContains, In, NotIn	A valid operating system version (like 6.1 for Windows 7, 6.2 for Windows 8, or 10.0 for Windows 10)	(device.operatingSystemVersion -in [“10.0.18363”, “10.0.19041”, “10.0.19042”])
pyhsicalIds	Contains, NotContains	As an example all Windows Autopilot devices store ZTDId (a unique value assigned to all imported Windows Autopilot devices) in device physicalIds property.	(device.devicePhysicalIDs -contains “[ZTDId]”)
profileType	Equals, NotEquals	A valid profile type set for a device. Supported values are: RegisteredDevice (default), SecureVM (used for Windows VMs in Azure enabled with Azure AD sign in.), Printer (used for printers), Shared	(device.profileType -notIn [“Printer”, “Shared”, “IoT”]

		(used for shared devices), IoT (used for IoT devices)	
systemLabels	Contains, NotContains	List of labels applied to the device by the system. Some of the supported values are: AzureResource (used for Windows VMs in Azure enabled with Azure AD sign in), M365Managed (used for devices managed using Microsoft Managed Desktop), MultiUser (used for shared devices)	(device.systemLabels -contains "M365Managed")
trustType	Equals, NotEquals	A valid registered state for devices. Supported values are: AzureAD (used for Azure AD joined devices), ServerAD (used for Hybrid Azure AD joined devices), Workplace (used for Azure AD registered devices)	(device.trustType -notIn 'ServerAD, Workplace')
extensionAttribute1-15	Equals, NotEquals, StartsWith, NotStartsWith, EndsWith, NotEndsWith, Contains, NotContains, In, NotIn	extensionAttributes1-15 are attributes that customers can use for device objects. Customers can update any of the extensionAttributes1 through 15 with custom values and use them in filters for devices condition in Conditional Access. Any string value can be used.	(device.extensionAttribute1 -eq 'SAW')

There are circumstances under which the Filter for devices conditions is applied or not applied depending on the Device registration state. So, it's important to realize that not everything that you build using the attributes and operators is supported. Microsoft describes these scenario in the following section of the documentation: Policy behavior with filters for devices.

It states for example that for unregistered devices, when using positive operators like Equals, StartsWith, EndsWith, Contains or In for any attribute the device filter is not applied, but when using the negative operators like NotEquals, NotStartsWith, NotEndsWith, NotContains or NotIn for any attribute the device filter gets applied.

Which while reading it from the documentation doesn't make sense, let's see if it makes more sense if we define an example. Let's use the attribute manufacturer for that.

So based on the documentation the filter is not applied when we define that the device manufacturer should start with Microsoft, but the filter is applied when the device manufacturer property doesn't start with Microsoft.

It can also mean though that the filter gets applied for negative operators on unregistered devices because the attribute cannot be determined. The means for example that in the case that the unregistered device is has the device manufacturer attribute of Microsoft and the operator is NotEquals Microsoft. the filter will apply because the attribute “device manufacturer” cannot be detected on the unregistered device. If this is the case, this is a caveat that must really be made clearer.

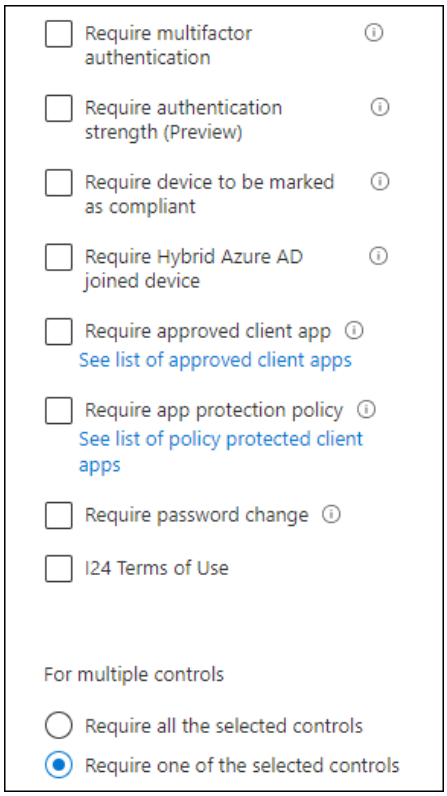
4.4 Access Controls

Access Controls define the "then do this" part of the conditional access policy. Based on the conditions the policy can:

- Block access to the cloud app
- Grant access to the cloud app
- Grant access to the cloud app but require an additional control (either one or all) from a list of selected controls (like MFA, must be compliant, Azure AD joined)

The grant access controls available are:

4.4.1 Grant



- Require multi-factor authentication (MFA)
  - Users are required to provide an extra authentication before access is granted
- Require authentication strength (Preview)
  - Allows you to define which MFA methods are supported, see “Authentication Strength” in the next chapter for more information on that subject.
- Require device to be marked as compliant
  - Device from which the user is accessing the cloud app must be managed and compliant
- Require Hybrid Azure AD joined device
  - Device is Hybrid Azure AD joined, meaning member of Active Directory, and registered in Azure AD
- Require approved client app
  - Approved client apps are apps which can be managed using MAM functionality in Intune, for a list of supported apps see: Supported mobile applications and desktop clients
- Require app protection policy
  - Here you can specify that besides the fact that the application must be capable of being managed using MAM, that also app protection policies must have been applied.
- Require password change
  - Usually used in combination with a policy for high risk users as part of Azure AD Identity Protection.
- Optional: Terms of use, or other custom controls

Microsoft want to move towards App Protection policy checks, but this requires the App developers to incorporate a newer version of the Intune SDK into their product.. Therefore, we now see that some applications are supported under the Require App Protection Policy part, where these are not listed as Approved Client App. On the other hand, having the require Approved Client App functionality available could be handy if you only have Azure AD P1 and no Enterprise Mobility + Security license (giving you MAM capabilities).

The Microsoft documentation currently raises a lot of questions, when trying to figure out my options I decided to create a matrix comparing the options available:

- Intune Protected Apps, the Official list that Microsoft references.
- Approved Client App, the list of supported Apps when using a CA policy with Grant access controls
- Require App Protection policy, the list of supported Apps when using a CA policy with Grant access controls.
- Applications listed under either iOS/iPadOS or Android when creating an App Protection Policy. (Keep in mind that some apps can be added using their custom id)

Based on this comparison, the only Apps with full support on all points are:

- Microsoft Edge
- Microsoft Excel
- Microsoft Lists
- Microsoft Office
- Microsoft OneDrive
- Microsoft OneNote
- Microsoft Outlook
- Microsoft Planner
- Microsoft Power BI
- Microsoft PowerPoint
- Microsoft SharePoint
- Microsoft Teams
- Microsoft Word

Because of this, we can only create a Conditional Access policy requiring either an Approved App, or require an App Protection policy targeting Office 365, you can build Conditional Access policy targeting specific Cloud Apps, if their corresponding iOS and/or Android app is either a Approved Client App, or on the Require App Protection Policy list.

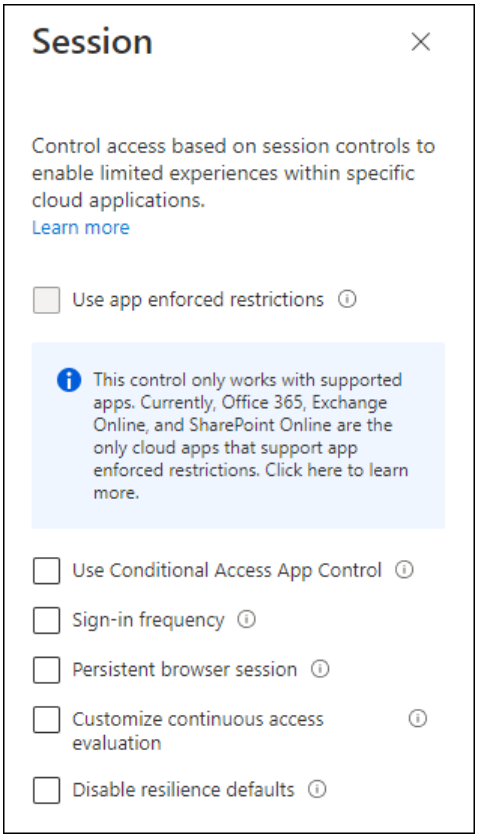
The Office 365 Cloud app enforcement is reflected in CA policy CAD003 which is described later. Downside of this, is that we cannot use CA to enforce the usage of apps to access a certain cloud app. If in our App Protection Policy, we do include the non-supported apps from a Approved App/Require App Protection Policy point of view, they will be able to communicate with the CA enforced apps though.

4.4.2 Session Controls

Session controls are also part of the Access controls and will be applied after the session is granted, they allow you to specify the experience within a cloud app and have the following options:

- Use app enforced restrictions

When this option is enabled, Conditional Access passes the device information to the cloud app, for now only SharePoint Online (SPO) and Exchange Online (EXO). In the cloud app a limited or full experience is offered depending on the device information.



- Use Conditional Access App Control

Routes the session through Microsoft Defender for Cloud Apps (MDA), which protects data by applying access and session controls acting as a reverse proxy. Some examples are: Prevent data exfiltration, protect on download, prevent upload of unlabeled files, and monitor user session for compliance. The options available here are: Monitor only (preview), Block downloads (preview) and Use custom policy.

- Sign-in frequency

With sign-in frequency you can specify the period before a user is asked to sign in again when attempting to access a resource. You can either choose Hours (between 1 and 23) or days (between 1 and 365) or you can choose that users must reauthenticate every time.

- Persistent browser session

A persistent browser session allows users to remain signed in after closing and reopening their browser window. With this control, which can only be set when all cloud apps are selected you can choose between Always persistent or Never persistent. Never persistent requires the user to login again after the browser window is closed.

- Customize continuous access evaluation

- Disable resilience defaults

Resilience defaults are automatically enabled for all new and existing policies, and Microsoft highly recommends leaving the resilience defaults enabled to mitigate the impact of an outage. Admins may disable resilience defaults for individual Conditional Access policies.

4.4.3 App enforced restrictions

App enforced restrictions are only supported with Exchange Online and SharePoint Online. When enabled for SharePoint Online, users without a compliant device will see the following when accessing SharePoint via a web browser. See "Control access from unmanaged devices" for more information.

If a browser session to Exchange Online is configured to use App enforced restrictions, what can be done in Outlook Web Access can be restricted, for example offline mode and downloads can be restricted. The restrictions are defined in a so called OWA Mailbox Policy which can be set using

PowerShell. See "Conditional Access in Outlook on the web for Exchange Online" for more information on how to configure the Mailbox policy.

For more information about how to use Conditional Access App Control, please read the following article on my blog: "Extending Conditional Access to Microsoft Defender for Cloud Appsusing Conditional Access App Control"

4.4.4 Control Azure AD Conditional Access policy behavior during an Azure AD outage

Microsoft announced that per April 1, 2021 they updated their service level agreement(SLA) for Azure AD user authentication from 99.9% to 99,99%. While this might seem like a small update in reality it makes a difference of 473 minutes (in a year with 365 days).

With 99.9% the allowed downtime was 525 minutes and with 99.99% this is reduced to 52 minutes. This is still inconvenient though if you experience downtime for let's say 45 minutes on Monday morning while everyone is starting their work.

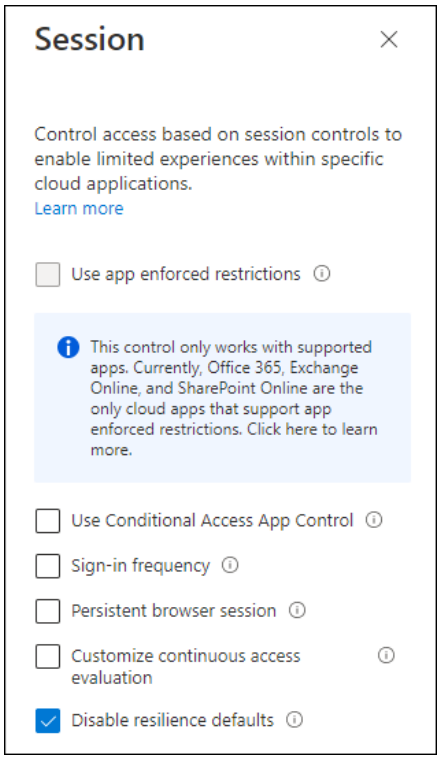
One of the techniques that Microsoft rolled out in order to support the new uptime, was the Azure AD Backup Authentication service, which Microsoft describes as the following:

Azure AD Backup Authentication service runs with decorrelated failure modes from the primary Azure AD system. This backup service transparently and automatically handles authentications for participating workloads as an additional layer of resilience on top of the multiple levels of redundancy in Azure AD. You can think of this as a backup generator or uninterrupted power supply (UPS) designed to provide additional fault tolerance while staying completely transparent and automatic to you.

So, what the Azure AD backup authentication service basically does is issuing tokens to applications for existing sessions if there is an outage of the primary authentication service. New sessions, or authentications by guest users are not supported.

Which brings us to Azure AD Conditional Access, since access tokens are re-evaluated by Conditional Access policies before issued. But when the Backup authentication service is used, not all conditions can be evaluated in real time.

By introducing a new session control in Conditional Access, called "Disable resilience defaults" it's now possible to let the policy block authentications in the case that the backup authentication service is active. By default this option is turned off, which means that "in this case" when conditions cannot be evaluated in real time or evaluated with data collected at the beginning of the user's session that the authentication will not be blocked.



The following conditions cannot be re-evaluated during an outage:

- Group Membership
- Role Membership
- Sign-in risk
- User risk

- Country location

That means that all other conditions, like MFA can be evaluated and those policies will not be impacted by this session. If you want to block based on Sign-in/User risk and don't want to make concessions even during an outage, then this option is for you.

5 Designing your Conditional Access policy using a strategy

When you are designing a Conditional Access strategy we first need to start with an inventory of the environment, in the most ideal situation you would design and implement conditional access in a green field scenario, but I for sure never had that luxury before so it is better to assume that the customer is already using cloud apps and wants to implement conditional access as a security measure.

The points to be inventoried are (but not limited to):

- What kind of devices does the customer use to access cloud apps?
    - Are the devices company owned, and fully managed?
    - Are the devices user owned, and non-managed?
- What kind of applications are currently used to access cloud apps?
- Is this a green field implementation, or are the cloud apps already in use without any conditional access policies in action?
- Does the customer use Intune and which scenarios are built into Intune?
    - Mobile Device Management
    - Mobile Application Management
- Is every user treated equally when it comes to access to the cloud apps, or can we distinct personas with different requirements when it comes to Conditional Access?
- Which licensing is the customer using? You need E5 functionality for administrators at least nowadays.
- How are licenses being assigned to users (groups, directly)
- Are there any service accounts used that interact with the cloud apps?
- Is Modern Authentication already enabled for Exchange Online and Skype for Business online?
- Is the company storing password hashes in Azure Active Directory?
- Are there cloud apps depending on each other?

When it comes to licensing while administrating Microsoft 365 services, please be aware that there are some things you need to be aware of, see also this article on my blog: "License requirements for administering Microsoft 365 services"

Microsoft has a document available helping in planning setting up Conditional Access, called the "Azure Active Directory Conditional Access Deployment Plan". The document in word format can be downloaded from the following location: https://aka.ms/CADPDownload. Microsoft also provides planning documentation online at: Plan a Conditional Access deployment - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/plan-conditional-access

When designing a Conditional Access strategy in my experience it is important to really think on a high level on what you want to accomplish. It is very easy to start creating Conditional Access all kinds of individual Conditional Access policy and get lost from what you wanted to accomplish along the way.

Based on my experience the main goal of implementing Conditional Access is that you want to prevent access to your company data in situations where you do not have control over the data. That means that ideally cloud apps can only be accessed by:

1. Devices which are under company control and are compliant.
2. Applications which are under company control and are compliant.
3. Browser sessions on managed devices where data can be stored locally.
4. Browser sessions on non-managed devices where data can only be opened in the browser session and no data is left behind on the device.

All the other scenario's possible are either to fulfill requirements to successfully use Conditional Access or are additional security measures like always enforcing MFA when Azure AD administrators log in. It might also be that you need some "temporary" conditional access policies while migrating to the designed situation.

Below are some example scenario's which can influence the outcome of your design.

- Scenario 1: Allow devices managed by Intune access all the cloud apps using Apps and Desktop Clients and Modern Authentication Clients if compliant

Access to "All Cloud Apps" by "Users with EMS License" using "Any" device platform" coming from "any location" using "Mobile Apps and Desktop Clients" or "Modern authentication clients" is allowed, but device must be compliant.

- Scenario 2: Only allow Apps we can manage to access cloud apps when device is not managed.

Allow users with EMS License using devices not managed by Intune to access (portion of) cloud apps, using clients which we can manage using MAM policies (approved clients list)

- Scenario 3: Allow browser access to all the cloud apps from a trusted location

When users access the cloud apps from a trusted location, they can login without using any additional form of authentication.

- Scenario 4: Allow browser access to all the cloud apps from an untrusted location but use MFA and restrict the browser session (when possible)

When users access the cloud apps from a non-trusted location, they can login but have to use MFA and when possible, the browser session is restricted.

- Scenario 5: Block browser access to all the cloud apps from some geographic areas

Users cannot access cloud apps from regions where the company does not operate.

Once you know your scenario's trying to model the conditional access policy in a spreadsheet, by doing this you can determine if policies can be combined, or if more than one policy needs to be created to meet the requirements of the scenario. Keep in mind that the less is more.

5.1 Service dependencies

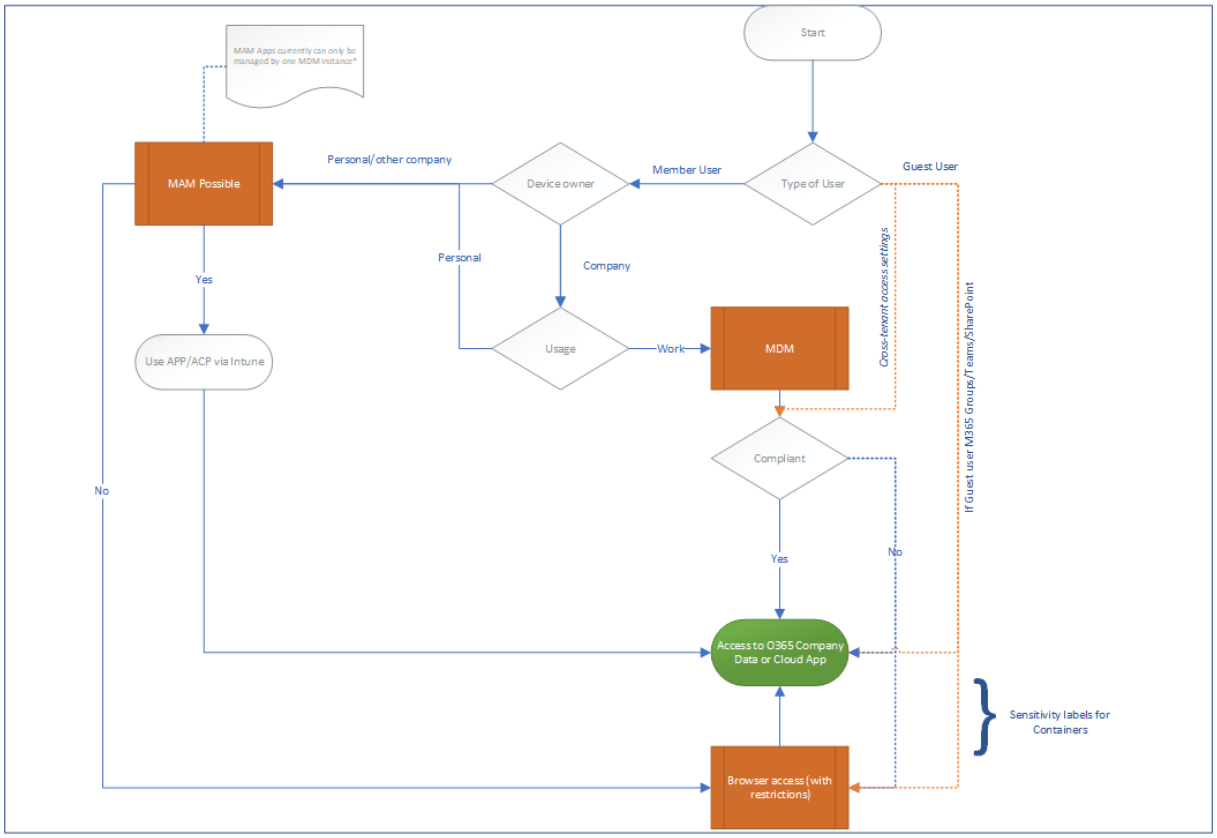
Many cloud apps have dependencies to other cloud apps, Microsoft Teams is a good example since it also provides access to SharePoint Online, and Planner for example. When this situation occurs, you must know how the application will behave, since policies may be applied either early-bound or late-

bound. See the following article with more information about this: "What are service dependencies in Azure Active Directory Conditional Access?"

I've created a spreadsheet which can hopefully help you to document and write down your conditional access policies, the spreadsheet is available for download from the following location: Conditional Access Policy Description-v1.2.xlsx

5.2 Functional Design

The functional flowchart below gives an overview of what I want to accomplish when designing an Conditional Access environment, I always use this flowchart and adjust it where needed to determine the use cases which must be supported.



As you can see, the flowchart is situated around giving access to company data but also about protecting the identity of the user, since I think that Data and the Identity are the assets you must protect using Conditional Access policies.

So basically, we support several scenarios in this flowchart, let me describe some of them:

5.2.1 Guest users

In a default Microsoft 365 configuration, each user can invite Guest users into your Azure Active Directory. This is mostly done by either sharing a specific file hosted on OneDrive/SharePoint with that Guest user, or by inviting that Guest user to a Microsoft Teams environment, where that Guest user can participate in the team.

With Conditional Access policies we can control how Guest users can access the environment. The access scenario's are also applicable for your own users working on non-compliant devices. Which setting is being used is either determined by your Global SharePoint settings, on a per SharePoint site basis, or by the Sensitivity label applied to the M365 group governing the access to the SharePoint site.

The options we have are:

5.2.1.1 Allow full access to the environment.

When you allow full access to the environment (which is the default), Guest users can use Desktop applications to access the data hosted by your company. In teams they can switch to your tenant and work in the teams that they are member of just like an internal user. They can even setup synchronization of SharePoint sites and have the data in that site available on their device.

You should ask yourself If you want to allow this, this totally depends on the data you are sharing with these guest users, in my opinion, if that data is confidential, you should not allow that data to be one a device which you not manage. It is also advised to implement a Governance procedure for Guest users and clean up occasionally, because without that Guest users can keep unlimited access the files shared and the Teams/SharePoint sites they have access to.

5.2.1.2 Allow Browser/Browser restricted access to the environment.

We can also disable access via Desktop clients and offer browser-based access only. In this way we have some better control since we can apply App Enforced Restrictions to the browser session and by doing so denying users the ability to download and print any company data. I've described this scenario in the following article: Limit Access to Outlook Web Access, SharePoint Online and OneDrive using Conditional Access App Enforced Restrictions. We can have the restrictions on all the data, or on just a part of the data by using the sensitivity label functionality for containers.

5.2.1.3 Block access

We can also decide to block access to the environment, if for example it's content is highly sensitive.

5.2.2 Device Owner and usage

When it comes to device owner and usage, we have several options:

5.2.2.1 Company owned.

Company owned devices in most cases are devices, mostly Windows and macOS laptops which are enrolled and managed using Microsoft Endpoint Manager/Intune. The device has several policies applied, and we can measure if the device is compliant to our security policies by using Compliance policies. Mobile devices can also be enrolled and managed, but with the current capabilities of what MAM can offer, and the way these devices are being used, I see that less often. Even though there are endless possibilities to manage mobile devices using MDM, these implementations are complex and must also be maintained.

5.2.2.2 Personally owned.

Personally owned devices, are owned by the user. In my opinion it does not make sense to start managing those devices, and most of the time these devices aren't even suitable to be company managed. If you ask the user bringing the device if he or she wants IT to manage the device, they will probably say NO. The Bring Your Own Device (BYOD) principle was a nice idea, but when it comes to

managing a device and measure its integrity by using compliance policies, in the end you have to make the choice on whether you want to manage the device as a company yes or no. Managing a device comes at a cost and you should really ask yourself if the benefits outweigh the costs.

5.2.2.3 Company owned, personally used

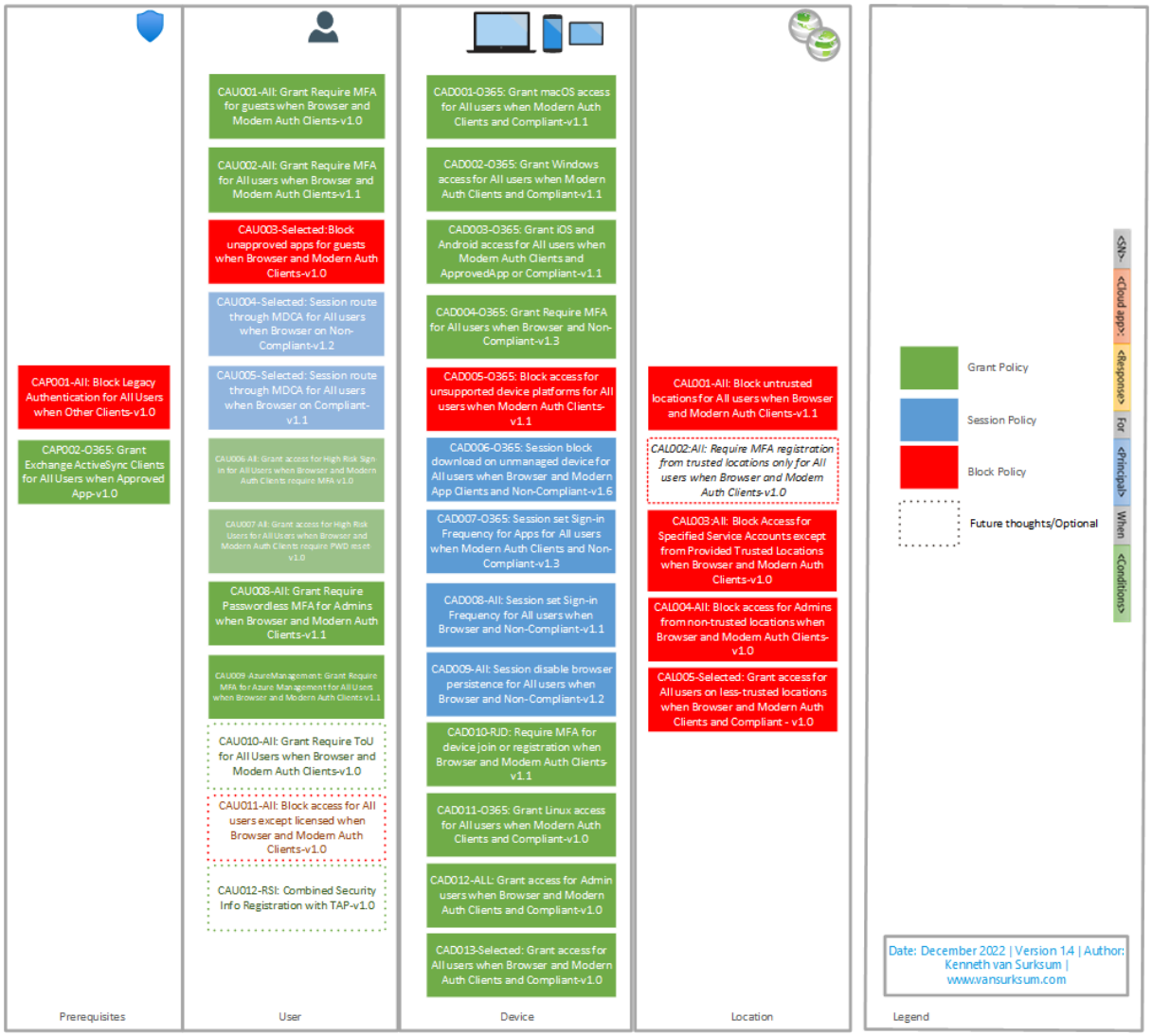
When it comes to mobile devices, this is the scenario encountered the most. Even though the company bought the device, the user using the device considers it personal. So besides hosting applications containing company data, the device will contain personal email, personal pictures, personal apps, and more personal data as well. If that is the status of the device, you will have a real hard story to tell your end users if you want to bring that device back under MDM and fully manage it from that point forward.

5.2.2.4 Other Company owned

If you work with external consultants but do supply those consultants with an Azure AD account (because you want those consultants to act on behalf of your company) it might be that those consultants already have a device managed by their own company. One of the limitations of MAM is that you cannot have more than one MDM solution managing the App. In that case the only option left over is to allow those external consultants to read their email using the web browser (which works quite well if you ask me).

The functional flowchart is my recommendation, it might be that you have another opinion or other requirements which require you to revise the flowchart. I think that the flowchart is a good starting point, in my work as a consultant I see many implementations where the IT department started with the Conditional Access policies, not having a clear idea on what they want to accomplish functionally.

Based on the Functional specifications I created a set of Default Conditional Access policies, these policies are described in more detail in chapter 5.


CAU001-All: Grant Require MFA for guests when Browser and Modern Auth Clients-v1.0
CAU002-All: Grant Require MFA for All users when Browser and Modern Auth Clients-v1.1
CAU003-Selected:Block unapproved apps for guests when Browser and Modern Auth Clients-v1.0
CAU004-Selected: Session route through MDCA for All users when Browser on Non-Compliant-v1.2
CAU005-Selected: Session route through MDCA for All users when Browser on Compliant-v1.1
CAU006-All: Grant access for High Risk Sign in for All Users when Browser and Modern Auth Clients require MFA v1.0
CAU007-All: Grant access for High Risk Users for All Users when Browser and Modern Auth Clients require PWD reset v1.0
CAU008-All: Grant Require Passwordless MFA for Admins when Browser and Modern Auth Clients-v1.1
CAU009-AzureManagement: Grant Require MFA for Azure Management for All Users when Browser and Modern Auth Clients v1.1
CAU010-All: Grant Require ToU for All Users when Browser and Modern Auth Clients-v1.0
CAU011-All: Block access for All users except licensed when Browser and Modern Auth Clients-v1.0
CAU012-RSI: Combined Security Info Registration with TAP-v1.0
CAD001-O365: Grant macOS access for All users when Modern Auth Clients and Compliant-v1.1
CAD002-O365: Grant Windows access for All users when Modern Auth Clients and Compliant-v1.1
CAD003-O365: Grant iOS and Android access for All users when Modern Auth Clients and ApprovedApp or Compliant-v1.1
CAD004-O365: Grant Require MFA for All users when Browser and Non-Compliant-v1.3
CAD005-O365: Block access for unsupported device platforms for All users when Modern Auth Clients-v1.1
CAD006-O365: Session block download on unmanaged device for All users when Browser and Modern App Clients and Non-Compliant-v1.6
CAD007-O365: Session set Sign-in Frequency for Apps for All users when Modern Auth Clients and Non-Compliant-v1.3
CAD008-All: Session set Sign-in Frequency for All users when Browser and Non-Compliant-v1.1
CAD009-All: Session disable browser persistence for All users when Browser and Non-Compliant-v1.2
CAD010-RJD: Require MFA for device join or registration when Browser and Modern Auth Clients-v1.1
CAD011-O365: Grant Linux access for All users when Modern Auth Clients and Compliant-v1.0
CAD012-ALL: Grant access for Admin users when Browser and Modern Auth Clients and Compliant-v1.0
CAD013-Selected: Grant access for All users when Browser and Modern Auth Clients and Compliant-v1.0
CAL001-All: Block untrusted locations for All users when Browser and Modern Auth Clients-v1.1
CAL002:All: Require MFA registration from trusted locations only for All users when Browser and Modern Auth Clients-v1.0
CAL003:All: Block Access for Specified Service Accounts except from Provided Trusted Locations when Browser and Modern Auth Clients-v1.0
CAL004-All: Block access for Admins from non-trusted locations when Browser and Modern Auth Clients-v1.0
CAL005-Selected: Grant access for All users on less-trusted locations when Browser and Modern Auth Clients and Compliant - v1.0
CAP001-All: Block Legacy Authentication for All Users when Other Clients-v1.0
CAP002-O365: Grant Exchange ActiveSync Clients for All Users when Approved App-v1.0
Prerequisites
User
Device
Location
Grant Policy
Session Policy
Block Policy
Future thoughts/Optional
<Name> <Cloud apps> <Response> For <Principals> When <Conditions>
Date: December 2022 | Version 1.4 | Author: Kenneth van Surksum | www.vansurksum.com
Legend

5.3 Browser restrictions and configuration when using Conditional Access.

Even though you are working in the browser on a compliant device, does not necessarily mean that Azure AD can detect that. Therefore, you must make sure that your browsers are configured correctly before you implement the Conditional Access policy.

Some examples I often encounter: End user is working on a compliant device, but cannot download or print files when using the web interface to connect to SharePoint online, this is caused by the App Enforced Restrictions policy being active (see: Limit Access to Outlook Web Access, SharePoint Online and OneDrive using Conditional Access App Enforced Restrictions). Or, MDA blocks the download of a file, even though the user is working on a compliant device. (see: Extending Conditional Access to Microsoft Defender for Cloud Appsusing Conditional Access App Control)

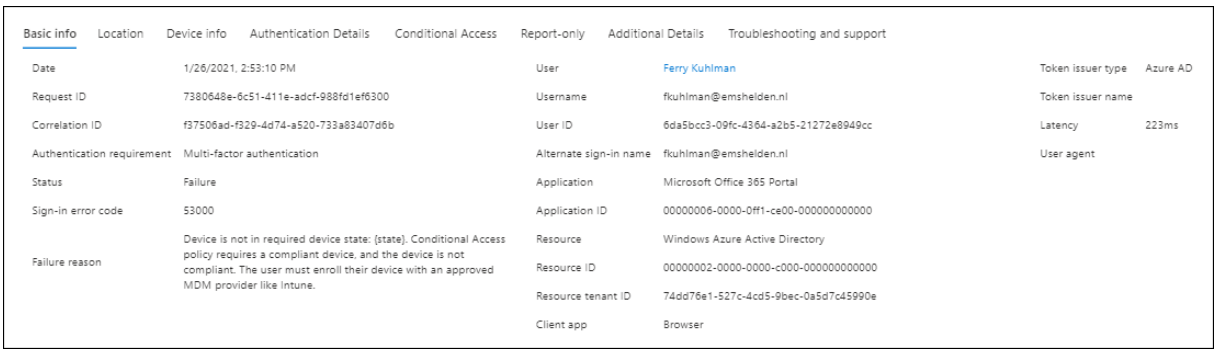
5.3.1 Browser support

This all has to do with browser support and configuration, below is an overview of the requirements and what is, and what is not supported. Currently Microsoft supports the following browsers:

OS	Browsers
Windows 10	Microsoft Edge, Chrome, Firefox 91+
iOS	Microsoft Edge, Safari
Android	Microsoft Edge, Chrome
Windows Server 2019	Microsoft Edge, Chrome
Windows Server 2022	Microsoft Edge, Chrome
macOS	Microsoft Edge, Chrome, Safari

5.3.1.1 Sign-in Logging

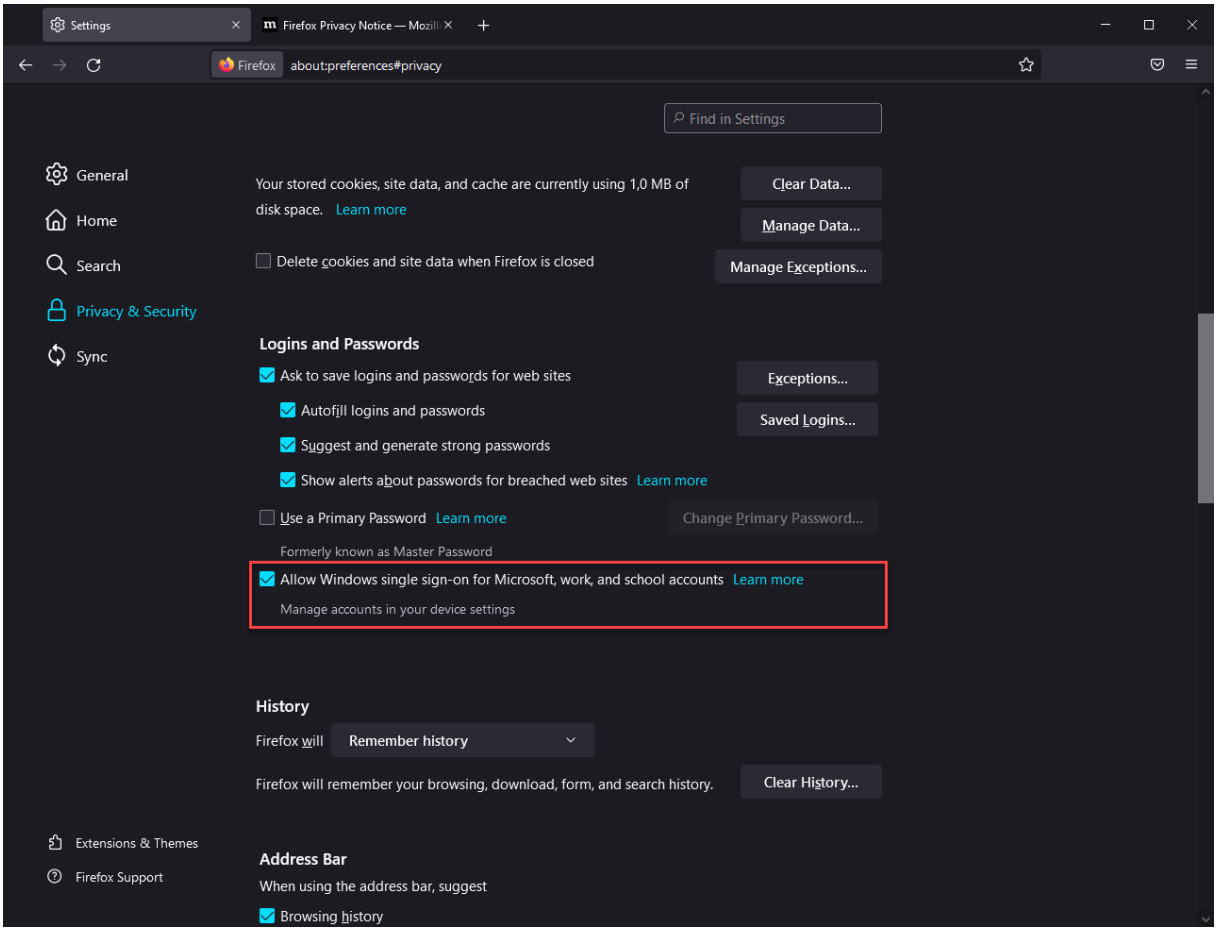
When users are using a non-supported configuration, this might reflect as followed in the Azure AD sign-in logging. As you can see the Conditional Access policy requires a compliant device before access to the resource can be given. And in this case, our test user Ferry was working on a compliant device (you have to take my word for it).



5.3.2 Mozilla FireFox

Starting with Firefox version 91, Mozilla is now supporting Single sign-on support (SSO) and device-based Conditional Access as announced by Microsoft in the What's new in Azure Active Directory for August 2021. The feature is still in Public Preview from a Microsoft point of view, and considered Advanced and experimental from a Mozilla point of view.

For now the option must be enabled, this can be done by opening Firefox, going to the menu and by selecting settings, from there you can go to Privacy & Security and go to the Logins and Passwords section. From there you can enable the feature by selecting the checkbox in front of "Allow Windows single sign-on for Microsoft, work and school accounts".



5.3.2.1 Configuring the settings from Microsoft Endpoint Manager

Since Mozilla is also providing ADMX files for configuring it's browser, we can use this functionality to configure the "Allow Windows single sign-on for Microsoft, work and school accounts" setting using a custom Configuration Profile.

I will not go into too much detail on how you can use a custom ADMX file and leverage its settings in a Microsoft Endpoint Manager configuration profile. If you want to know more about that, I suggest that you read the following article from fellow-MVP Peter Klapwijk: Manage new ADMX Backed Windows 10 policies with Microsoft Intune and the Microsoft documentation "Win32 and Desktop Bridge app ADMX policy Ingestion"

You can download the latest ADMX file for FireFox from the Mozilla Github page. Mozilla even has a Knowledge base article explaining how to ingest and set settings which you can find here: "Managing Firefox with Microsoft Endpoint Manager (Intune)". The OMA0URI configuration string and possible values can be found under the WindowsSSO section of the Readme.md file.

Based on the information provided above we can create our custom configuration profile in Microsoft Endpoint Manager.

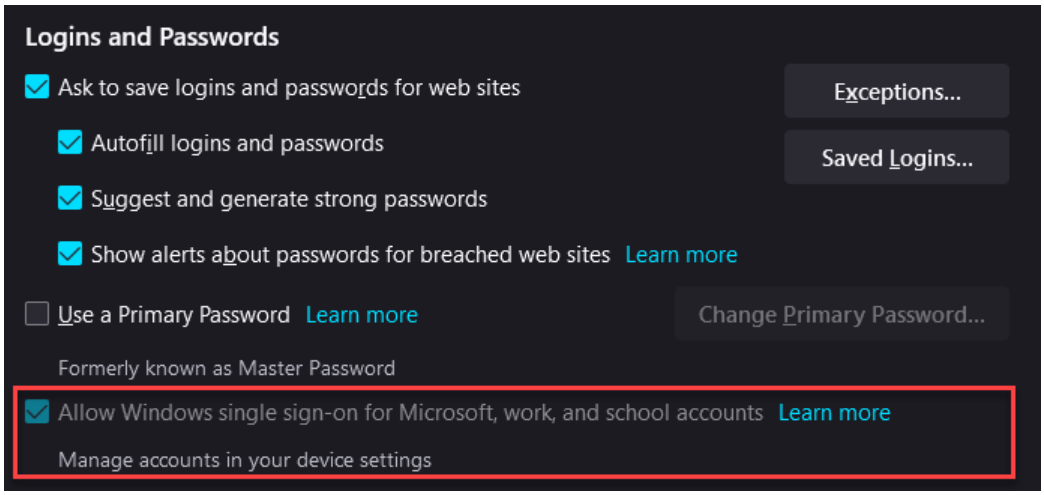
5.3.2.2 *Create the custom Device Configuration Profile*

Name: W10 - CP - Mozilla Firefox Configuration - v1.0 (or any other name you want to provide)
Profile type: Custom
OMA-URI settings:

Name: FireFox ADMX Ingestion
Description: FireFox ADMX Ingestion
OMA-URI: ./Device/Vendor/MSFT/Policy/ConfigOperations/ADMXInstall/Firefox/Policy/FirefoxAdmx
Value (String): Copy the content of the firefox.admx file into the Value field

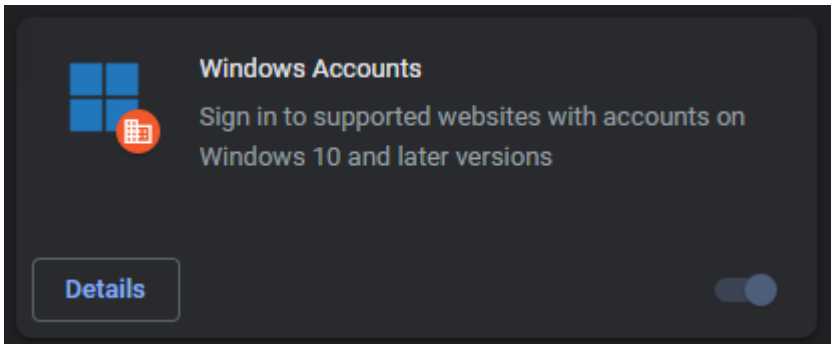
Name: Enable Windows SSO
Description: Enable Windows SSO
OMA-URI: ./Device/Vendor/MSFT/Policy/Config/Firefox~Policy~firefox/WindowsSSO
Value (String): <enabled/> <data id="WindowsSSO" value="1"/>

After the Configuration profile is successfully applied, you will notice that in FireFox the option is grayed-out meaning that it cannot be changed by the user.



5.3.3 Google Chrome

In order for the Google Chrome browser to support the device authentication you must deploy the Windows accounts extension in the Chrome browser to your devices. You'll need this extension if you want to use the device compliancy within your Conditional Access policies.



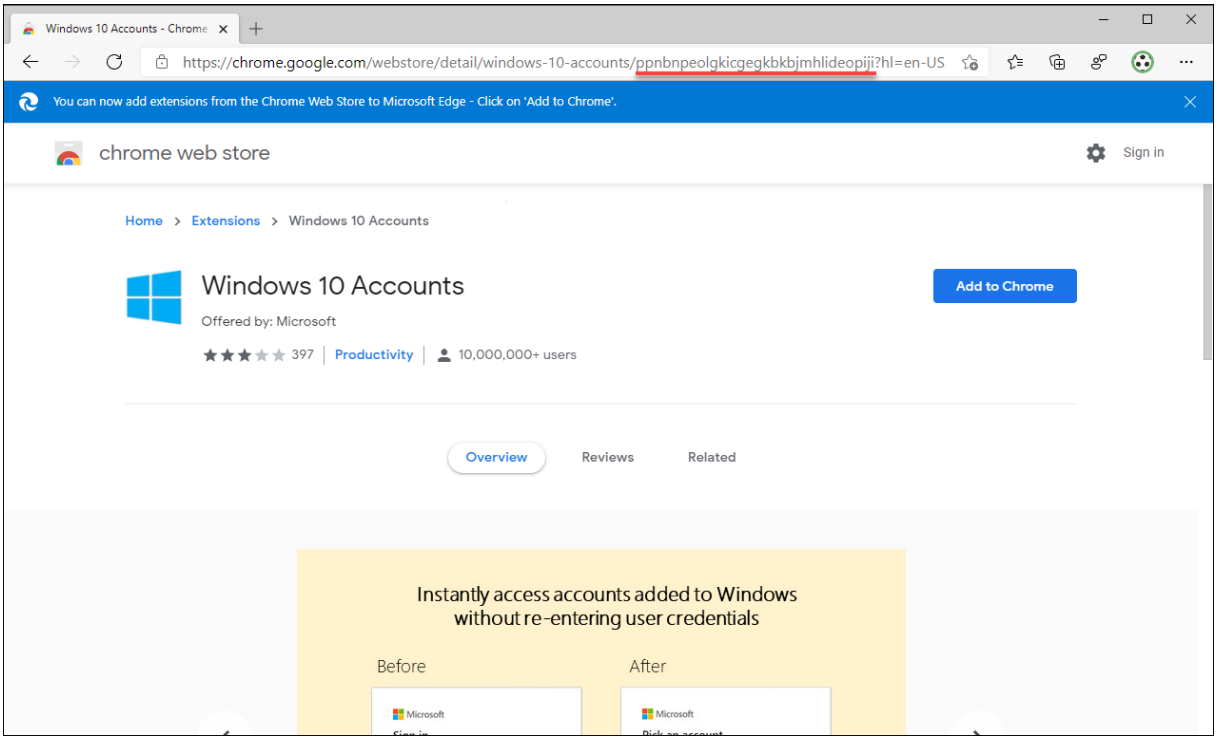
5.3.3.1 Deploying extensions for Google Chrome using Microsoft Endpoint Manager

You can configure the Google Chrome browser running on a Intune/MEM managed Windows 10 device by using a Configuration Profile with a custom profile type.

For this to work, we first need to download the Google Chrome Bundle. Within that bundle you can find a folder called ADMX. From that folder you will need the chrome.admx file. Within the ADMX file we will use the ExtensionInstallForceList parameter to define the extensions we want to have installed.

```
<policy class="Both" displayName="$(string.ExtensionInstallForcelist)" explainText="$(string.ExtensionInstallForcelist_Explain)"
key="Software\Policies\Google\Chrome" name="ExtensionInstallForcelist" presentation="$(presentation.ExtensionInstallForcelist)">
  <parentCategory ref="Extensions"/>
  <supportedOn ref="SUPPORTED_WIN7"/>
  <elements>
    <list id="ExtensionInstallForcelistDesc" key="Software\Policies\Google\Chrome\ExtensionInstallForcelist" valuePrefix=""/>
  </elements>
</policy>
```

Secondly we must determine the unique identifiers for the extension(s) we want to install. You can determine this by browsing to the chrome web store. From the chrome web store we need the Windows 10 Accounts extension, which at time of writing has the following id:"ppnbnpeolgkicgegkbkbjmhlideopiji" make sure that if you are configuring this yourself to doublecheck whether the id still matches.

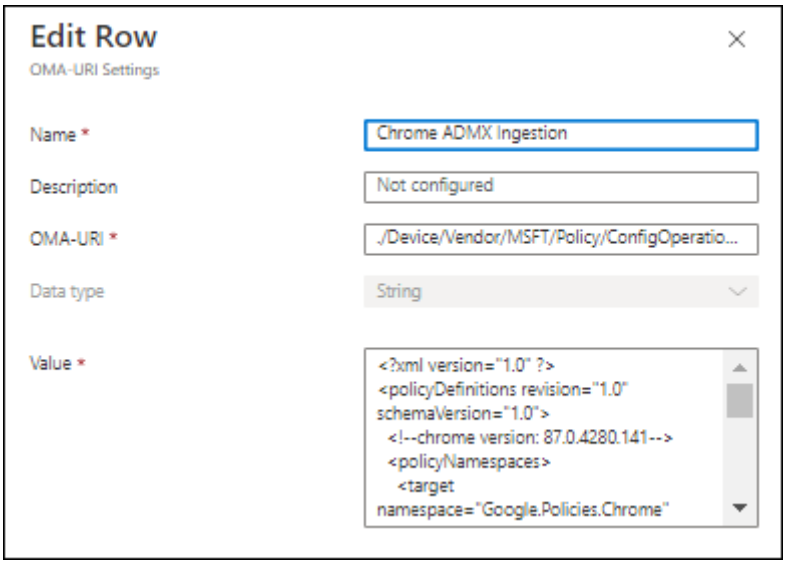


After downloading the ADMX file and figuring out which extensions we want to install by their ID in the chrome web store we can define our Configuration Policy.

First make sure that the custom policy ingests the ADMX file, use the following OMA-URI settings to configure this:

Name: Chrome ADMX Ingestion (can be any name, but make it easy to understand what it does)
Description: <Your description if you prefer>

OMA-URI:
./Device/Vendor/MSFT/Policy/ConfigOperations/ADMXInstall/Chrome/Policy/ChromeAdmx
Data type: String
Value: Copy/Paste here the contents of the ADMX file.



Secondly, we can define the setting we want to make as defined in the ingested ADMX file. Be very careful here.

Name: ExtensionInstallForcelist (can be any name, but make it easy to understand what it does)
Description: <Your description if you prefer>
OMA-URI:
./Device/Vendor/MSFT/Policy/Config/Chrome~Policy~googlechrome~Extensions/ExtensionInstallForcelist
Data type: String
Value: <enabled/> <data id="ExtensionInstallForcelistDesc" value="1ppnbnpeolgkicgegkbkbjmhlideopiji;https://clients2.google.com/service/update2/crx2bkbeeeffjjeopflfhgeknacdieedcoml;https://clients2.google.com/service/update2/crx"/>

In the example above I also install another extension (The Microsoft Defender Browser Protection) as you can see the value must be carefully composed.

First, the ExtensionInstallForceList will eventually end up as a REG_MULTISZ registry string. Which means that each entry must be separated by the Unicode character 0xF000 (or  when encoded). You can also see that the URL https://clients2.google.com/service/update2/crx is being used. That URL is needed for the browser to determine the download path once its ready to download the extension. Each extension is also numbered, so the Windows 10 Accounts extension is number 1 and the Microsoft Defender Browser Protection extension is number 2. If you want to add more use 3, 4, etc…

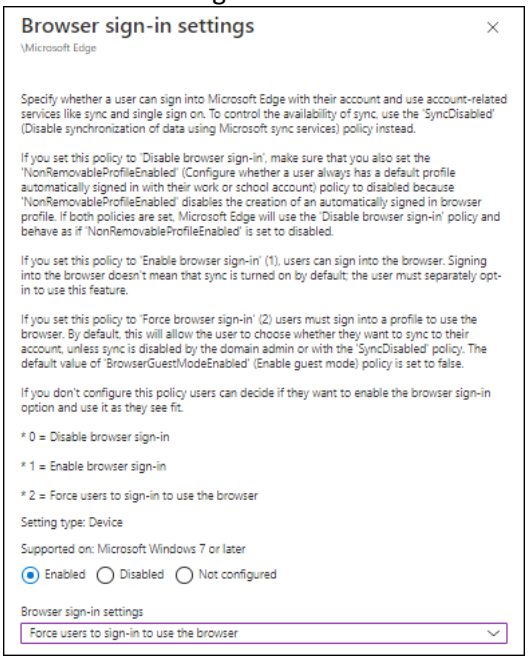
Once configured assign the policy to a group and you verify whether the necessary extensions are installed within the Google Chrome browser.

5.3.4 Microsoft Edge

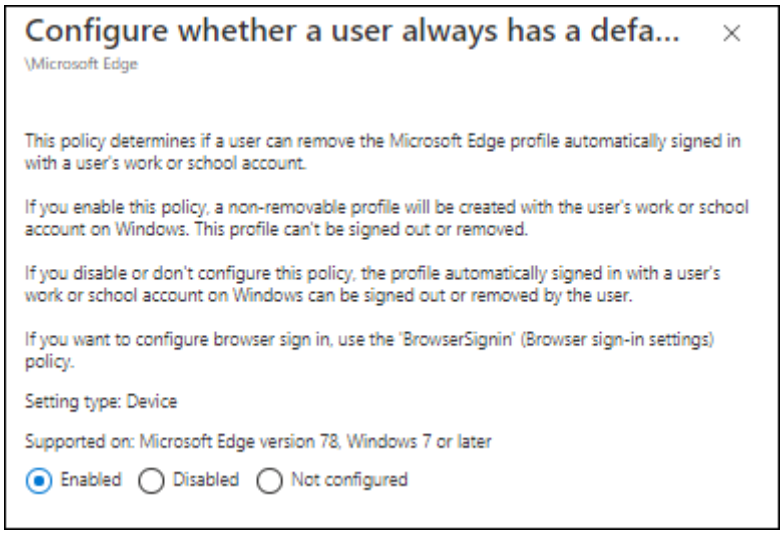
Microsoft Edge obviously supports device authentication, but whether this is being used is depending on the profile you are signed into. When you are signed into a Microsoft Edge profile with enterprise Azure AD credentials, Microsoft Edge allows seamless access to enterprise cloud resources protected using Conditional Access. On a compliant device, the identity accessing the resource should match the identity on the profile. See: [Accessing Conditional Access protected resources in Microsoft Edge] for more information.

If you want to configure this sign-in for your devices you can use two settings using a Configuration Profile with an Administrative Template.

The first setting we must modify is the "Browser sign-in settings", make sure that the setting is enabled, and that the option "Force users to sign-in to use the browser" is selected.

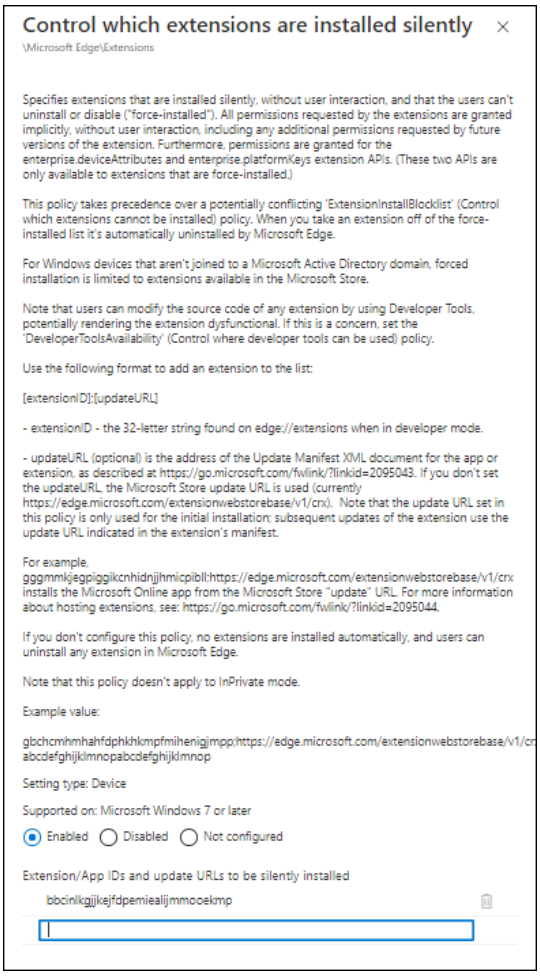


The second setting we must modify is called: "Configure whether a user always has a default profile automatically signed in with their work or school account". Make sure that this setting is enabled.



Finish the Configuration Profile and assign it to a group of your choosing.

On a sidenote, installing extensions in Microsoft Edge is much easier, since it is also part of the Administrative Templates, search for "Control which extensions are installed silently" and just supply the unique id or more in a list.



5.4 Azure AD Conditional Access authentication context

On May 26th 2021, Conditional Access authentication context was made available in public preview as announced by Alex Simons and Caleb Baker in the following article: Conditional Access authentication context now in public preview.

5.4.1 Current data protection options

With Conditional Access, we are able to protect the identity of our Azure Active Directory users and control the access to company data hosted in Office 365 and SaaS applications. In order to protect the company data we have several options available.
We can allow full access on managed devices, these devices are either Azure AD Hybrid joined devices, or devices which are compliant and therefore managed by Microsoft Endpoint Manager.
We can restrict access on non-managed devices, by either blocking or limiting the access given.
We can protect the data itself, by using sensitivity labels which are either applied by the user, or automatically.

5.4.2 Conditional Access App Enforced Restrictions

We can use a Conditional Access policy, to either restrict or block access to data hosted in SharePoint Online on tenant or granular level. When the access control for unmanaged devices in SharePoint online is configured to Allow limited, web-only access you can restrict access to the content hosted in SharePoint online to be only available in a web browser where downloading and printing of an opened file is prohibited. This functionality can be accomplished by leveraging the Conditional Access App Enforced Restrictions option as part of the session access controls in a Conditional Access policy. I've described this functionality in the following article: Limit Access to Outlook Web Access, SharePoint Online and OneDrive using Conditional Access App Enforced Restrictions.

If you require more granularity for your Conditional Access App Enforced Restrictions you can either specify the Control access from unmanaged devices setting on a per SharePoint site level using PowerShell or use Sensitivity labels, as described in the following article: Defining more granularity for your Conditional Access App Enforced Restrictions using Sensitivity Labels

5.4.3 Microsoft Defender for Cloud Apps (MDA)

When our company data is not hosted on SharePoint online and Single Sign On (SSO) to a SaaS based 3rd party application is configured, we can route the users session to the third party application through Microsoft Defender for Cloud Apps(MDA) which has a reverse proxy functionality (session control). Routing a user's session through MDA is accomplished by leveraging the Conditional Access App Control functionality available under the session access controls.

Once a session is reversed proxied through MDA we have several options available to control what can be done with files downloaded from the 3rd Party SaaS application. You can also decide not to use App Enforced Restrictions and use App Control instead for your SharePoint Online sites as well. I've explained some of the use cases in the following article before: Extending Conditional Access to Microsoft Defender for Cloud Apps using Conditional Access App Control

5.4.4 Conditional Access Authentication Context

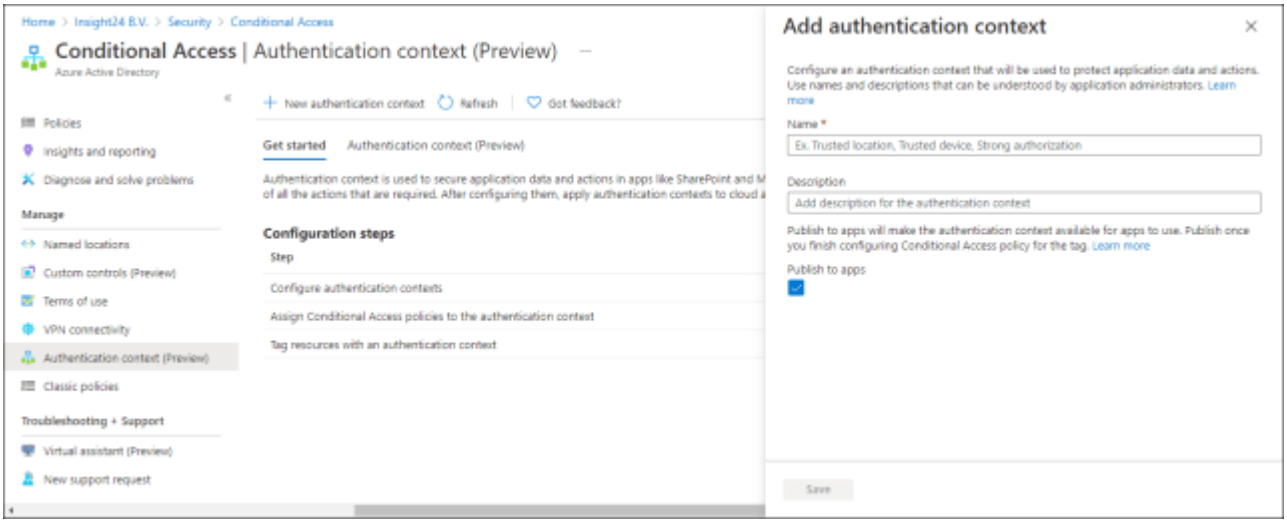
Conditional Access Authentication Context is able to trigger a Conditional Access policy when sensitive content is accessed. With this new functionality new scenarios become available, some examples are:

- You can require MFA when a SharePoint site with a certain sensitivity label gets accessed on an unmanaged device.
- You can block the session from a Conditional Access policy when a SharePoint site with a certain sensitivity label gets accessed on a unmanaged device.
- You can require that Terms of Use must be accepted first before access to a SharePoint site with a certain sensitivity label gets accessed.
- You can set the sign-in frequency to a low value when a SharePoint site with a certain sensitivity label gets accessed.
- You can trigger the Authentication Context from within a session control policy defined in MDA, for example to trigger MFA as defined in the Conditional Access policy. You could for example trigger MFA when a document with a certain sensitivity label gets downloaded or require the user to accept a Terms of Use first.
- App developers can leverage the authentication context in their own apps

So, how does it work?

5.4.5 Create Authentication context labels

The first thing that needs to be done in order to start working with Authentication context is to create a new label/new labels for authentication context. These labels can be created under the Authentication context (Preview) menu in the Conditional Access section of the Azure AD Admin portal. You can create a new label by clicking on + New authentication context which will open up a new windows where the name and description for the new label can be supplied. By default the Publish to apps option is selected, making the label available for use within apps.



To explain the concept I'm going to create the following authentication context labels

Label name	Explanation
Internal – High Authentication Context	Requires compliant device, TOU, or MFA
Internal – Medium Authentication Context	Requires compliant device
Internal – Low Authentication Context	MFA

With these labels we can build the following scenario's:

- Only grant access to internal users when they work on a compliant device, and require them to provide MFA
- Only grant access to internal users when they work on a compliant device
- Only grant access to internal users after they have provided MFA
- Only grant access to guest users after they accepted the Terms of Use (TOU), and require them to provide MFA
- Only grant access to guest users after they have provided MFA

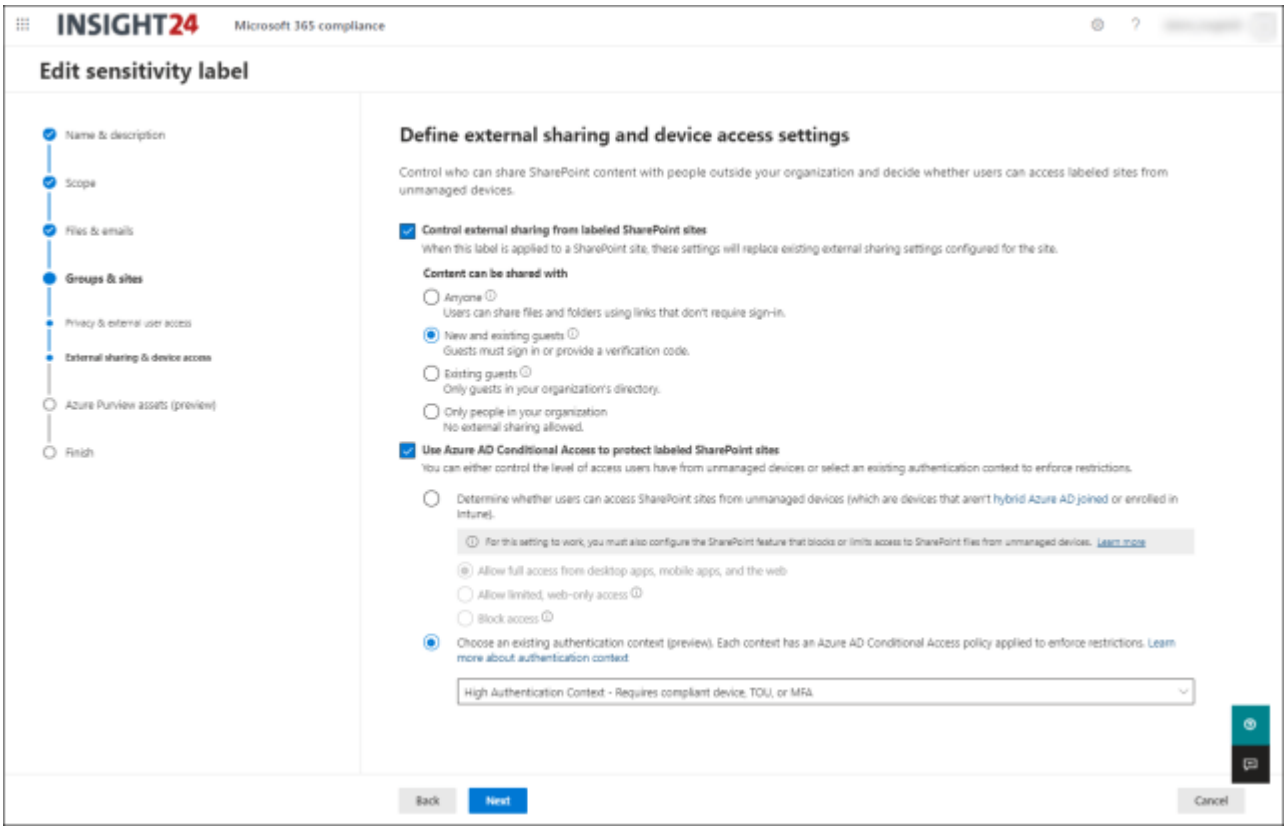
Creating the total solution requires the following steps:

- Create Authentication context label
- Apply authentication context applicability in sensitivity label, MDA or App configuration
- Create Conditional Access policy which gets triggered if the authentication context applies.

5.4.6 Use the Authentication context label in your sensitivity labels

You can use authentication context in a sensitivity label. The authentication context in this case will replace the options which are available in SharePoint on how to handle unmanaged devices. So, you either need to choose between unmanaged device options in SharePoint (full access, limited web only access, or block access) or use authentication context.

When you create a new or modify an existing sensitivity label, you will get the screen similar to the one below.

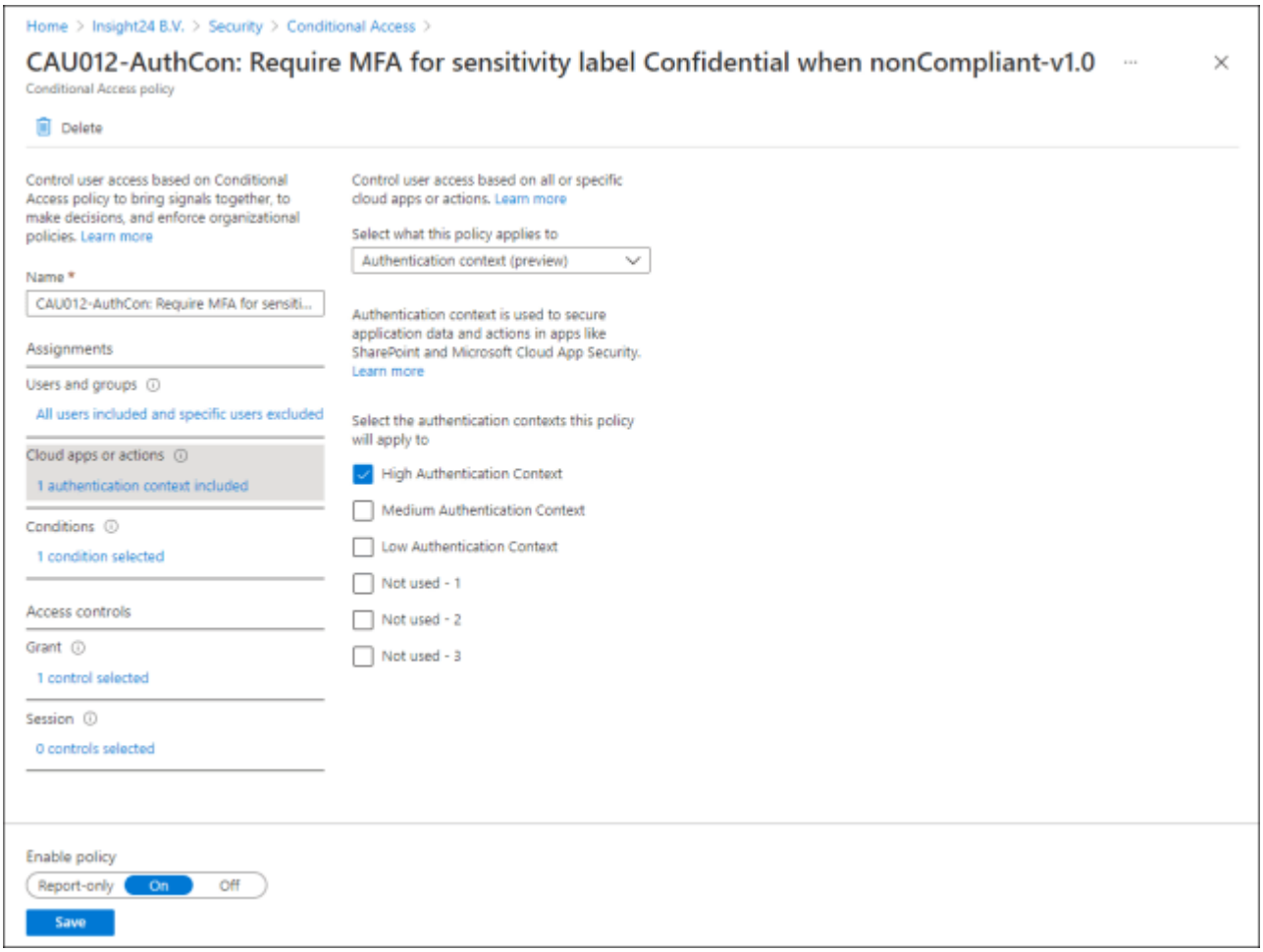


The option to apply authentication context is only available within the Groups & sites section of the sensitivity label, which means that you can only define it on the container level and not on file level.

5.4.7 Creating the Conditional Access policy for the Authentication context label

So, now that the authentication context is added to the sensitivity label (sensitivity labels) we can define the Conditional Access policy which gets triggered at the moment that the SharePoint site which has the sensitivity label applied is accessed.

The Authentication context is a configurable action as part of the “Cloud apps or actions” section of the Assignments within a Conditional Access policy. Here you can a list of the Authentication Context label which have been defined, notice that you can also use multiple authentication context labels at once.



Below are some examples on possible Conditional Access policies which you can configure
In the example below we require both a compliant device **and** MFA for All users except Guest and External users (since we can’t require them to have a compliant device)

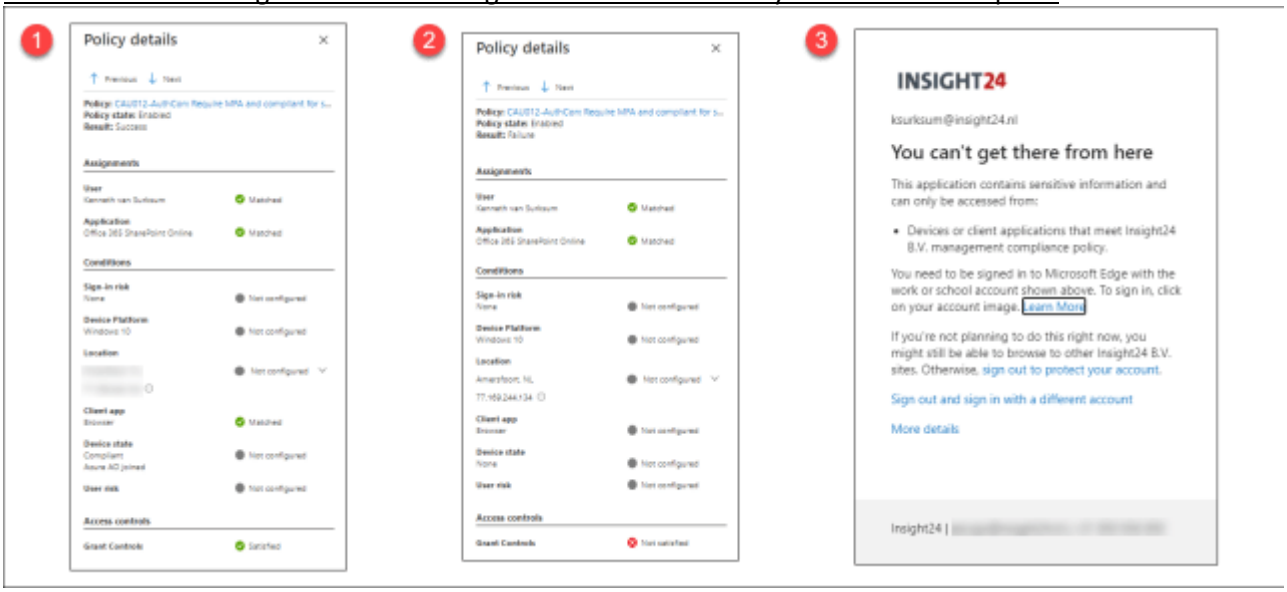
CAU012-AuthCon: Require MFA and compliant for sensitivity label confidential-v1.0

Assignments				Access Controls	
Users	Cloud Apps or Actions	Conditions		Grant	Session
All Users Except AAD_AA_ConAcc-Breakglass AAD_AA_CAU012-Exclude All Guest and External Users	Authentication Context: High Authentication Context	Client Apps: Browser Mobile Apps and Desktop Clients		Grant Require multi-factor authentication And Require device to be marked as compliant	

With this policy active, from that point forward we can see that we can only access the SharePoint site from a Compliant device, with multi-factor authentication (1). We are not required to provide

MFA, since the device I'm using for this test is using Windows Hello for Business (WHfB) which meets the MFA requirement.

If we would access the environment from a non-compliant device, we would receive the message as mentioned in (3), which reflects to the Failure described in (2). Keep in mind that also your browser must meet some requirements in order to be able to leverage this functionality, see: Browser restrictions and configuration when using Conditional Access on your modern workplace.



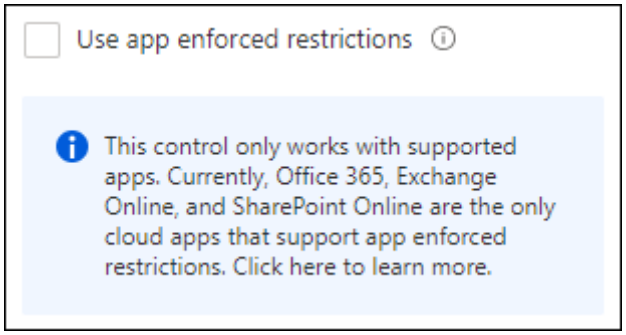
For the conditional access policies we can create some variants, for example the policy below for Guest and External users which requires both MFA and acceptance of the Terms of Use.

CAU013-AuthCon: Require MFA and Accept TOU for sensitivity label confidential-v1.0

Assignments				Access Controls	
Users	Cloud Apps or Actions	Conditions		Grant	Session
All Guest and External Users	Authentication Context: High Authentication Context	Client Apps: Browser Mobile Apps and Desktop Clients		Grant Require multi-factor authentication And 24 - Terms of Use	
Except					
AAD_AA_ConAcc-Breakglass					
AAD_AA_CAU013-Exclude					

5.4.8 Impact on using App enforced restrictions

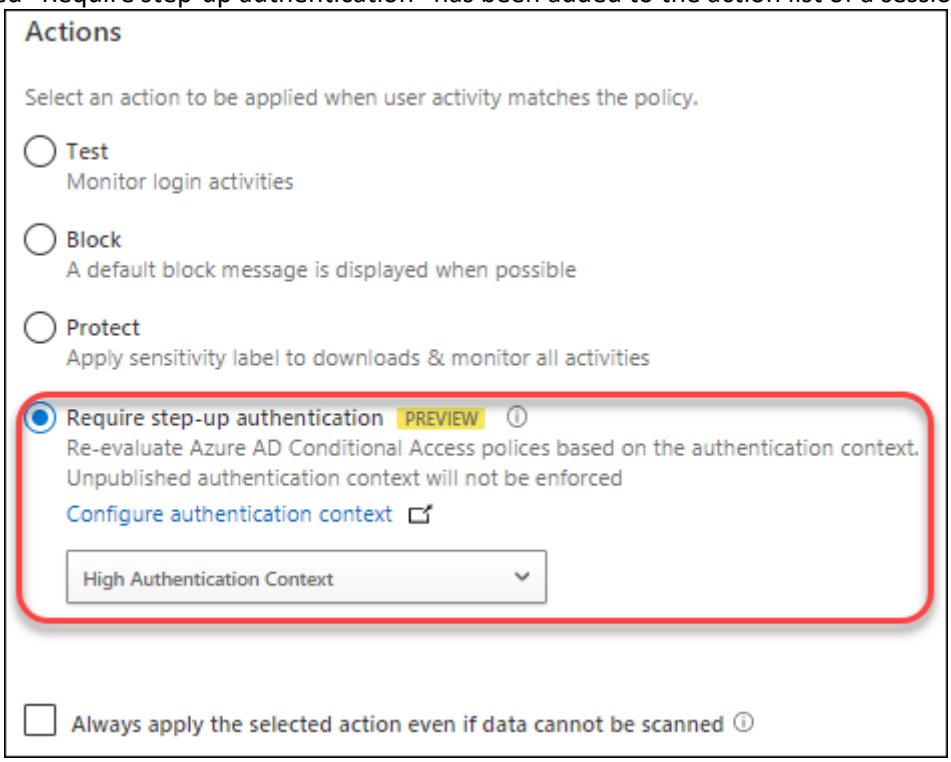
Since we can either choose between unmanaged device options in SharePoint (full access, limited web only access, or block access) or use authentication context. We cannot offer the option anymore to provide limited web only access and require MFA for example. Since we are using Authentication context in our Conditional Access policy, the App enforced restrictions option under session control in the Conditional Access policy is not available, since you can only use this option if you have Office 365, Exchange Online and SharePoint Online is selected as a cloud app.



So unless we reverse-proxy the session through Microsoft Cloud App Security, we lose the option to restrict users from downloading and printing files as part of App enforced restrictions. See: Limit Access to Outlook Web Access, SharePoint Online and OneDrive using Conditional Access App Enforced Restrictions for more information about the functionality that

5.4.9 Use the Authentication context label in your MDA Session Policy

We can also use authentication context in a Microsoft Defender for Cloud Appssession policy. A new action called “Require step-up authentication” has been added to the action list of a session policy.



So, in the case of the scenario which I described before in the following article: Extending Conditional Access to Microsoft Defender for Cloud Appsusing Conditional Access App Control, we described the scenario where we reverse-proxied the SharePoint online connection through MDA. Within MDA we defined a session policy with a session control type of “Control file download (with inspection) where we used the action “Block” if the Device was not Intune Compliant or Hybrid Azure AD joined, and the file name contains the word “BSN”

We can now extend this scenario to not block this file download, but trigger the Conditional Access policy which requires the approval of TOU and MFA.

5.5 Understanding and governing reauthentication settings in Azure Active Directory

Governing when users receive authentication prompts when authenticating to Azure Active Directory (Azure AD) is depending on more than one setting, on which functionalities are in use and in which scenario you authenticate (Browser, Modern clients or other). Reauthentication can take place by asking for a single factor, like password, FIDO, the password less option in the Microsoft Authenticator app or by using Multi Factor Authentication (MFA)

So you might understand that how reauthentication must be configured really depends per company and per scenario, so luckily Microsoft provides options which you can configure.

Some examples:

- You want users to reauthenticate more often when they come from a non-managed or non-registered device.
- You want users to reauthenticate more often when using a certain cloud application which you make available via Azure AD single sign on
- You might want some users in your organization to authenticate more often than others based on their risk profile.

5.5.1 The settings which make up the experience

Azure AD has a default sign-in frequency of 90 days, this might seem like a long time but there are some scenario's which require the user to sign in again, like:

- A change in the compliancy status of the managed device
- Disabling the account
- Revoking the sessions for the user
- Changing the password

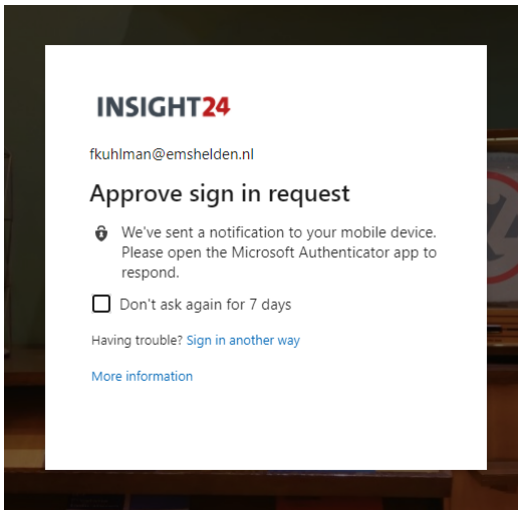
Microsoft explains this default configuration as followed: "*don't ask users to provide their credentials if security posture of their sessions has not changed*", and states: "*If users are trained to enter their credentials without thinking, they can unintentionally supply them to a malicious credential prompt.*"

Personally, I think this default sign-in frequency makes sense when users work on managed devices, for other scenario's some adjustments might be necessary.

5.5.1.1 Azure Multi Factor Authentication Settings

When configuring Multi Factor Authentication you have the option to remember the multi factor authentication on trusted devices. When configured, the option allows you to bypass verifications for a specified number of days.

When using the Browser this is achieved by setting a cookie which expires after the specified time. In the screenshot below this has been set to 7 days.

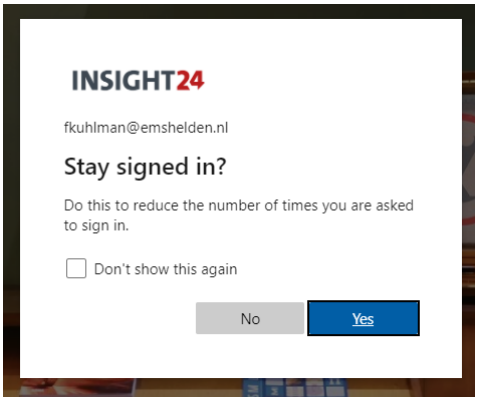


Non-browser apps use refresh tokens with a default validity of 1 hour, while validating the refresh token the check for MFA is performed as well.

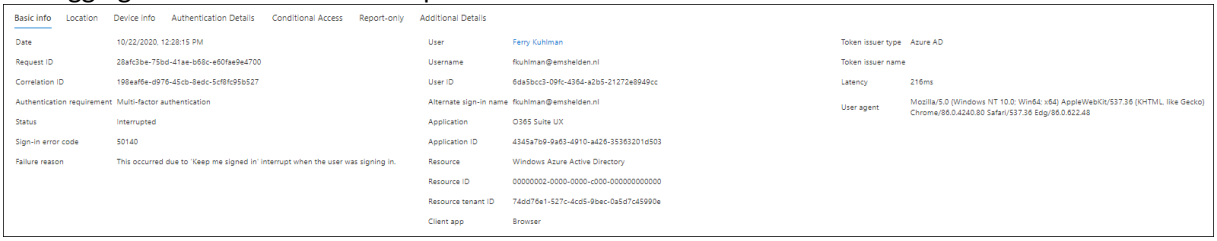
Microsoft recommends that you set this setting to 90 days, in line with the default sign-in frequency. If needed, you must revoke the MFA session to force the user to re authenticate using MFA.

5.5.1.2 Show option to remain signed in (KMSI)

Another option influencing the experience is the option "Show option to remain signed in, also known as "Keep me signed in (KMSI)" for which the configurable settings can be found in the User Settings.

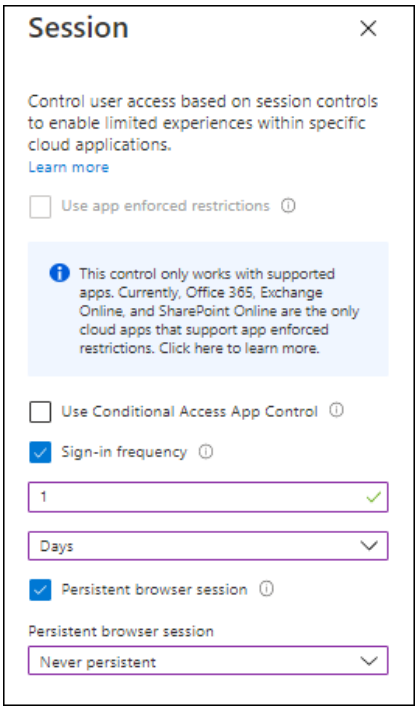


When users are presented with the Stay signed in option but abandon it, this is reflected in the Azure AD logging with a status of "Interrupted" and error code 50140.



5.5.1.3 Conditional Access

Conditional Access policies can be used to override some of the default settings in certain scenarios. By using session controls you can control how users must authenticate in different scenarios.



5.5.1.4 Sign-in Frequency

By setting the Sign-in Frequency session control you can override the default setting of 90 days to a lower setting, you can do this for example if users access your Office 365 environment from a non-managed device via the Browser, in the screenshot above we have set a sign-in frequency for 1 day.

See: Policy 1: Sign-in frequency control for an example on how to create a Conditional Access policy leveraging the sign-in frequency session control.

5.5.1.5 Persistent Browser session

A persistent browser session setting controls if users remain signed in after closing and reopening their browser window. We have 2 options here, either "Always persistent" or "Never persistent".

- This setting works correctly when "All cloud apps" are selected.
- This does not affect token lifetimes or the sign-in frequency setting.
- This will **override** the "Show option to stay signed in" policy in Company Branding.
- "Never persistent" will override any persistent SSO claims passed in from federated authentication services.
- "Never persistent" will prevent SSO on mobile devices across applications and between applications and the user's mobile browser.

See: Policy 2: Persistent browser session for an example on how to create a Conditional Access policy leveraging the "Persistent browser session" session control.

5.5.2 The scenarios which make up the experience

The scenarios under which users authenticate to your Azure AD environment are diverse, and you should understand which scenarios you will encounter and want to support within your organization. Below are some topics which should be considered when defining your scenarios.

5.5.2.1 Supported applications.

This sign-in frequency works with applications that have implemented Open Authorization (oAuth2) or OpenID Connect (OIDC) authentication protocols, which is supported for most applications working with Azure AD. Sign-in frequency also works with applications implementing the Security Assertion Markup Language (SAML) protocol for authorization and authentication as well.

When working in a Microsoft 365 modern environment you can assume that the Office desktop and mobile apps will work, also accessing the Office 365 web portals will support this without any issue. When it comes to 3rd party applications it depends. For example if an applications drops its cookies for some reason and therefore redirects back to Azure AD, then the sign-in frequency can be less.

5.5.2.2 Nonregistered devices

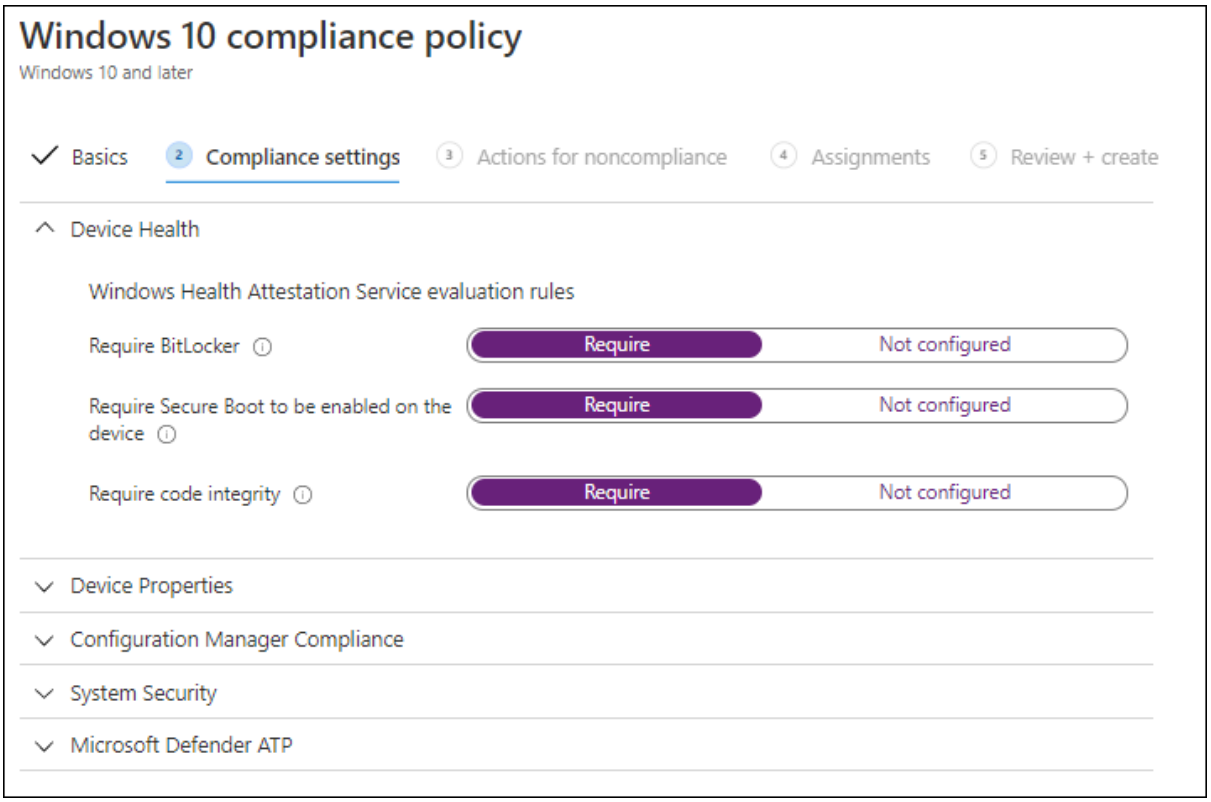
If you work on devices which are not registered in Azure AD, it might also be that applications running on top of that device are not sharing their oAuth refresh token with each other, requiring the user to authenticate multiple times.

5.5.2.3 Azure AD joined or Registered devices.

Devices which are either Azure AD Joined, or Active Directory Joined/Azure AD registered via Hybrid AD join receive a so-called Primary Refresh Tokens (PRT) allowing them to use this token for Single Sign-on (SSO) functionality. A PRT is valid for 14 days and continuously renewed (every 4 hours at least) if the user actively uses the device. For more information about how the PRT works, I suggest to read the following article: "What is a Primary Refresh Token?" on the Microsoft website.

5.5.3 Managed devices

Managed devices are devices on which you can measure compliance using Microsoft Endpoint Manager/Intune. Even though these devices are also registered, you also have the option to measure whether other security requirements are met, like for example BitLocker and Secure Boot being enabled on the device.



5.5.4 Browser used.

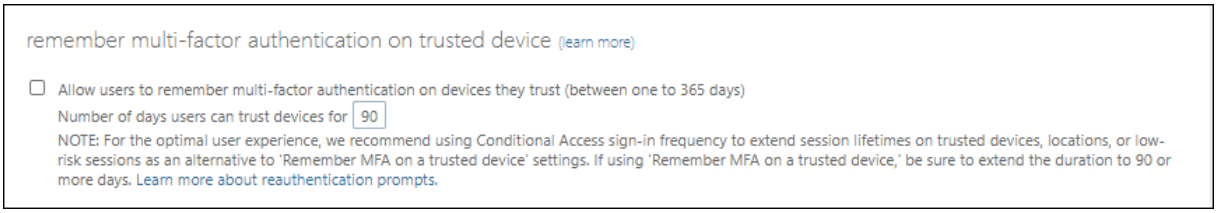
Which browser is used is an important factor when determining the scenarios. Is the browser being used an old browser like Internet Explorer, or a modern browser like Google Chrome, the new Microsoft Edge and Mozilla Firefox

5.5.5 Default settings when creating a new tenant.

When you create a new tenant today, the following default settings are available.

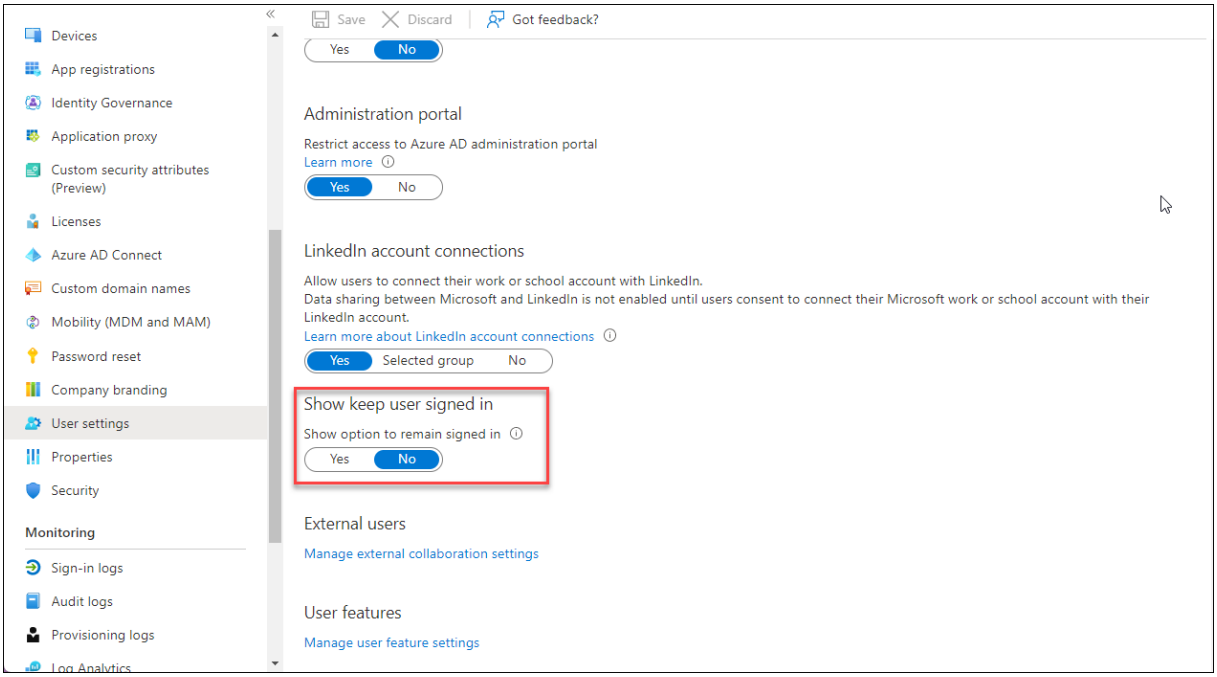
5.5.5.1 Multi factor authentication

Multi Factor Authentication by default is configured to not remember MFA on devices people trust. If you enable the setting the default number of days is set to 90 days.



5.5.5.2 Show option to remain signed in

By default, in a new tenant, the show option to remain signed in is turned on.



5.5.5.3 Conditional Access

A new tenant does not have any Conditional Access policies configured. In the tenant I provisioned even the default security settings were not applied, but that can have something to do with the fact that I used a temporary tenant which was already hydrated.

The Default Security settings provide the following settings by default.

- Requiring all users to register for Azure Multi-Factor Authentication.
- Requiring administrators to perform multi-factor authentication.
- Blocking legacy authentication protocols.
- Requiring users to perform multi-factor authentication when necessary.
- Protecting privileged activities like access to the Azure portal.

For more information, see: [Microsoft deprecates Conditional Access baseline policies in favour of Security Defaults, here is what you need to know and do]

5.5.6 Bringing it all together

So now that we know the different options on how to configure reauthentication behavior and have an idea of the different scenarios we can face we can start designing our reauthentication scenarios.

5.5.6.1 Managed devices

For MDM managed devices, having the option to measure compliance gives us options to check whether the device is secure. Because of this I would advise to keep the defaults of the supplier in this case to keep the sign-in frequency to 90 days. MFA

5.5.6.2 Managed applications

For applications which you manage using Mobile Application Management (App Protection Policies) you can set a stricter sign-in frequency policy. For example, set this 7 days using a Conditional Access policy.

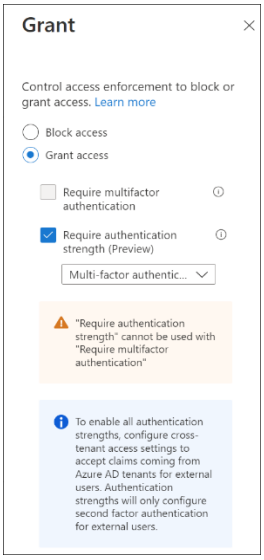
5.5.6.3 Non-managed devices

On non-managed devices (devices not compliant or not hybrid AD joined), especially when accessing the environment using the web browser (which is at this moment the real manageable option to keep your company data protection IMHO), you should even set a more restrict sign-in policy and disable browser persistence. An example would be to set the sign-in frequency to 1 day/4 hours and disable browser persistence.

Microsoft also provides some recommended settings depending on whether you have Azure AD Premium yes or no, which you can find here: Recommended settings

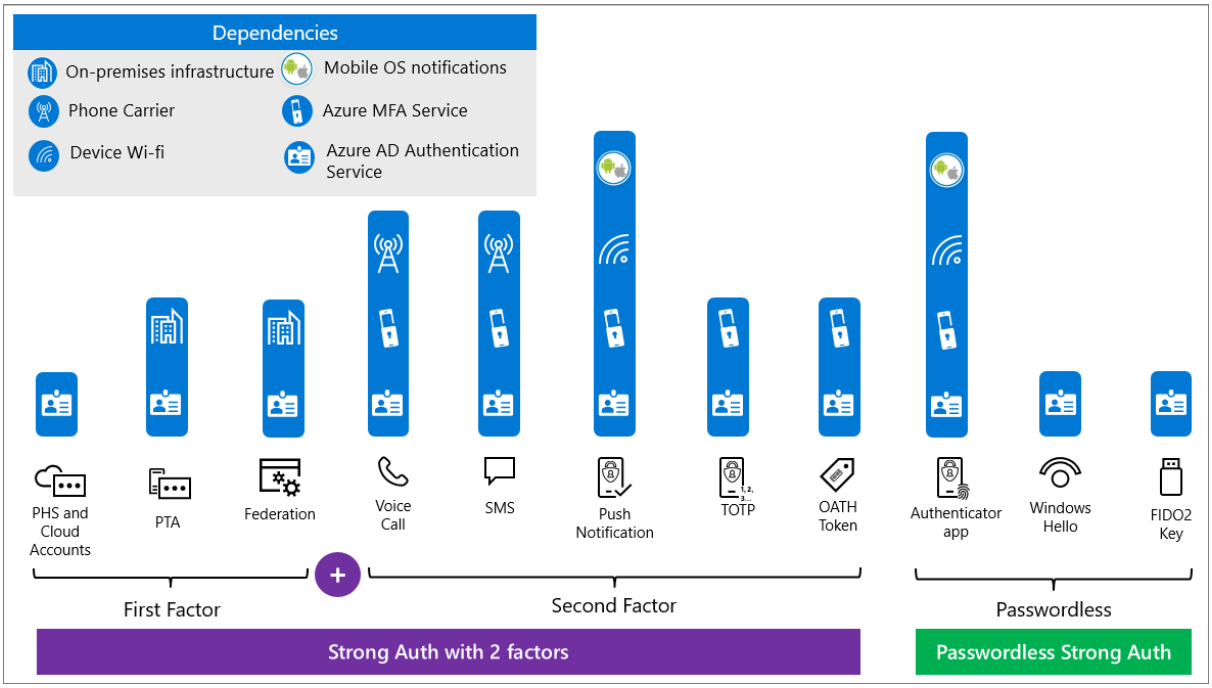
5.6 Authentication Strength

On October 19th, Alex Weinert the Director of Identity Security at Microsoft announced the public preview of authentication strength. Authentication Strength is a new Grant access control option available when you create or modify an existing Conditional Access policy.



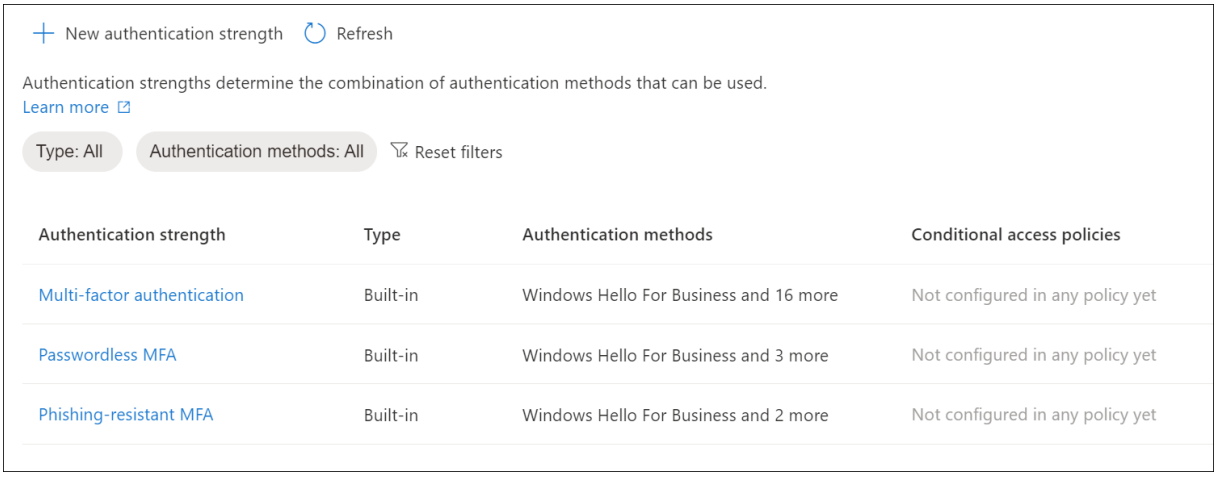
With Authentication Strength we have the option to distinct between the Multi Factor Authentication (MFA) method that can be used to fulfil the Access Control eventually granting access to the targeted resource app that you define in your conditional access policy.

Fact is today that not all MFA methods can be considered equally secure, and in my opinion customers should start moving away from these lesser secure authentication methods like SMS, phone call, but also only letting users allowing/denying an MFA request they receive in the authenticator app which gets exploited nowadays as well. If you want to know a bit more about this specific subject I want to suggest that your read this excellent article written by Jeffrey Appel: How to mitigate MFA fatigue and learn from the Uber breach for additional protection (jeffreyappel.nl)



As you can see from the screenshots below, you have the ability to choose between different Authentication strengths configurations. Three built-in configuration are provided by Microsoft, but you can also create your own. Let's go through the built-in ones first, which are:

- Multi-factor authentication
- Passwordless MFA
- Phishing-resistant MFA



5.6.1 Multi-factor authentication

Microsoft calls the Multi-factor authentication authentication strength a medium assurance authentication strength that includes multi-factor, for example password + SMS.

The Multi-factor authentication authentication strength when used allows the following methods:

- Windows Hello for Business
- FIDO2 Security Key
- Certificate Based Authentication (Multi-factor)
- Microsoft Authenticator (Phone Sign-in)
- 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
- Password + Voice
- Federated Multi-Factor
- Federated Single Factor + Microsoft Authenticator (Push Notification)
- Federated Single Factor + Software OATH token
- Federated Single Factor + Hardware OATH token
- Federated Single Factor + SMS
- Federated Single Factor + Voice

5.6.2 Passwordless MFA

Microsoft calls the Passwordless MFA authentication strength a high assurance authentication strength that includes methods with Cryptographic keys, for example FIDO2 security key.

The Passwordless MFA authentication strength when used allows the following methods:

- Windows Hello for Business
- FIDO2 Security Key
- Certificate Based Authentication (Multi-Factor)
- Microsoft Authenticator (Phone Sign-in)

5.6.3 Phishing-resistant MFA

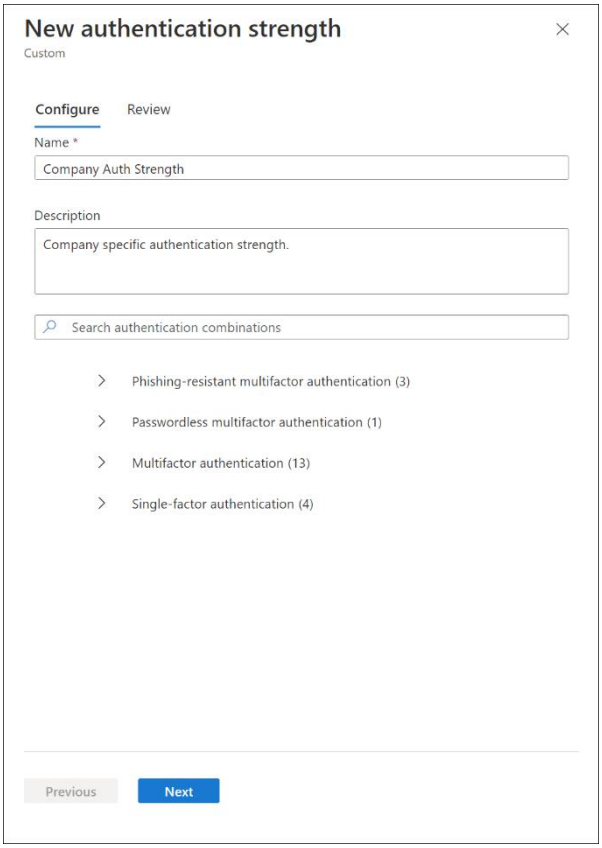
Microsoft calls the Phishing-resistant MFA authentication strength a method that is phishing resistant that includes methods like FIDO2 and Windows Hello for Business

The Phishing resistant MFA authentication strength when used allows the following methods:

- Windows Hello for Business
- FIDO2 Security Key
- Certificate Based Authentication (Multi-Factor)

5.6.4 Creating your own Authentication Strength Method

You can also define your own Authentication Strength method(s), by clicking on "+ Authentication Strength" which will start the New authentication strength wizard.



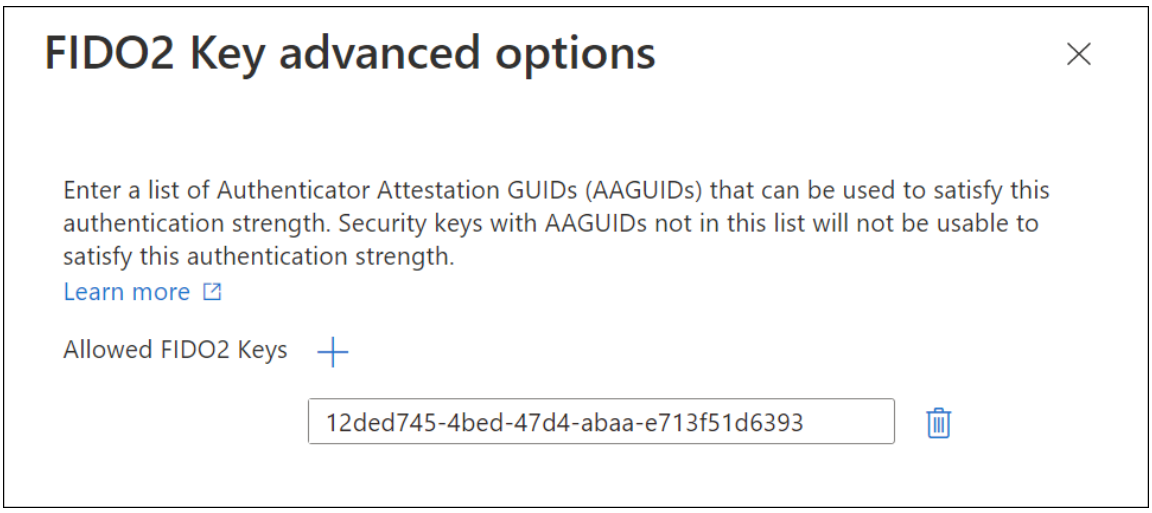
In the custom authentication strength wizard you can choose methods from the following categories:

- Phishing-resistant multifactor authentication
    - Windows Hello for Business

    - FIDO2 Security Key
    - Certificate Based Authentication (Multi-factor)
- Passwordless multifactor authentication
    - Microsoft Authenticator (Phone Sign-in
- Multifactor authentication
    - 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
    - Password + Voice
    - Federated Multi-Factor
    - Federated Single Factor + Microsoft Authenticator (Push Notification)
    - Federated Single Factor + Software OATH token
    - Federated Single Factor + Hardware OATH token
    - Federated Single Factor + SMS
    - Federated Single Factor + Voice
- Single-factor authentication
    - Certificate Based Authentication (Single Factor)
    - SMS
    - Password
    - Federated Single-Factor

5.6.4.1 Configure specific allowed FIDO2 keys

For the FIDO2 Security key you can even specify the allowed FIDO2 keys by specifying their Authenticator Attestation GUIDs (AAGUIDs). In the example below I'm configuring the Feitan AllinPass Fido 2 as allowed FIDO2 method



5.6.5 Authentication Strength use cases

The Authentication Strength option allows for all kinds of new scenario's which can be accomplished using Conditional Access.

Personally I would start with making sure that Administrative accounts can only sign in using Password-less MFA options, removing legacy MFA factors like SMS and Phone voice call if still in use

and you are not able to turn those off on the global level. Another interesting option is to combine Authentication Strength with Authentication Context, so that you can require a specific MFA method, when a SharePoint site with a specific sensitivity label gets accessed, or when a certain session control policy is triggered in Microsoft Defender for Cloud Apps (MDA).

https://www.vansurksum.com/2021/06/23/a-first-look-at-azure-ad-conditional-access-authentication-context/

We could also leverage authentication strength when user risk or sign-in risk as part of Azure AD identity protection is High. Another option is to use Authentication Strength in combination with cross-tenant settings as configured in Azure AD.

5.6.6 Authentication Strength in action

Before you start exploring Authentication Strength, make sure to read the known issues, which are documented here: Overview of Azure Active Directory authentication strength (preview) - Known issues.

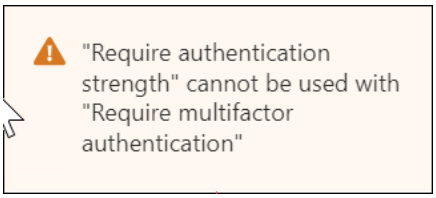
In my case, I tried building the following scenario:

See what happens if an Administrator role eligible user logs in with an authentication method, which isn't supported as an authentication method after elevating to its administrator role using Privileged Identity Management (PIM).

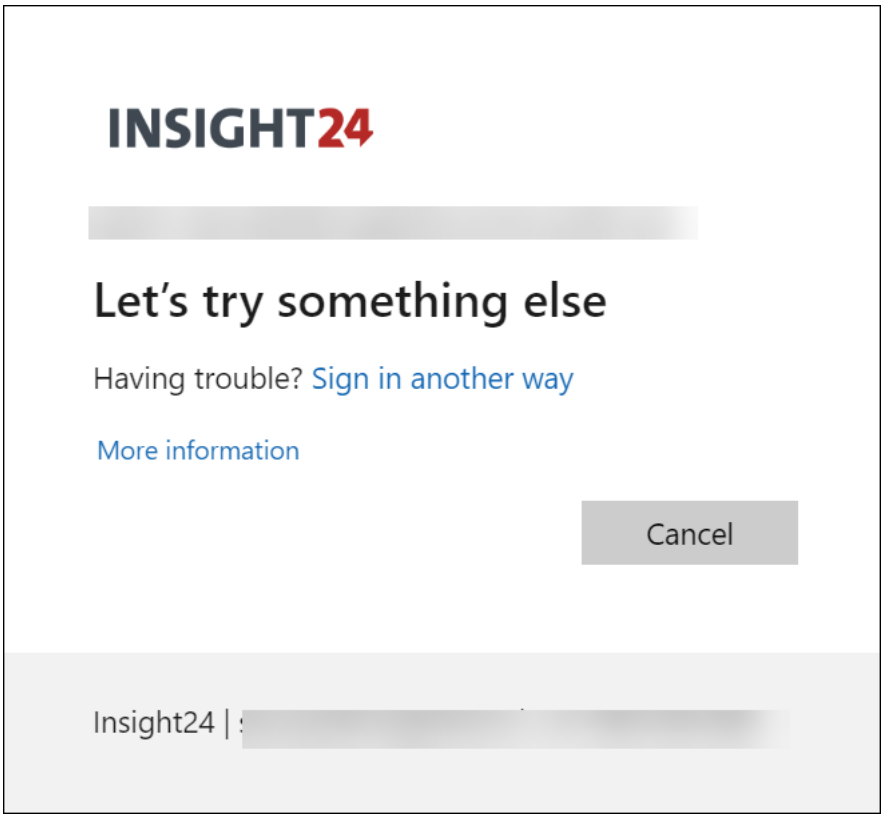
While building this solution I learned the following:

2. If for example SMS is turned off in the Global Settings, it's not available for the user to use if it's included in the Authentication Strength policy.

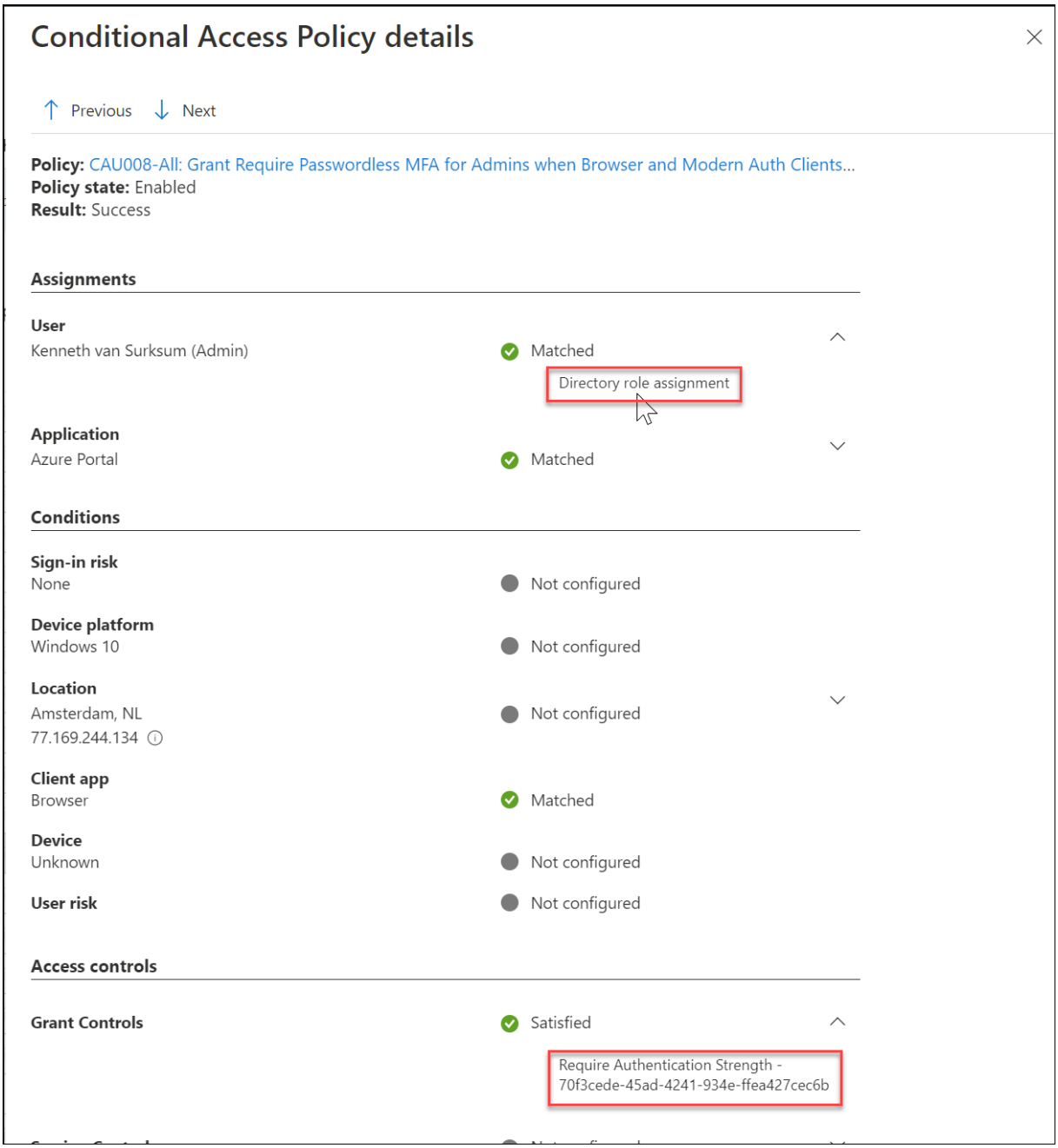
3. It's not possible to combine Multi Factor Authentication and Authentication Strength as grant controls in your Conditional Access policy, this is also noted when you create the policy.



4. You cannot combine Authentication Strength as a grant access control, while still having applicable Conditional Access policies using the "Multi Factor Authentication" access control. All applicable policies must use the Authentication Strength method, if one of them is using MFA you will receive a message similar like the one below.



6. When having multiple Conditional Access policies setting Authentication Strength, but with different authentication strengths set, the outcome can become unreliable. Say for example you have One CA policy with the "Multi-factor Authentication" authentication strength and another with the "Passwordless MFA" authentication strength, and both are applicable to the scenario, the outcome of the applied authentication strength can differ on a per scenario basis, making the use case unreliable.
7. I also had issues with using the Authentication App, when making the Passwordless MFA default option available to the CA policy requiring MFA for the Admin roles. In this case after elevating my rights, I was prompted for MFA but couldn't use the Phone sign-in which I configured on my admin account in the Microsoft Authenticator App, resulting in the "Let's try something else" method. I could use another method (my configured FIDO2 security key) successfully luckily. The only way for me to make this scenario work was to make the a new Authentication Strength policy and include the Password + Microsoft Authenticator (Push Notification) option as well.



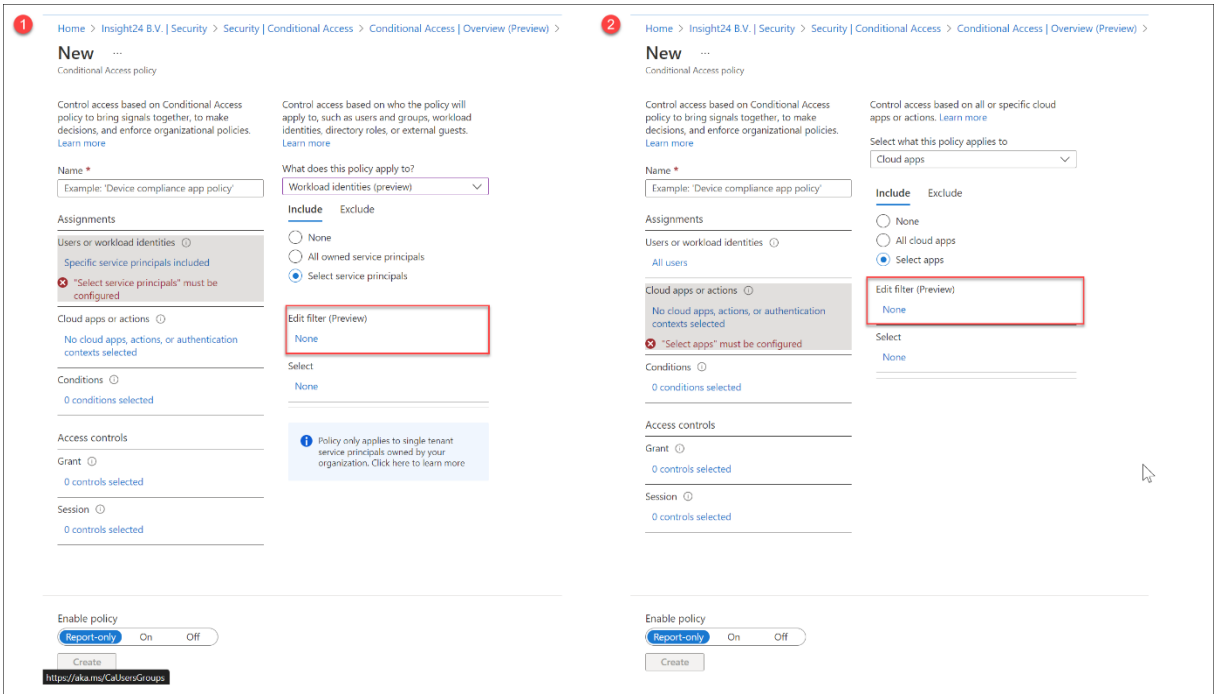
5.6.7 Authentication Strength Conclusion

Authentication Strength is a welcome addition to the Conditional Access Grant access control options already available and will eventually replace the current "Require MFA" option. Most ideally you want to configure your Conditional Access policies in such a way that all MFA required policies are enforced in the most secure way. Authentication Strength can help to setup a phased approach in order to move your users to these new secure ways of accessing your company apps and data, and provide you with options to enforce stronger authentication methods in certain use cases.

Be careful though with the current caveats causing authentication loops or , and thoroughly test your modified/new Conditional Access policies before deploying this new grant control.

5.7 Conditional Access filters for Apps and Workload Identities

On October 26th, Alex Weinert the Director of Identity Security at Microsoft announced the public preview of Conditional Access filters for apps. With filters for apps, we can use custom security attributes to "tag" Enterprise Applications and Service Principals and use that tagging when targeting your workload identities and Apps in Conditional Access.



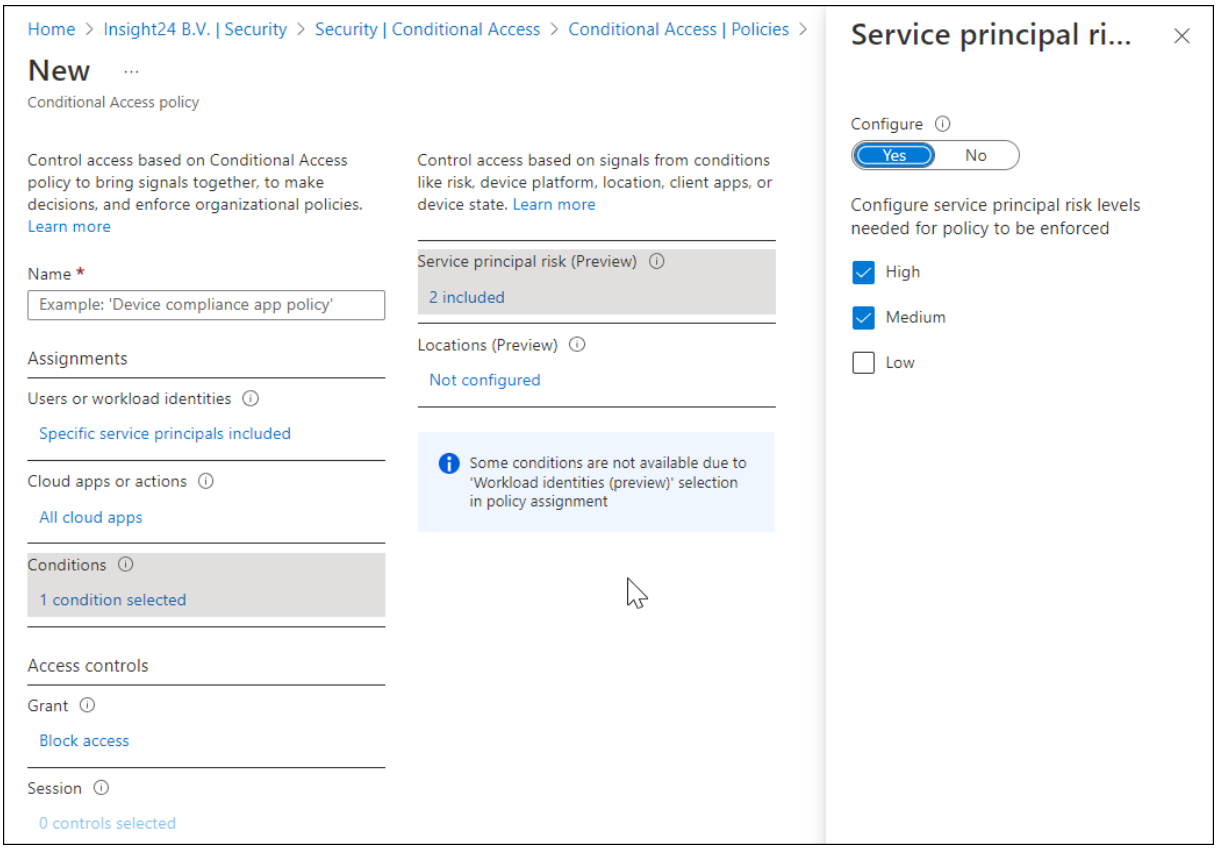
5.7.1 Conditional Access filters for Apps and Workload Identities use cases

I think the best feature of the filtering for apps and workload identities is that it allows you to configure your Conditional Access policy once, and adapt it based on the filtering being used. By doing this you prevent mistakes that could happen when modifying the Conditional Access policy. Another advantage is that you can delegate the "tagging" to another group of administrative users, without the need for those users to be able to create of modify Conditional Access policies.

Please note that Microsoft makes the following comment about scoping Conditional Access policies to service principals: *In public preview, you can scope Conditional Access policies to service principals in Azure AD with an Azure Active Directory Premium P2 edition active in your tenant. After general availability, additional licenses might be required.*

Using this filter functionality we can implement some new use cases like:

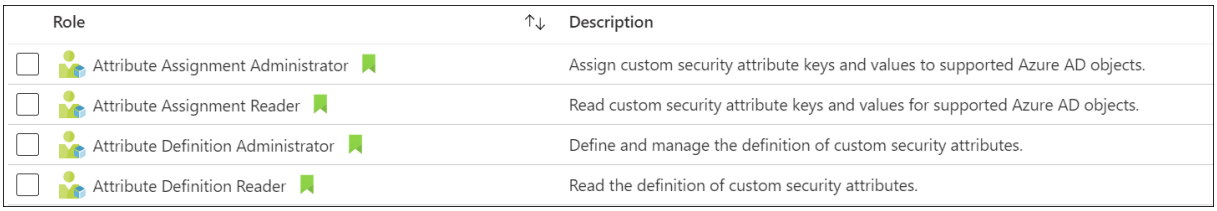
- Create a specific Conditional Access policy for all Apps which are tagged with a tag UsedByDepartment and value Finance
- Create a specific Conditional Access policy only allowing "tagged" workload identities to be used from trusted locations
- Create a specific Conditional Access policy, which blocks medium and high "Service Principal risk" for "tagged" workload identities



5.7.2 Rights needed for the Custom Security Attributes

Before you can define, or assign Security Attributes, you must be authorized via an Azure AD role. For this the following Azure AD roles are available:

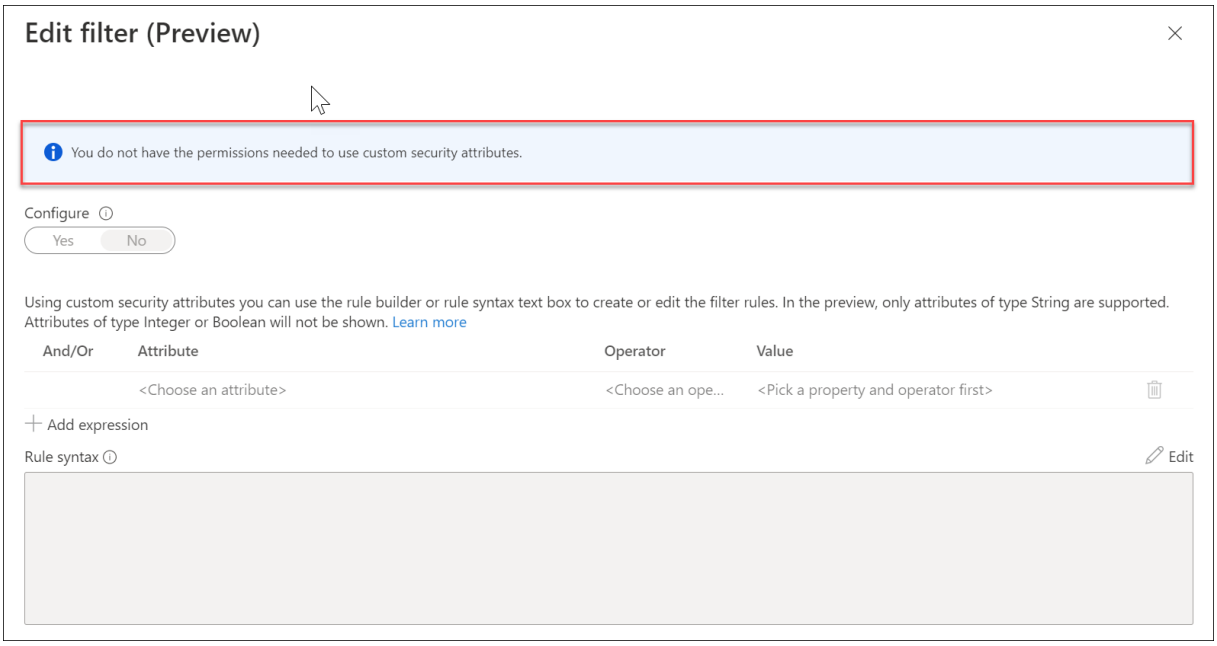
- Attribute Definition Administrator
- Attribute Definition Reader
- Attribute Assignment Administrator
- Attribute Assignment Reader



What's interesting here is that the rights of these roles are not available to the Global Administrator role.

The Attribute Definition Administrator can create definitions of the attributes, which can then be used by an Attribute Assignment Administrator and put them on Apps, Workload Identities, but also to use these definitions in an Conditional Access policy. If you have the rights to create/modify Conditional Access policy but are not a Attribute Assignment Administrator you cannot assign the filtering. I also did some further testing, and could assign security attributes in a Conditional Access policy using the Attribute Definition Reader role, but for example not assign any attributes to an

account in Azure AD. If the Attribute Definition role is not available for your account, you won't be able to assign at all.



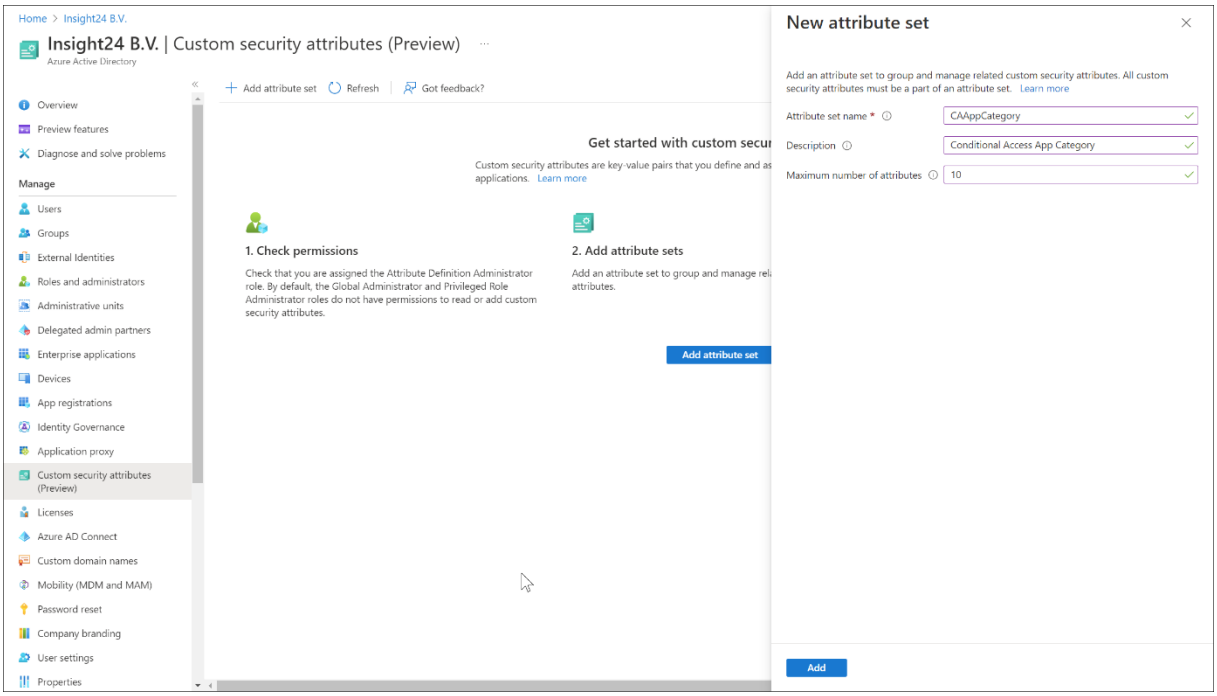
5.7.3 Creating custom Security Attributes

You can create custom security attributes, by going to the "Custom security attributes (Preview)" section in the Azure AD Admin portal. Creating custom security attributes consists of the following steps:

1. Create attribute set
2. Specify attribute

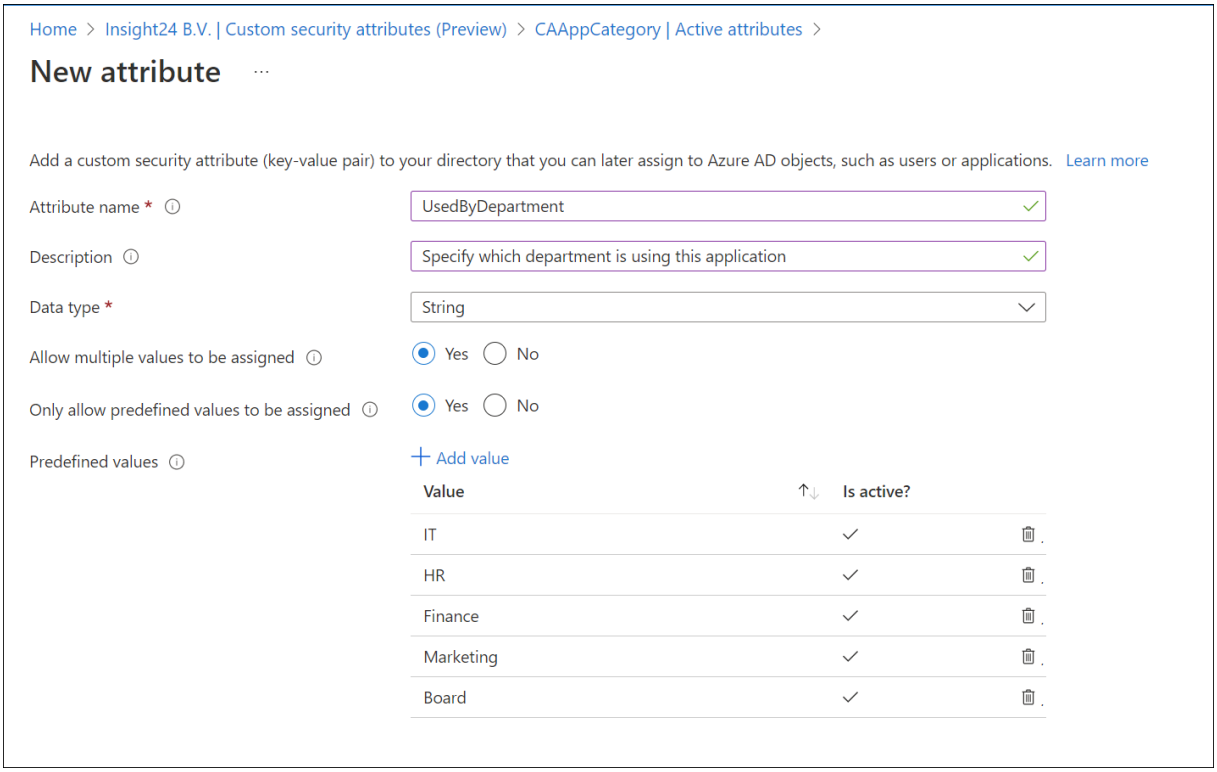
5.7.3.1 Create Attribute set

The first thing you must do, is create an Attribute Set. An attribute set is a collection of related attributes. You can create a new attribute set, by clicking on the "+ Add attribute set", after which you must define an Attribute set name, optionally provide a description and specify the maximum number of attributes.

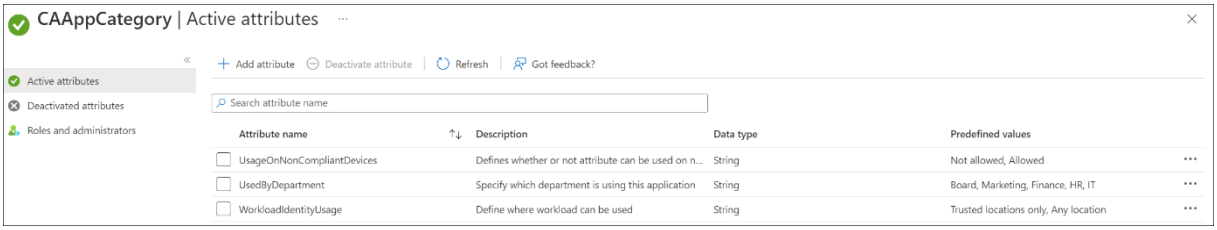


5.7.3.2 *Specify Attribute*

Within an Attribute Set you can specify custom security attributes, which is a key-value pair. You are required to specify the name of the Attribute, it's data type (String, Boolean or Integer), whether multiple values can be assigned, and whether you only allow predefined values to be assigned. Last but not least, you can create predefined values, so that only those values can be used while assigning the attributes.



Eventually you can create multiple attributes, with predefined values within an Attribute set



5.7.4 Assigning Security Attributes

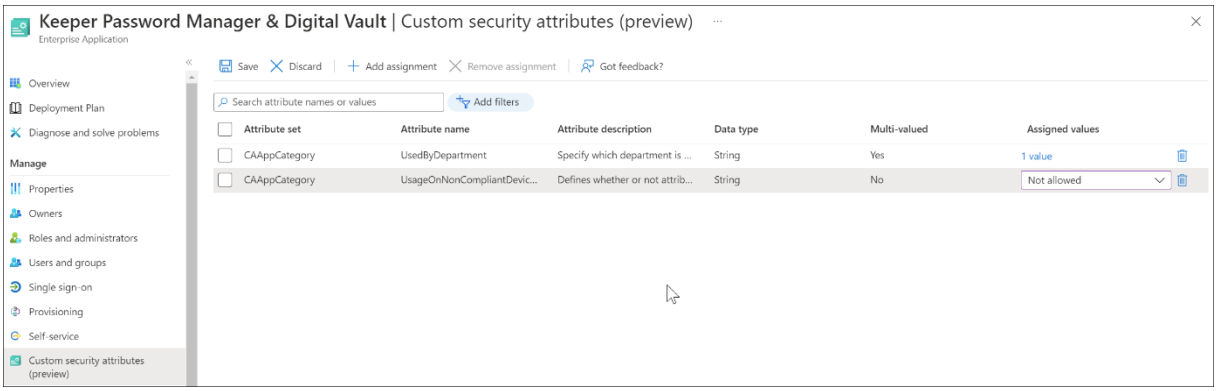
Assigning the security attributes must be performed on the resource (the App or the workload identity) and you should reference the security attribute in a filter in your Conditional Access policy.

- Assigning security attributes to an Enterprise Application
- Specify security attributes in a filter for Apps in your Conditional Access policy
- Specify security attributes in a filter for workload identities in your Conditional Access policy

5.7.5 Assigning security attributes to an Enterprise Application

An Attribute Assignment Administrator can assign the predefined security attributes as part of an attribute set to Enterprise Applications. In the example below, we tag the "Keeper Password Manager & Digital Vault" Enterprise Application with the following attributes:

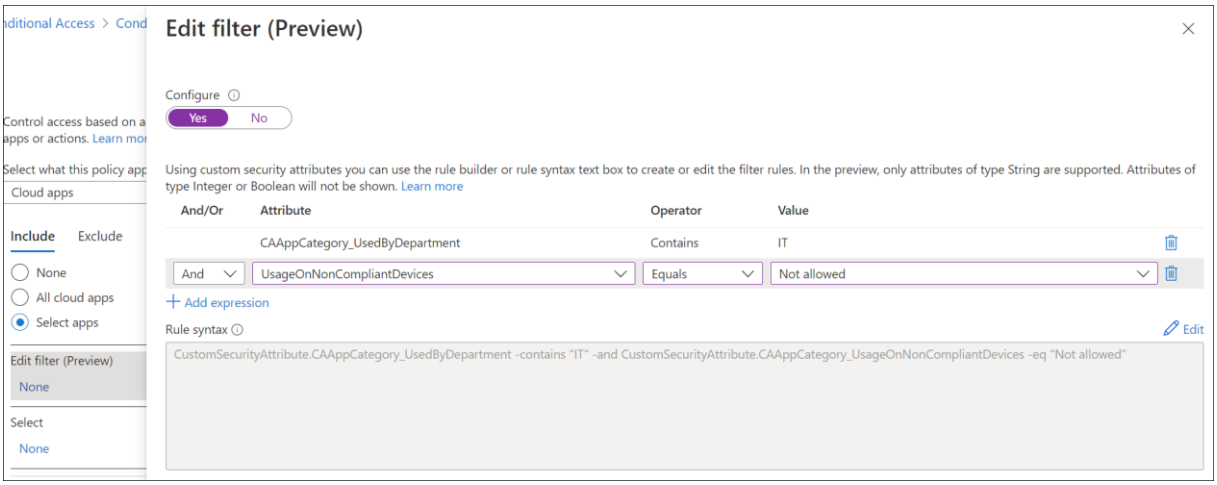
UsedBydepartment: IT
UsageOnNonCompliantDevices: Not allowed



Keep in mind that only an Conditional Access policy can only be applied to single tenant service principals that have been registered in your tenant. Third party SaaS, multi-tenanted apps and Managed Identities are out of scope, but can be provisioned with security attributes.

A single-tenant application has only one service principal (in its home tenant), created and consented for use during application registration. A multi-tenant application also has a service principal created in each tenant where a user from that tenant has consented to its use.

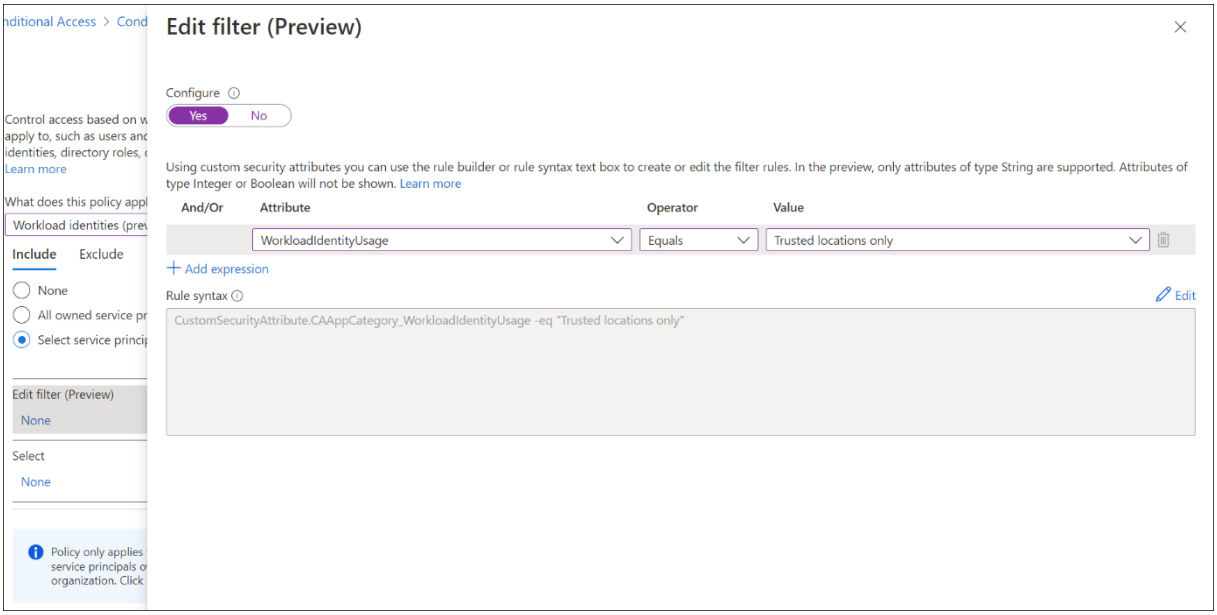
5.7.6 Specify security attributes in a filter for Apps in your Conditional Access policy



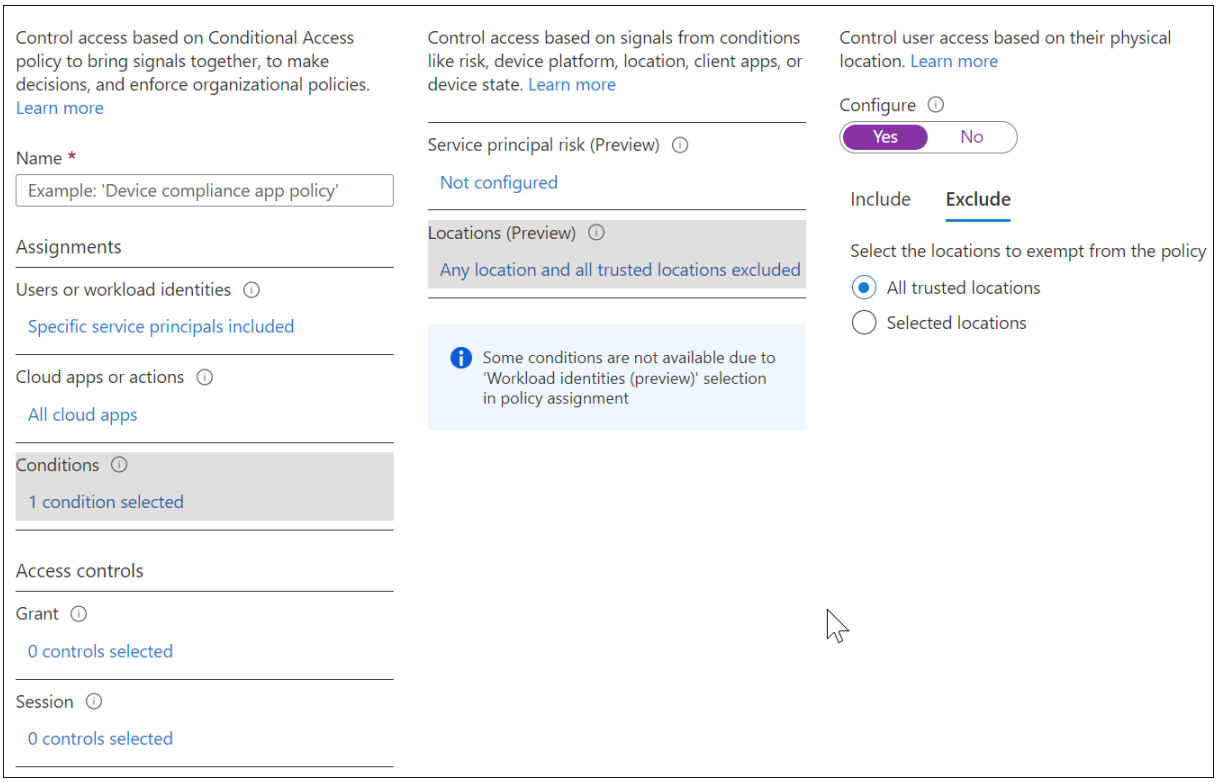
Using this filter we can create an Conditional Access policy which only allows Compliant devices to access the app.

5.7.7 Specify security attributes in a filter for workload identities in your Conditional Access policy

Policy can be applied to single tenant service principals that have been registered in your tenant. Third party SaaS and multi-tenanted apps are out of scope. Managed identities are not covered by policy.



Using this filter will allow us to create a Condition which will block access to the Service Principal, except from a trusted location.



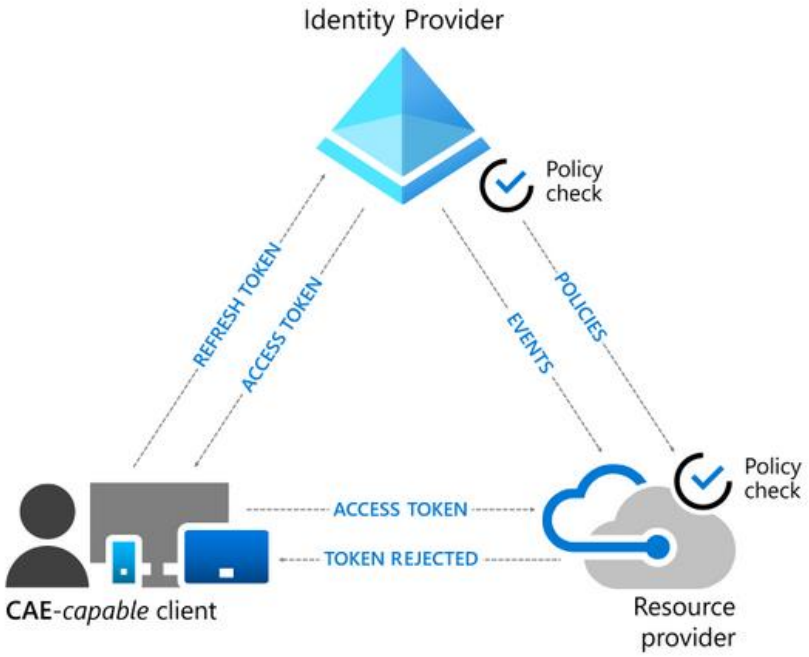
5.7.8 Conditional Access filters for Apps and Workload Identities Conclusion

The filters for Apps and Workload Identities functionality can really help to further mature your Conditional Access implementation. By delegating control to the Cloud Application administrators to use the predefined Security Attributes to tag their Enterprise Applications and Service Principals, we have the option to define generic Conditional Access policies for the different use cases. (for example, only allow Finance applications on Compliant devices)

With options comes more complexity though, be very careful with that. Make sure that you have a solid and stable Conditional Policy framework in place before starting with these kind of scenario's.

5.8 Continuous Access Evaluation

In April 2020 Alex Weinert, Director of Identity Security at Microsoft announced that Microsoft was working on moving towards real time policy and security enforcement. The first implementation for this move is now available as an option you can enabled within Azure AD, called Continuous access evaluation (CAE).



Continuous access evaluation allows for a quicker response by forcing an access token refresh in case of a certain events taking place. In this version of the preview the following events will be supported:

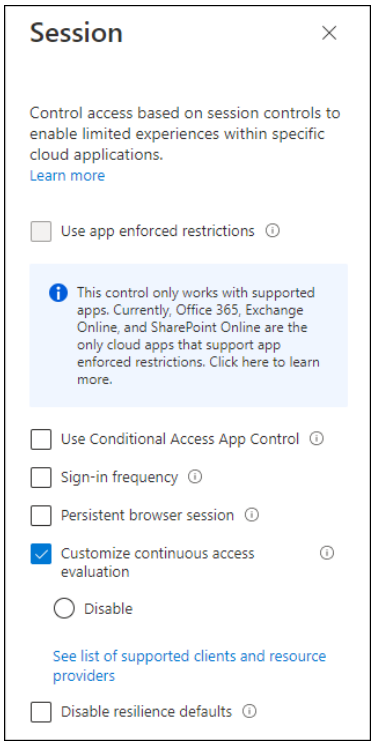
- User Account is deleted or disabled
- Password for a user is changed or reset
- MFA is enabled for the user
- Admin explicitly revokes all Refresh Tokens for a user
- High user risk detected by Azure AD Identity Protection (not supported by SPO)

Microsoft already announced that in the future these events will be extended to include for example location and device state changes. So, we can for example request MFA when the user switches from the trusted network to an untrusted network or block access when the device is becoming non-compliant.

Up until now, tokens for these scenarios were refreshed every hour, this can now be reduced to near real-time by using CAE. There is also an option to configure the lifetime for tokens overall, using configurable token lifetimes, which is also in preview at this moment, see: Configurable token lifetimes in Microsoft identity platform (preview). These configurable token lifetimes can be used in Conditional Access policies using the Sign-in frequency session controls, see: Configure authentication session management with Conditional Access, Sign-in frequency will be honoured with or without CAE.

Continuous access evaluation is implemented by enabling services (resource providers) to subscribe to critical events in Azure AD so that those events can be evaluated and enforced near real time

Continuous access evaluation is enabled by default, and can be disabled by using a session control in Conditional Access.



Keep in mind that Not all client app and resource provider combinations are supported, therefore I advise you to read the following article to get a better idea on what should be working and what not: [Continuous access evaluation in Azure AD - Microsoft Entra | Microsoft Learn]

6 Implementing Conditional Access

Before you start implementing your Conditional Access policies you should define an implementation strategy, some things to consider are:

1. Verify whether Legacy/Basic authentication is still being used. Even though Microsoft is now turning off legacy/basic authentication for some of the protocols which you can use to access Exchange Online, there are still options to connect to SharePoint Online using legacy/basic authentication or use Authenticated SMTP using basic authentication.

2. I strongly advise to configure Azure AD Diagnostic settings in such a way that Azure AD sign-in logs (containing information about Conditional Access applicability) is forwarded to a Log Analytics workspace. Even with the default retention of 7 days, you still get added value since the Workbooks functionality comes available as well. Within the workbooks some of the workbooks cover Conditional Access applicability.

3. Create 2 break glass accounts, these accounts, which are global administrator should have complex passwords and will be excluded from any conditional access policy created and must have MFA disabled (or at least on one of the two accounts). More information about creating break glass accounts can be found here: Manage emergency access accounts in Azure AD.

4. For each conditional access policy created, we will create an exclusion group, so that we can deal with exceptions in our environment. These exception groups will be setup with Access review functionality (if available) to make sure that the memberships of these groups are evaluated on a regular basis.

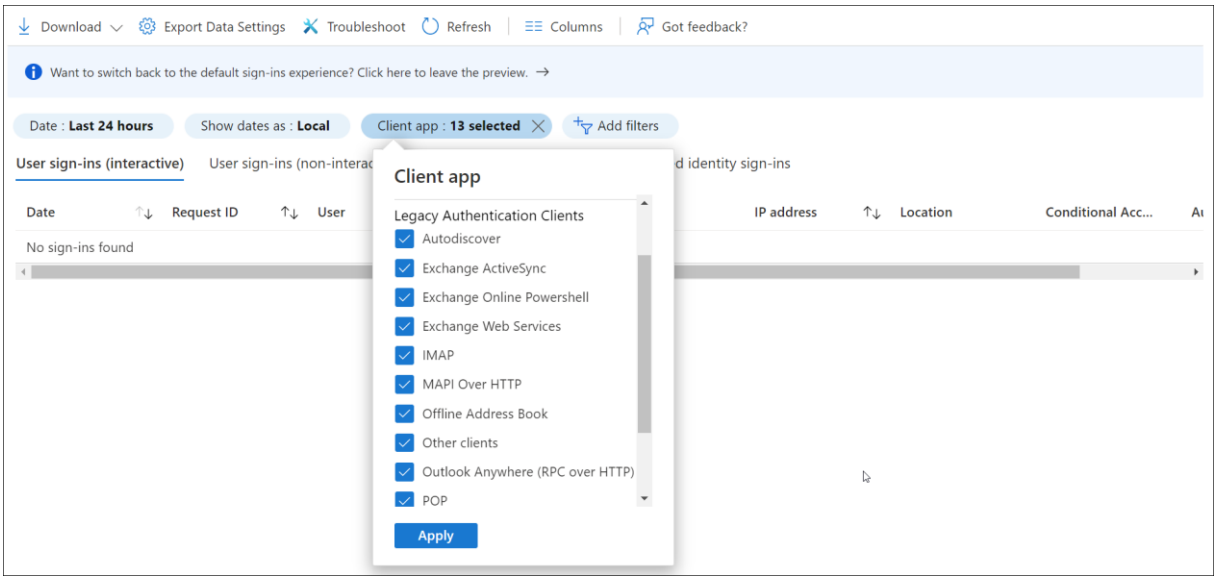
Based on this we can define the following steps needed to implement your Conditional Access policies in the most ideal way.

Step 1: Check if legacy/basic authentication is still being used withing your environment.

You can check whether legacy/basic authentication is still in use in several ways depending on your tenant configuration. The fastest way to check for legacy/basic authentication usage is to use the Azure AD sign-in logging.

Go to the Azure AD Admin Portal -> Sign-in logs. From there select Add filter -> Client App. From the list of Client App select all the options available under “Legacy Authentication Clients”. There protocols are: *Autodiscover, Exchange ActiveSync, Exchange Online PowerShell, Exchange Web Services, IMAP, MAPI over HTTP, Offline Address Book, Other clients, Outlook Anywhere (RPC over HTTP), POP, Reporting Web Services, SMTP and Universal Outlook.*

Some of the protocols in the list under Legacy Authentication Clients, also support Modern Authentication (like for example IMAP and POP), but when those protocols are being used, they will report as a “Mobile Apps and Desktop clients” in the reporting.



Step 2: Configure Azure AD Diagnostic settings

By configuring Azure AD Diagnostic settings we have the ability to store Azure AD sign-in and Audit logging in an Azure Log Analytics workspace, were we can optionally extend the storage of these logs past the standard 30 days offered in the Azure AD portal. In some cases, the Log Analytics workspace configured can serve as a basis to run your Microsoft Sentinel instance.

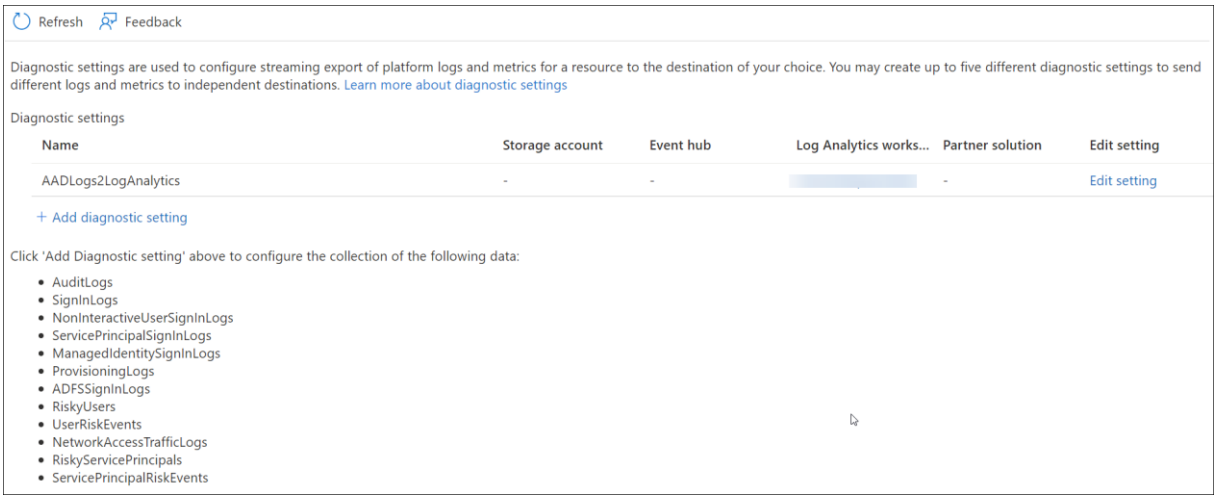
You can configure the Diagnostic settings by going to the Azure AD portal -> Diagnostic settings. From there you can use the "+ Add diagnostic setting" to configure a new connection towards an external service where the logs will be sent. You can choose the following destinations:

- Send to Log Analytics Workspace
- Archive to a storage account
- Stream to an event hub
- Send to partner solution

Once you configured your destination, you can also select what kind of logs need to be sent to the Log Analytics workspace, here you have the option to choose between:

- AuditLogs
- SignInLogs
- NonInteractiveUserSignInLogs
- ServicePrincipalSignInLogs
- ManagedIdentitySignInLogs
- ProvisioningLogs
- ADFSSignInLogs
- RiskyUsers
- UserRiskEvents
- NetworkAccessTrafficLogs
- RiskyServicePrincipals
- ServicePrincipalRiskEvents

For Conditional Access reporting we need all the Sign-in Logs.



Step 3: Create 2 Break glass accounts.

If you do not have them already, create 2 break glass accounts which will be excluded from any policy which could potentially block access to the Azure environment in case something goes wrong and you are being locked out of the environment.

See the following article from Microsoft for more information: [Manage emergency access accounts in Azure AD]

Make sure that the accounts are protected with Strong Authentication, for example by registering a FIDO2 Security key, and setting the account to MFA Enforced in the global MFA settings.

Also make sure that you can monitor usage of the break glass accounts, if you configured the Log Analytics integration in the previous step you have the option to configure Alert rules which can sent an email or SMS whenever a successful, but also important an unsuccessful login attempt using one of the break glass accounts is detected.

You can use the following Kusto Query Language (KQL) queries for your Alert ruling configuration.

Successful login of a break glass account

```
SigninLogs
| project UserPrincipalName, ResultType
| where UserPrincipalName contains "breakglass@customer.onmicrosoft.com"
    and ResultType == "0"
```

Unsuccessful login of a break glass account

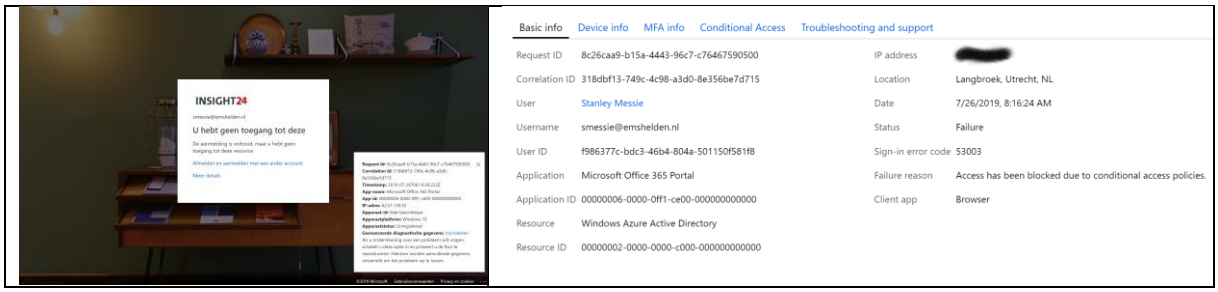
```
SigninLogs
| project UserPrincipalName, ResultType
| where UserPrincipalName contains "breakglass@customer.onmicrosoft.com"
    and ResultType != "0"
```

Step 4: Implement your own custom Conditional Access policies based on your Conditional Access design. You can also decide to start with the baseline Conditional Access policies provided by me. Make sure that for every policy that you exclude a specific group for that policy and make sure that the break glass accounts are excluded as well. Make sure that you have a decent test plan to test the policies and modify your Conditional Access policies if needed in case something does not work as expected.

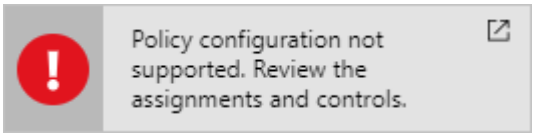
Testing conditional access policies can be quite tricky. Main reason for this is that once a policy is enabled it is not necessarily directly effective and it can take a while for it to become effective for your clients. This has to do with the fact that clients are not logging in all the time. Therefore, make sure that before you enable your Conditional Access policy that you know where you can find the logging (more on that in the next chapter) to centrally determine if something is possibly wrong. Also make sure that all company procedures are followed for a change like this.

Step 5: Create a block all policy to make sure that you do not miss anything and that holes in your conditional access strategy exist. The way this works is that you create the policy but exclude all the groups used (include and exclude groups) for the Conditional Access policies and of course exclude the break glass accounts. In this case we are **absolutely sure** that all users are covered by Conditional Access policy

Users not authorized within any of the Conditional Access groups will not be able to sign in from that point forward.



Azure has a safety feature that prevents you from creating a policy which violates the best practices for Conditional Access policies, so you cannot enable a policy for all cloud apps, for all users which denies access.



The safety feature is necessary because block all users and all cloud apps have the potential to block your entire organization from signing on to your tenant. You must exclude at least one user to satisfy the minimal best practice requirement.

Some other things to consider are:

For all the exception groups make sure that you enable the "Access review" functionality (Azure AD Premium P2 feature), for which you can find more information here:

- Use Azure AD access reviews to manage users excluded from Conditional Access policies -
- Create an access review of groups or applications in Azure AD access reviews
- Which users must have licenses?

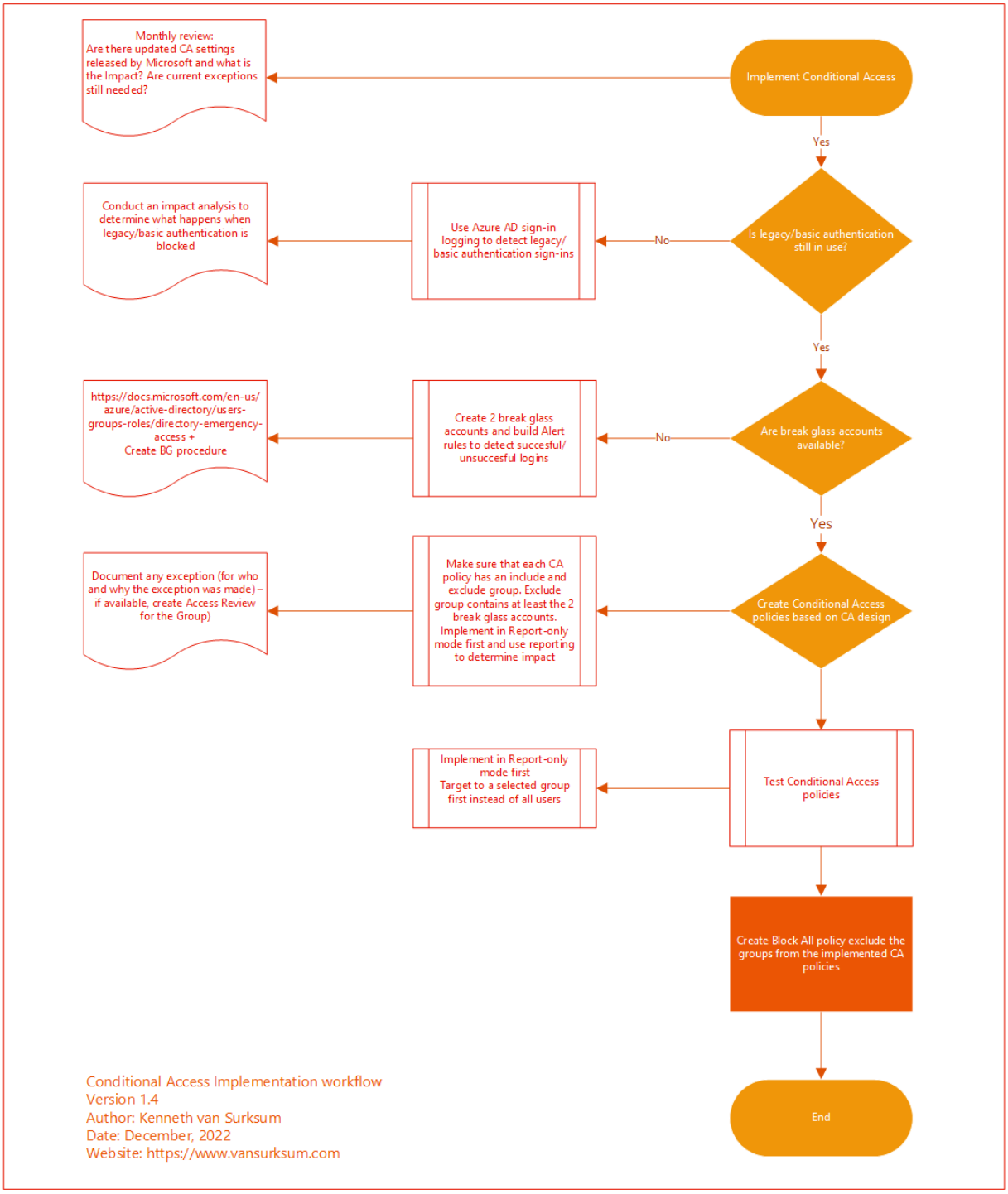
Make sure that you have defined operational procedures on what to do if certain functionality provided by Microsoft is down. A good example of this is the fact that in the past the Microsoft Multi Factor Authentication service has been down for a significant time. Having an operational procedure which allows you to make cloud apps available only from on premises when that happens without using MFA by disabling the "standard" policy and enabling a "temporary" policy might be a good idea.

Azure Active Directory Conditional Access changes on a regular basis, make sure that you have a procedure to check occasionally what is cooking on this topic. Make sure you understand what is coming, what is in preview and what is released. Invest time to determine the possible impact of these changes to the Conditional Access policies and requirements in place. You can see what's new in Azure Active Directory and on the Azure Updates webpage.

Some cloud apps have dependencies with other cloud apps, for example Microsoft Teams has dependencies of Exchange Online, SharePoint and Planner and perhaps even more.

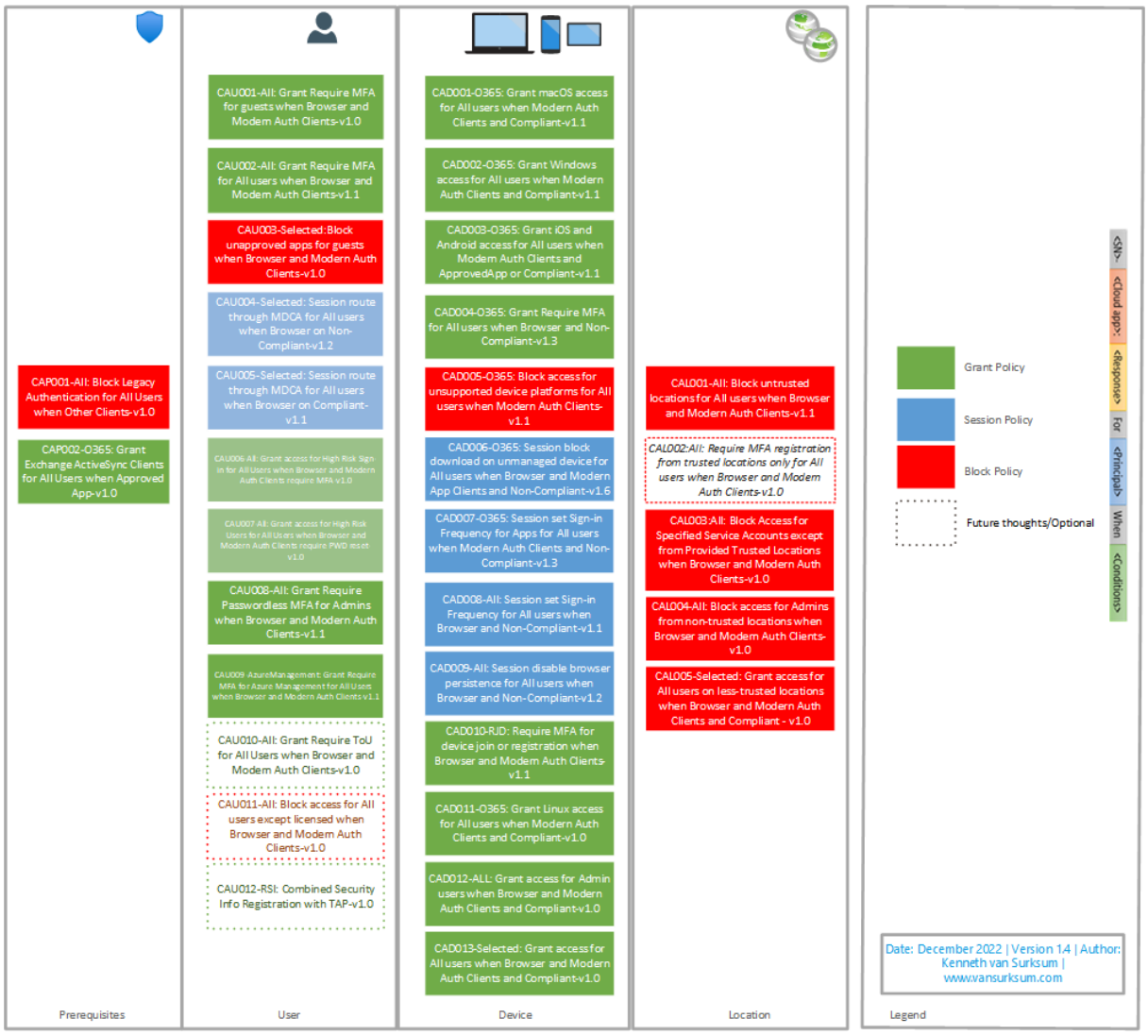
More information here: What are service dependencies in Azure Active Directory Conditional Access?

I've created a flowchart which describe the steps described above, you can download this flowchart for your reference here: Conditional Access Implementation Workflow - V1.4.pdf


Monthly review:
Are there updated CA settings released by Microsoft and what is the Impact? Are current exceptions still needed?
Implement Conditional Access
Yes
Conduct an impact analysis to determine what happens when legacy/basic authentication is blocked
Use Azure AD sign-in logging to detect legacy/ basic authentication sign-ins
No
Is legacy/basic authentication still in use?
Yes
https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-emergency-access +
Create BG procedure
Create 2 break glass accounts and build Alert rules to detect succesful/ unsuccesful logins
No
Are break glass accounts available?
Yes
Document any exception (for who and why the exception was made) – if available, create Access Review for the Group)
Make sure that each CA policy has an include and exclude group. Exclude group contains at least the 2 break glass accounts. Implement in Report-only mode first and use reporting to determine impact
Create Conditional Access policies based on CA design
Implement in Report-only mode first
Target to a selected group first instead of all users
Test Conditional Access policies
Create Block All policy exclude the groups from the implemented CA policies
End
Conditional Access Implementation workflow
Version 1.4
Author: Kenneth van Surksum
Date: December, 2022
Website: https://www.vansurksum.com

6.1 My recommended default set of policies

Once you have consensus about how you want to allow access to your company data, you can start describing your Conditional Access policies, below is an overview of the Conditional Access policies based on the functional design described in chapter 4.



As you can see, the policies are divided into several categories which I use in the naming of the policies as well. For the naming I use the Microsoft recommended naming policy as described in this article: [Set naming standards for your policies]

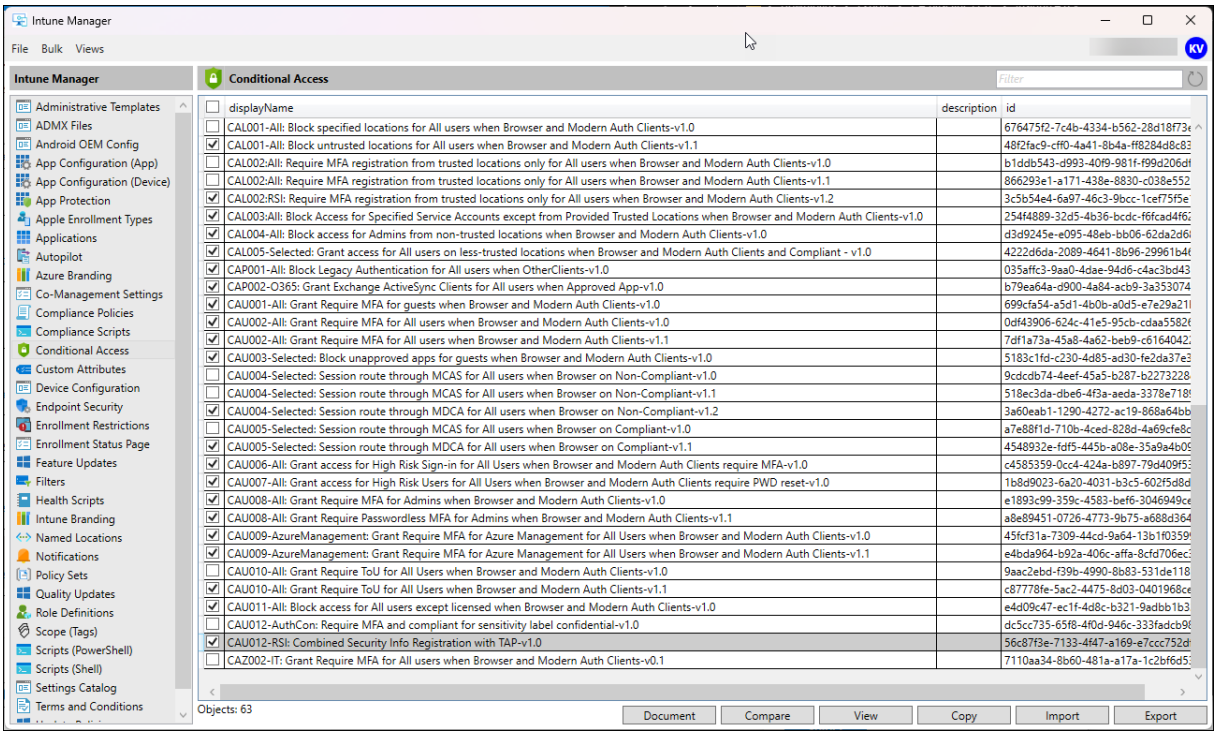
The detailed settings of the policies described, can be found in the new version of the Conditional Access Documentation spreadsheet which can be found here: [Conditional Access Policy Description-v1.4.xlsx]

For each policy, an exclusion group is created, and for each policy the group containing the break glass accounts will also be excluded from the policy.

6.1.1 Downloading and importing the baseline policies

New in this release is that the Conditional Access policies are also available in JSON format, and ready to be imported into your environment. In order to export the policies I used the amazing Intune Manager tool written by Mikael Karlsson

With this tool you should be able to import the policies into your environment, the policies themselves can be found on my GitHub page here: https://github.com/kennethvs/cabaseline202212



There is also the option to deploy the policies using Terraform, Robert Brandsø from Norway did an amazing job using my CA policies as an input for an automation solution which you can find here: https://github.com/robertbrandso/terraform-azuread-recommended-conditional-access-policies

6.1.2 Versioning

Each policy has a version number in it, by doing so we can update policies with a new version, test this new policy on a select set of users and/or Report-Only mode and then make the switch by turning the old version off, and implementing the new one.

6.1.3 Prerequisites

The first category are the prerequisites and contains two policies.

6.1.3.1 CAP001-All: Block Legacy Authentication for All users when OtherClients-v1.0

This policy blocks all Legacy authentication when clients not supporting Modern Authentication are being used. For more information about this policy, please read the documentation from Microsoft: How to: Block legacy authentication to Azure AD with Conditional Access

Keep in mind that just disabling Legacy authentication in an existing environment isn't a good idea. The end-users might still use applications only capable of performing legacy authentication and they might need your help first to transition to apps which support modern authentication. Please read

the following article for more context and how to start your own project to phase out legacy authentication. [Microsoft is going to disable basic/legacy authentication for Exchange Online. What does that actually mean and does that impact me?]

CAP001-All: Block Legacy Authentication for All users when OtherClients-v1.0

Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	*Conditions*	*Grant*	*Session*
All Users	All	Client Apps: Other	Block	
Except				
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAP001_Exclude				

6.1.3.2 CAP002-O365: Grant Exchange ActiveSync Clients for All users when Approved App-v1.0

Based on the functionalities provided, there is no use case to keep using Exchange Active Sync, that is the reason that we block Exchange Active Sync clients as well. Even though the policy grants, it blocks access to EAS clients because an Approved App is needed. (Outlook in our case). By using this mechanism, users still using EAS receive a message that they must transition to an application supporting Modern Authentication. If we would use a block policy, the user simply will not get any email messages anymore.

CAP002-O365: Grant Exchange ActiveSync Clients for All users when Approved App-v1.0

Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	All	Client Apps: Exchange Active Sync	Grant	
Except				
AAD_UA_ConAcc-Breakglass			Require Approved App	
AAD_UA_CAP002_Exclude				

6.1.4 User

The second category contains the policies for users and contains nine policies. Some of these policies require an Azure AD Premium P2 subscription, like CAU005 and CAU006 which require Azure AD Identity Protection and CAU004 which requires Microsoft Defender for Cloud Apps(MDA).

6.1.4.1 CAU001-All: Grant Require MFA for guests when Browser and Modern Auth Clients-v1.0

This one makes sure that guest users are required to use Multi Factor Authentication (MFA) when accessing resources that you host.

CAU001-All: Grant Require MFA for guests when Browser and Modern Auth Clients-v1.0

Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
Guest and External Users	All	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except				
AAD_UA_ConAcc-Breakglass			Require multi-factor authentication	
AAD_UA_CAU001_Exclude				

6.1.4.2 CAU002-All: Grant Require MFA for All users when Browser and Modern Auth Clients-v1.0

This policy requires each user to use MFA when accessing cloud apps.

CAU002-All: Grant Require MFA for All users when Browser and Modern Auth Clients-v1.0

Assignments			Access Controls	
Users	Cloud Apps	Conditions	Grant	Session
All Users	All	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except				
Guest and External Users			Require multi-factor authentication	
And				
AAD_AA_ConAcc-Breakglass				
AAD_AA_CAU002-Exclude				

6.1.4.3 CAU002-All: Grant Require MFA for All users when Browser and Modern Auth Clients-v1.1 (Optional)

This version of the policy must be used if you want to leverage Authentication Strength, we also found an issue where exclusion of Microsoft Intune and Microsoft Intune Enrolment was necessary in order to make device sync work on a Windows 365/Cloud PC machine.

CAU002-All: Grant Require MFA for All users when Browser and Modern Auth Clients-v1.1

Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	All	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except	exclude:			
Guest and External Users	Microsoft Intune		Require authentication strength (Preview):	
And	Microsoft Intune Enrollment		Multi Factor Authentication	
Global Administrator User Administrator SharePoint Administrator Security Administrator Password Administrator Helpdesk Administrator Compliance Administrator Exchange Administrator Conditional Access Administrator Billing Administrator Authentication Administrator				
And				
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAU002_Exclude				

6.1.4.4 CAU003-Selected: Block unapproved apps for guests when Browser and Modern Auth Clients-v1.0

With this policy, you can create a list of Cloud Apps for which guest users are not allowed to use them. These apps can be apps containing sensitive company data for example.

CAU003-Selected: Block unapproved apps for guests when Browser and Modern Auth Clients-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
Guest and External Users	<Selected>	Client Apps: Browser Mobile Apps and Desktop Clients	Block	
Except				
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAU003_Exclude				

6.1.4.5 CAU004-Selected: Session route through MDA for All users when Browser on Non-Compliant-v1.2

With this policy you can route the session through MDA and its reverse-proxy capability, allowing you to either block downloads or monitor the session for strange behavior. In this example we block downloads for Cloud Applications specified if the device is not compliant.

CAU004-Selected: Session route through MDCA for All users when Browser on Non-Compliant-v1.2				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
	<Selected>	Client Apps: Browser		Use Conditional Access App Control: Block downloads (Preview)
Except		Filter for devices: Exclude isCompliant Equals True Or trustType Equals Hybrid Azure AD Joined		
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAU004_Exclude				

6.1.4.6 CAU005-Selected: Session route through MDA for All users when Browser on Compliant-v1.1

With this policy you can route the session through MDA and its reverse-proxy capability, allowing you to either block downloads or monitor the session for strange behavior. In this example we just monitor the session on devices which are compliant.

CAU005-Selected: Session route through MDCA for All users when Browser on Compliant-v1.1				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
	<Selected>	Client Apps: Browser		Use Conditional Access App Control: Monitor (Preview)
Except		Filter for devices: Include isCompliant Equals True Or trustType Equals Hybrid Azure AD Jonied		
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAU005_Exclude				

6.1.4.7 CAU006-All: Grant access for High Risk Sign-in for All Users when Browser and Modern Auth Clients require MFA-v1.0

This policy will require MFA for sign-ins flagged as high-risk by Azure AD identity protection. See my article: Azure AD Identity Protection deep dive for more information.

CAU006-All: Grant access for High Risk Sign-in for All Users when Browser and Modern Auth Clients require MFA-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	All	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except		Sign-in Risk: High		
AAD_UA_ConAcc-Breakglass			Require multi-factor authentication	
AAD_UA_CAU006_Exclude				

6.1.4.8 CAU007-All: Grant access for High-Risk Users for All Users when Browser and Modern Auth Clients require PWD reset-v1.0

This policy will grant access for High-Risk users after but only after they have reset their password. This functionality is also provided by Azure AD Identity Protection.

CAU007-All: Grant access for High Risk Users for All Users when Browser and Modern Auth Clients require PWD reset-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	All	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except		User Risk: High		
AAD_UA_ConAcc-Breakglass			Require Password Change	
AAD_UA_CAU007_Exclude				

6.1.4.9 CAU008-All: Grant Require MFA for Admins when Browser and Modern Auth Clients-v1.0

This policy makes sure that Admins must always use MFA before signing into any cloud application.

CAU008-All: Grant Require MFA for Admins when Browser and Modern Auth Clients-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
Global Administrator User Administrator SharePoint Administrator Security Administrator Password Administrator Helpdesk Administrator Compliance Administrator Exchange Administrator Conditional Access Administrator Billing Administrator Authentication Administrator	All	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except				
AAD_UA_ConAcc-Breakglass			Require multi-factor authentication	
AAD_UA_CAU008_Exclude				

6.1.4.10 CAU008-All: Grant Require Passwordless MFA for Admins when Browser and Modern Auth Clients-v1.1 (Optional)

This policy is required when you want to start using Authentication Strength, requiring a step-up authentication is necessary

CAU008-All: Grant Require Passwordless MFA for Admins when Browser and Modern Auth Clients-v1.1				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
Global Administrator User Administrator SharePoint Administrator Security Administrator Password Administrator Helpdesk Administrator Compliance Administrator Exchange Administrator Conditional Access Administrator Billing Administrator Authentication Administrator	All	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except				
AAD_UA_ConAcc-Breakglass			Require authentication strength (Preview) Admin MFA	
AAD_UA_CAU008_Exclude				

6.1.4.11 CAU009-AzureManagement: Grant Require MFA for Azure Management for All Users when Browser and Modern Auth Clients-v1.0

This policy makes sure that MFA is required when the Azure Management portal is requested. The reason for this policy is that when using PIM, your admin users might first go to the Admin portal, to

request their rights using PIM afterwards. See: [Lessons learned while implementing Azure AD Privileged Identity Management (PIM)] for more information.

CAU009-AzureManagement: Grant Require MFA for Azure Management for All Users when Browser and Modern Auth Clients-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All	Microsoft Azure Management	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except			Require multi-factor authentication	
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAU009_Exclude				

6.1.4.12 CAU009-AzureManagement: Grant Require MFA for Azure Management for All Users when Browser and Modern Auth Clients-v1.1 (Optional)

This policy is required when you want to start using Authentication Strength, requiring a step-up authentication is necessary

CAU009-AzureManagement: Grant Require MFA for Azure Management for All Users when Browser and Modern Auth Clients-v1.1				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All	Microsoft Azure Management	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except			Require authentication strength (Preview) Admin MFA	
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAU009_Exclude				

6.1.4.13 CAU010-All: Grant Require ToU for All Users when Browser and Modern Auth Clients-v1.1 (Optional)

This one is optional, but the policy requires users to agree to the Terms of Use (TOU) first, before they are allowed to access the resources.

CAU010-All: Grant Require ToU for All Users when Browser and Modern Auth Clients- v1.1				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	All	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except	exclude:			
AAD_UA_ConAcc-Breakglass	Microsoft Intune		Terms of Use	
AAD_UA_CAU010_Exclude	Microsoft Intune Enrollment			

6.1.4.14 CAU011-All: Block access for All users except licensed when Browser and Modern Auth Clients-v1.0 (Optional)

This one is optional as well, but I personally recommend it even though it is a risky one. It will block access to any user which is not licensed. Make sure that you also exclude your admins from this policy. If you implement this policy you can really govern who can access the environment but requires careful planning.

CAU011-All: Block access for All users except licensed when Browser and Modern Auth Clients-v1.0				
Assignments			Access Controls	
Users	Cloud Apps	Conditions	Grant	Session
All Users	All	Client Apps: Browser Mobile Apps and Desktop Clients	Block	
Except				
AAD_AA_ConAcc-Breakglass				
AAD_AA_CAU011-Exclude				
License groups				

6.1.4.15 CAU012-RSI: Combined Security Info Registration with TAP-v1.0

Policy which you should start using if you want to allow Temporary Access Pass for new users outside of your company trusted locations.

CAU012-RSI: Combined Security Info Registration with TAP-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	Register Security Information	Location: Any location and all trusted locations excluded	Grant	
Except				
AAD_UA_ConAcc-Breakglass			Require authentication strength (Preview)	
AAD_UA_CAU012_Exclude			Multi Factor Authentication	

6.1.5 Device

The device policies relate to the device the user is coming from. Keep in mind here that if a device which is managed but for some reason is not compliant, other policies apply targeted to non-compliant devices. In this case for example, the user will not receive any email anymore within the Outlook desktop client but can login to the Office portal with browser based restrictions.

6.1.5.1 CAD001-O365: Grant macOS access for All users when Browser and Modern Auth Clients and Compliant-v1.1

Only grant access if the macOS device is compliant. In the policy below we exclude Guests and External users, allowing them to use Client Applications to access a Teams environment hosted in your tenant. If you only want to allow browser access (either full access or restricted) you should not exclude Guest and External users.

CAD001-O365: Grant macOS access for All users when Modern Auth Clients and Compliant-v1.1				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	Office 365	Client Apps: Mobile Apps and Desktop Clients	Grant	
Except		Device Platform: macOS		
Guest or External Users			Require device to be marked as compliant	
And				
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAD001_Exclude				

6.1.5.2 CAD002-O365: Grant Windows access for All users when Modern Auth Clients and Compliant-v1.1

Only grant access if the Windows device is compliant. In the policy below we exclude Guests and External users, allowing them to use Client Applications to access a Teams environment hosted in your tenant. If you only want to allow browser access (either full access or restricted) you should exclude Guest and External users.

CAD002-O365: Grant Windows access for All users when Browser and Modern Auth Clients and Compliant-v1.0				
Assignments			Access Controls	
Users	Cloud Apps	Conditions	Grant	Session
All Users	Office 365	Client Apps: Browser Mobile Apps and Desktop Clients	Grant	
Except		Device Platform: Windows		
Guest and External Users			Require device to be marked as compliant	
And				
AAD_AA_ConAcc-Breakglass				
AAD_AA_CAD002-Exclude				

6.1.5.3 CAD003-O365: Grant iOS and Android access for All users when Modern Auth Clients and ApprovedApp and Compliant-v1.1

Only grant access if the iOS or Android device is compliant or if an Approved Client App is used. In the policy below we exclude Guests and External users, allowing them to use Client Applications to

access a Teams environment hosted in your tenant. If you only want to allow browser access (either full access or restricted) you should exclude Guest and External users.

Microsoft is transitioning towards apps having a protection policy applied to eventually replace the Approved Apps functionality in some of the scenarios. For now, using this option isn't advised yet since the Microsoft Teams app is not yet supported. See my article: [Mobile Application Management for Mobile Devices with Microsoft Endpoint Manager/Intune deep dive]

Did you know that even if your apps are managed by another company, you can switch profiles with Microsoft Edge. So, in this case, even though the apps are managed by company A, you can switch the logged in session in the Microsoft Edge browser to company B requiring an Approved App as well. If you just want to be able to use any browser, do not include the Browser option, and the user will then receive "CAD006-O365: Session block download on unmanaged device when All users when Browser" which uses App enforced restrictions. (no download and no printing) just as on non-compliant Windows and macOS devices. If you require that web access on mobile devices should only be possible from a managed browser (Microsoft Edge) you must include the Browser in the Client App selection.

CAD003-O365: Grant iOS and Android access for All users when Modern Auth Clients and ApprovedApp or Compliant-v1.1				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	Office 365	Client Apps: Mobile Apps and Desktop Clients	Grant	
Except		Device Platform: iOS and Android		
Guest and External Users			Require device to be marked as compliant	
And			or	
AAD_UA_ConAcc-Breakglass			Require Approved Client App	
AAD_UA_CAD003_Exclude				

6.1.5.4 CAD004-O365: Grant Require MFA for All users when Browser and Non-Compliant-v1.2

Require MFA if the device falls out of compliance.

CAD004-O365: Grant Require MFA for All users when Browser and Non-Compliant-v1.2				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	Office 365	Client Apps: Browser	Grant	
		Filter for devices: Exclude isCompliant Equals True Or trustType Equals Hybrid Azure AD Joined		
Except				
AAD_UA_ConAcc-Breakglass			Require multi-factor authentication	
AAD_UA_CAD004_Exclude				

6.1.5.5 CAD004-O365: Grant Require MFA for All users when Browser and Non-Compliant-v1.3

CAD004-O365: Grant Require MFA for All users when Browser and Non-Compliant-v1.3				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	Office 365	Client Apps: Browser	Grant	
		Filter for devices: Exclude isCompliant Equals True Or trustType Equals Hybrid Azure AD Joined		
Except				
AAD_UA_ConAcc-Breakglass			Require Authentication Strength	
AAD_UA_CAD004_Exclude			Multi Factor Authentication	

6.1.5.6 CAD005-O365: Block access for unsupported device platforms for All users when Modern Auth Clients-v1.1

Block unsupported device platforms, like Windows Phone from accessing the environment.

CAD005-O365: Block access for unsupported device platforms for All users when Modern Auth Clients-v1.1					
Assignments				Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions		Grant	Session
All Users	Office 365	Client Apps: Browser (Optional) Mobile Apps and Desktop Clients		Block	
Except		Device Platform: Any, except: Android iOS Windows macOS Linux			
AAD_UA_ConAcc-Breakglass					
AAD_UA_CAD005_Exclude					

Before you enable this policy, make sure that you have no "unknown" clients accessing the environment. You should check Azure AD sign-in logging as described in the article: [Microsoft is going to disable basic/legacy authentication for Exchange Online. What does that actually mean and does that impact me?]

6.1.5.7 CAD006-O365: Session block download on unmanaged device for All users when Browser and Modern App Clients and Non-Compliant-v1.6

This policy uses the App Enforced Restrictions, blocking download of files in OneDrive/SharePoint and Outlook Web Access. See: [Limit Access to Outlook Web Access, SharePoint Online and OneDrive using Conditional Access App Enforced Restrictions].

CAD006-O365: Session block download on unmanaged device for All users when Browser and Modern App Clients and Non-Compliant-v1.6					
Assignments				Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions		Grant	Session
All Users	Office 365	Client Apps: Browser, Mobile Apps and Desktop Clients			Use App Enforced Restrictions
Except		Filter for devices: Exclude isCompliant Equals True Or trustType Equals Hybrid Azure AD Joined			
AAD_UA_ConAcc-Breakglass					
AAD_UA_CAD006_Exclude					

6.1.5.8 CAD007-O365: Session set Sign-in Frequency for Apps for All users when Modern Auth Clients and Non-Compliant-v1.3

With this policy, you force users using Modern Authentication Clients to reauthenticate after a specified number of hours/days. I normally set this to once per 7 days.

CAD007-O365: Session set Sign-in Frequency for Apps for All users when Modern Auth Clients and Non-Compliant-v1.3					
Assignments				Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions		Grant	Session
All Users	Office 365	Client Apps: Mobile Apps and Desktop Clients			Sign-in Frequency: 7 Days
Except		Filter for devices: Exclude isCompliant Equals True Or trustType Equals Hybrid Azure AD Joined			
AAD_UA_ConAcc-Breakglass		Device Platform: iOS Android			
AAD_UA_CAD007_Exclude					

6.1.5.9 CAD008-All: Session set Sign-in Frequency for All users when Browser and Non-Compliant-v1.1

With this policy, you force users using the Browser to reauthenticate after a specified number of hours/days. I normally set this to once per day.

CAD008-All: Session set Sign-in Frequency for All users when Browser and Non-Compliant-v1.1				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	All	Client Apps: Browser		Sign-in Frequency: 1 Days
Except		Filter for devices: Exclude isCompliant Equals True		
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAD008_Exclude				

6.1.5.10 CAD009-All: Session disable browser persistence for All users when Browser and Non-Compliant-v1.1

This policy makes sure that the session is not persisted in the Browser when the browser is closed. See: [Understanding and governing reauthentication settings in Azure Active Directory] for more information.

CAD009-All: Session disable browser persistence for All users when Browser and Non-Compliant-v1.3				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	All	Client Apps: Browser		Persistent Browser Session
Except		Filter for devices: Exclude isCompliant Equals True Or trustType Equals Hybrid Azure AD Joined		Never persistent
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAD009_Exclude				

6.1.5.11 CAD010-All: Require MFA for device join or registration when Browser and Modern Auth Clients-v1.0

This policy specifies that during Azure AD join or register operations, MFA is required

CAD010-All: Require MFA for device join or registration when Browser and Modern Auth Clients-v 1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	User actions: Register or join devices		Grant	
Except				
AAD_AA_ConAcc-Breakglass			Require multi-factor authentication	
AAD_AA_CAD010-Exclude				

6.1.5.12 CAD011-O365: Grant Linux access for All users when Modern Auth Clients and Compliant-v1.0

Only grant access if the Linux device is compliant We will most probably not have any Modern Auth Client on the Linux device itself, since Microsoft sunsetted the Teams client for Linux in favour of using a Progressive Web App, which is web based (and therefore out of scope of this policy).

CAD011-O365: Grant Linux access for All users when Modern Auth Clients and Compliant-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	Office 365	Client Apps: Mobile Apps and Desktop Clients	Gramt	
Except		Device Platform: Linux		
AAD_UA_ConAcc-Breakglass			Require device to be marked as compliant	
AAD_UA_CAD011_Exclude				

6.1.5.13 CAD012-ALL: Grant access for Admin users when Browser and Modern Auth Clients and Compliant-v1.0 (Optional)

This optional policy, will only allow admins, to access the environment while on a Compliant device, this will require the Admin to sign-in to the browser so that the Compliance status is available for evaluation, this will also trigger a register of the admin user on the device.

CAD012-ALL: Grant access for Admin users when Browser and Modern Auth Clients and Compliant-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
Global Administrator User Administrator SharePoint Administrator Security Administrator Password Administrator Helpdesk Administrator Compliance Administrator Exchange Administrator Conditional Access Administrator Billing Administrator Authentication Administrator	All Cloud Apps	Client Apps: Browser and Mobile Apps and Desktop Clients	Grant Require device to be marked as compliant or Require Hybrid Azure AD joined device	
Except				
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAD012_Exclude				

6.1.5.14 CAD013-Selected: Grant access for All users when Browser and Modern Auth Clients and Compliant-v1.0 (Optional)

This policy is optional as well, if you have applications which you want only to be available on devices which you manage (either Compliant or Hybrid AD joined) then you should add those applications to this policy so that it can be enforced.

CAD013-Selected: Grant access for All users when Browser and Modern Auth Clients and Compliant-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Grant	Session
All Users	Selected Apps	Client Apps: Browser and Mobile Apps and Desktop Clients	Grant Require device to be marked as compliant or Require Hybrid Azure AD joined device	
Except				
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAD013_Exclude				

6.1.6 Location

Location based Conditional Access policies relate to the location the user is coming from. I'm personally not a fan of excluding company locations from MFA policies. I had some cases for example where company cases were excluded but came to the conclusion that the Guest WiFi network allowing all customers used the same internet IP address, making this network trusted as well.

6.1.6.1 CAL001-All: Block untrusted locations for All users when Browser and Modern Auth Clients-v1.1 (Optional)

If your company doesn't do business in certain countries, you might as well block access from these countries. Even though there are ways to circumvent this (like using a VPN for example), it might make your security a little bit better than your neighbor.

CAL001-All: Block untrusted locations for All users when Browser and Modern Auth Clients-v1.1				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Block	Session
All Users	All	Client Apps: Browser Mobile Apps and Desktop Clients		
Except		Locations: Untrusted Locations		
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAL001_Exclude				

6.1.6.2 CAL002:RSI: Require MFA registration from trusted locations only for All users when Browser and Modern Auth Clients-v1.2 (Optional)

MFA registration, is something that you might want to allow only when the user is in a trusted location. For example, when onboarding the user.

CAL002:All: Require MFA registration from trusted locations only for All users when Browser and Modern Auth Clients-v1.0				
Assignments			Access Controls	
Users	Cloud Apps	Conditions	Grant	Session
All Users	All	Client Apps: Browser Mobile Apps and Desktop Clients	Block	
Except	Actions: Register Security Information	Locations: All Locations, except Trusted Locations		
AAD_AA_ConAcc-Breakglass				
AAD_AA_CAL002-Exclude				

6.1.6.3 CAL003:All: Block Access for Specified Service Accounts except from Provided Trusted Locations when Browser and Modern Auth Clients-v1.0

This policy will only allow selected accounts (usually so called Service Account a.k.a. non-personal accounts) to login while coming from the specified locations. By doing so we could exclude those accounts from the policies requiring MFA. We can accomplish this by adding those accounts to the AAD_UA_CAL003_Include group, and nesting that group in the following exclude groups:

- AAD_UA_CAD005_Exclude
- AAD_UA_CAU002_Exclude
- AAD_UA_CAU008_Exclude
- AAD_UA_CAU009_Exclude

CAL003:All: Block Access for Specified Service Accounts except from Provided Trusted Locations when Browser and Modern Auth Clients-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Block	Session
AAD_UA_CAL003_Include	All Cloud Apps	Client Apps: Browser Mobile Apps and Desktop Clients		
Except		Locations: All Locations, except Selected Locations		
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAL003_Exclude				

6.1.6.4 CAL004-All: Block access for Admins from non-trusted locations when Browser and Modern Auth Clients-v1.0 (Optional)

This policy, will restrict the admin access to the environment from the specified locations only, this can be used in combination with CAD012, or instead of that policy.

CAL004-All: Block access for Admins from non-trusted locations when Browser and Modern Auth Clients-v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	Block	Session
Global Administrator User Administrator SharePoint Administrator Security Administrator Password Administrator Helpdesk Administrator Compliance Administrator Exchange Administrator Conditional Access Administrator Billing Administrator Authentication Administrator	All Cloud Apps	Client Apps: Browser Mobile Apps and Desktop Clients		
Except		Locations: All Locations, except trusted Locations		
AAD_UA_ConAcc-Breakglass				
AAD_UA_CAL004_Exclude				

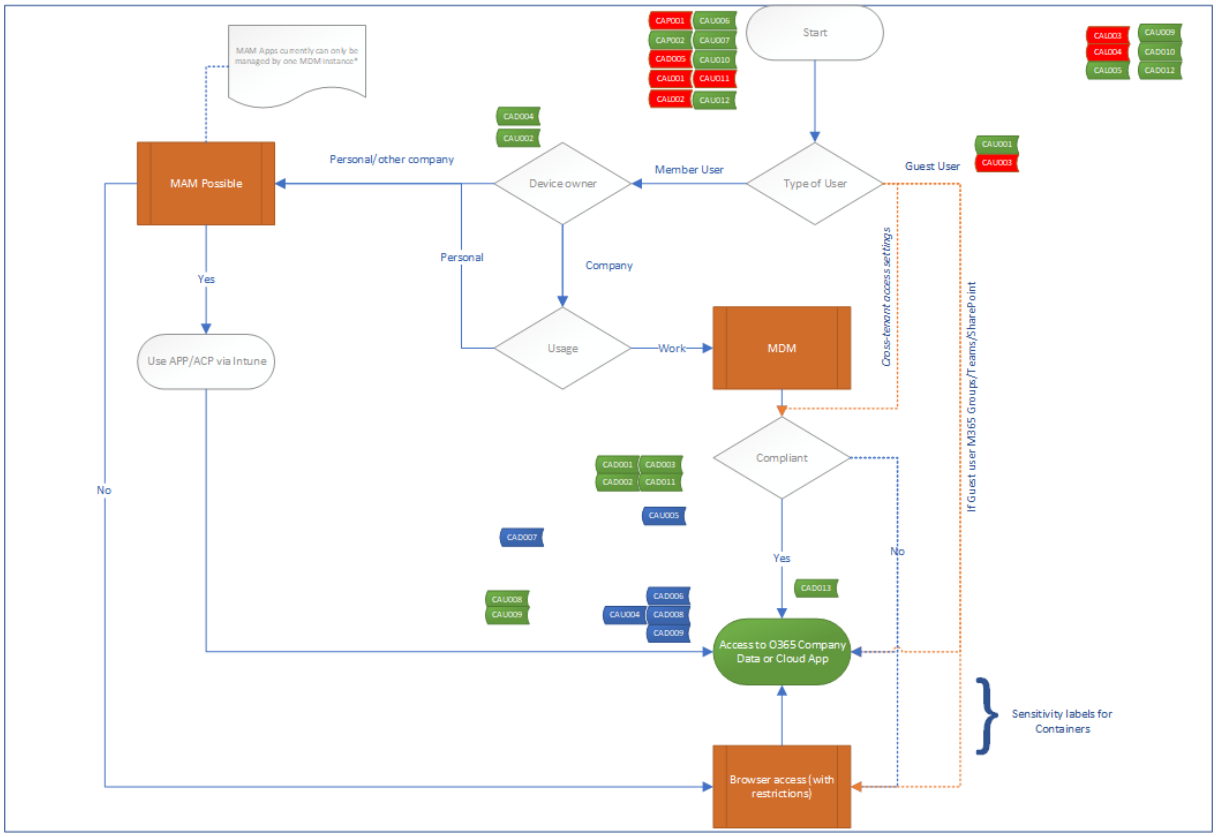
6.1.6.5 CAL005-Selected: Grant access for All users on less-trusted locations when Browser and Modern Auth Clients and Compliant - v1.0 (Optional)

This policy can be used to only allow access from Compliant or Hybrid AD joined device in less trusted locations (for example certain defined countries). Since Office 365 is excluded, users can still access Office 365 by using the browser.

CAL005-Selected: Grant access for All users on less-trusted locations when Browser and Modern Auth Clients and Compliant - v1.0				
Assignments			Access Controls	
Users	*Cloud Apps, User Actions or Authentication Context*	Conditions	*Grant*	Session
All Users	All Cloud Apps	Client Apps: Browser Mobile Apps and Desktop Clients	*Require device to be marked as compliant* *or* *Require Hybrid AD Joined device* *Or* *Require approved client app*	
Except	Exclude	Selected Locations: Less-trusted locations		
AAD_UA_ConAcc-Breakglass	Office 365			
AAD_UA_CAL005_Exclude				

6.1.7 Translating the functional design to the technical implementation

Below is an overview of the functional flowchart, with tags for the Conditional Access policies. Even though the position of the CA policy is not always fully correct or can be applicable in more than one scenario, it gives an idea on where the policy is applied and might help while troubleshooting.



6.2 Limit Access to Outlook Web Access and SharePoint Online and OneDrive using Conditional Access App Enforced Restrictions

One of the scenario's we can build with Conditional Access, is the scenario where we restrict access inside the web application itself. By doing so, you could for example limit the functionality of the web applications on non-managed devices, or when accessing the web application from a country where your company normally does not operate. The web applications can be configured to behave differently if the user is applicable for a Conditional Access policy where App Enforced restrictions are configured.

Within the Office 365 suite of applications, the following web applications are supported for App Enforced Restrictions:

- Outlook Web Access
- SharePoint and OneDrive

6.2.1 Configure Outlook Web Access for limited access via App Enforced Restrictions

Before you can enable Conditional Access App Enforced Restrictions you first need to enable the feature in the default OWA mailbox policy, since by default this functionality is turned off, this can be done using the Set-OwaMailBoxPolicy cmdlet as part of the Exchange Online PowerShell module.

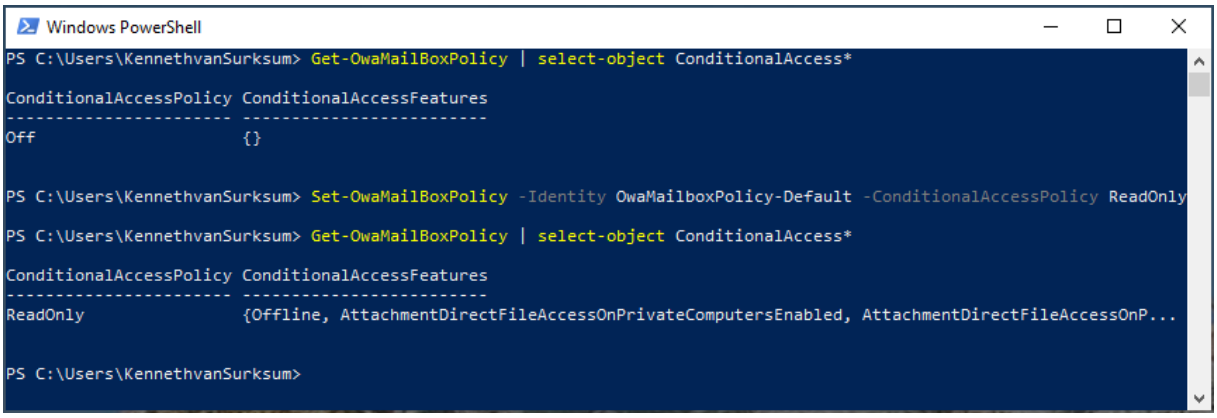
First request the current status of the OWA mailbox policy by executing the following command: Get-OwaMailBoxPolicy |select-object ConditionalAccess*. This command will return the current status of the ConditionalAccessPolicy (On or Off) and the ConditionalAccessFeatures.

If the ConditionalAccessPolicy is set to Off, you can enable the functionality which allows for restrictions when used in combination with a Conditional Access App Enforced Restriction policy. You have either the option to configure the policy in two modes:

- ReadOnly, where users can't download attachments to their local computer, and can't enable Offline Mode on non-compliant computers
- ReadOnlyPlusAttachmentsBlocked, in the ReadOnly setting viewing attachments in the browser is possible, when using this setting viewing attachments in the browser is blocked.

In this example we are going to enable the ConditionalAccessPolicy in the OWA Mailbox Policy and use the ReadOnly mode, this can be accomplished by executing the following command: Set-OwaMailbBoxPolicy -Identity OwaMailboxPolicy-Default -ConditionalAccessPolicy ReadOnly

After executing the command make sure that you check whether the setting was succesfull by executing the Get-OwaMailBoxPolicy |select-object ConditionalAccess* command again and check whether the ConditionalAccessPolicy is set to ReadOnly



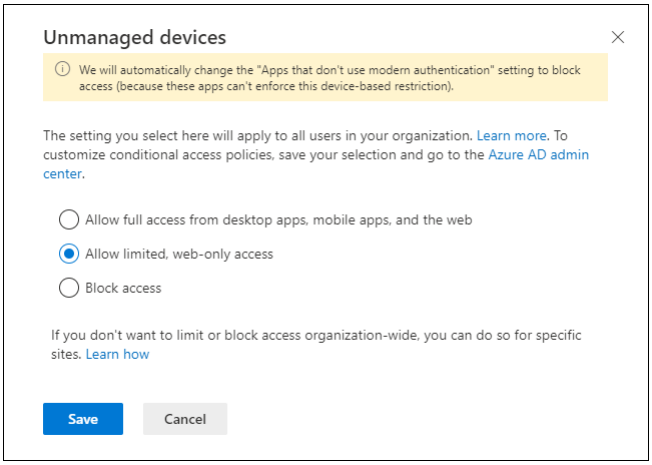
6.2.2 Configure SharePoint Online and OneDrive for limited access via App Enforced Restrictions

SharePoint Online and OneDrive can be configured in several ways. The first option is to set a Global setting which becomes effective for all SharePoint Online and OneDrive sites in your environment, and the Per site option allows you to specify options per site.

6.2.2.1 Global settings

The Global or organizational-wide settings can be configured from the SharePoint admin center (https://<tenantname>-admin.sharepoint.com). In the SharePoint Admin center select Policies | Access Control and select Unmanaged devices.

By default, for unmanaged devices the option "Allow full access from desktop apps, mobile apps, and the web" is selected, and by modifying the option to either "Allow limited, web-only access" or "Block access" you configure limited access for your whole environment.



If you configure the unmanaged devices settings, 2 new Conditional Access policies will be created. While I'm not a big fan of letting a setting like create the Conditional Access policies for you, they do provide some valuable information. I would therefore suggest to let the wizard create them, and that you turn them off immediately after creation. Keep in mind that if you are playing with these, it might be that several Conditional Access policies are created (one per day, since the day when created in defined in the name)

The first one "[SharePoint admin center]Block access from apps on unmanaged devices - 2020-06-25" has the following properties

Name: [SharePoint admin center]Block access from apps on unmanaged devices - 2020-06-25
Assignments
    Users and Groups: All Users
    Cloud apps or actions: Office 365 SharePoint Online
    Conditions
        Client apps: Modern Authentication clients
Access controls
    Grant: Require device to be marked as compliant OR Require Hybrid Azure AD joined device

So, to summarize this policy grants access to either Compliant Azure AD joined devices or Hybrid joined (AD joined, Azure AD registered) devices when the client supports Modern Authentication while accessing SharePoint Online.

So having clients which support Modern Authentication is crucial for this.

The second one "[SharePoint admin center] Use app-enforced Restrictions for browser access - 2020-06-25" has the following properties

Name: [SharePoint admin center]Use app-enforced Restrictions for browser access - 2020-06-25
Assignments
    Users and Groups: All Users
    Cloud apps or actions: Office 365 SharePoint Online
    Conditions
        Client apps: Browser
Access controls
    Session: Use app enforced restrictions

This conditional access policy when applicable gives SharePoint online, the signal that the limited access is applicable.

6.2.2.2 Per site settings

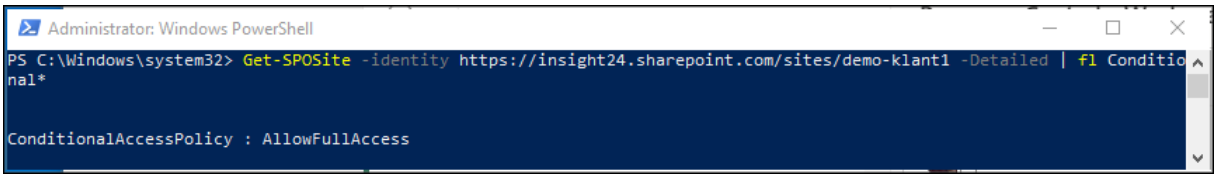
For a basic environment, having these global settings might be enough, but perhaps you want to more granularly control whether you want the limited access applied to a SharePoint or OneDrive site. This can be accomplished by using PowerShell (for now, more on that later) and described in the following section "Block or limit access to a specific SharePoint site or OneDrive" of the article: "Control access from unmanaged devices" in the SharePoint online documentation.

In order to use PowerShell you must have the SharePoint Online Management Shell installed, once installed, you can connect using

Connect-SPOService -Url https://<tenantname>-admin.sharepoint.com which will log you in, into the Management Shell

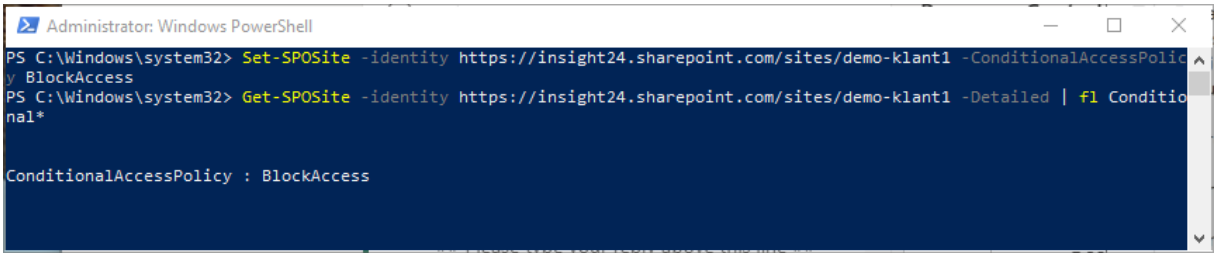
From there you can determine the current status of a particular SharePoint site using the following command.

Get-SPOSite -Identity https://<SharePoint online URL>/sites/<name of site or OneDrive account> -Detailed | fl Conditional*



As you can see, this command has a strange outcome, since on a Global level we just defined Read Only Access. Turns out that if you specify the setting on site level, this will override the global policy. So for example, you could set the option "Allow limited, web-only access" as described above in the Global policy, but define a setting to Block Access on the Site level.

You can set the policy on a site level by executing the following command: Set-SPOSite -Identity https://<SharePoint online URL>/sites/<name of site or OneDrive account> -ConditionalAccessPolicy <value>



For the ConditionalAccessPolicy parameter the following values are available:

- AllowFullAccess: The default setting
- AllowLimitedAccess: The setting allowing limited access
- BlockAccess: Blocks access

When using the AllowLimitedAccess option, you can supply additional parameters to further define the behavior, as detailed in "Advanced Configurations", for example you can provide the option -LimitedAccessFileType OtherFiles after the -ConditionalAccess AllowLimitedAccess parameter to allow users to download files that can't be previewed, such as .ZIP files.

If you must specify this for a lot of SharePoint sites, you can of course automate these settings, but wouldn't it be nice if we could enable this from the GUI while creating the SharePoint site, or while the SharePoint site is created as part of a Teams environment.

6.2.2.3 Using sensitivity labels for containers

What if you want more granularity and want to decide on a per SharePoint site basis whether or not these App Enforced Restrictions should be applicable?

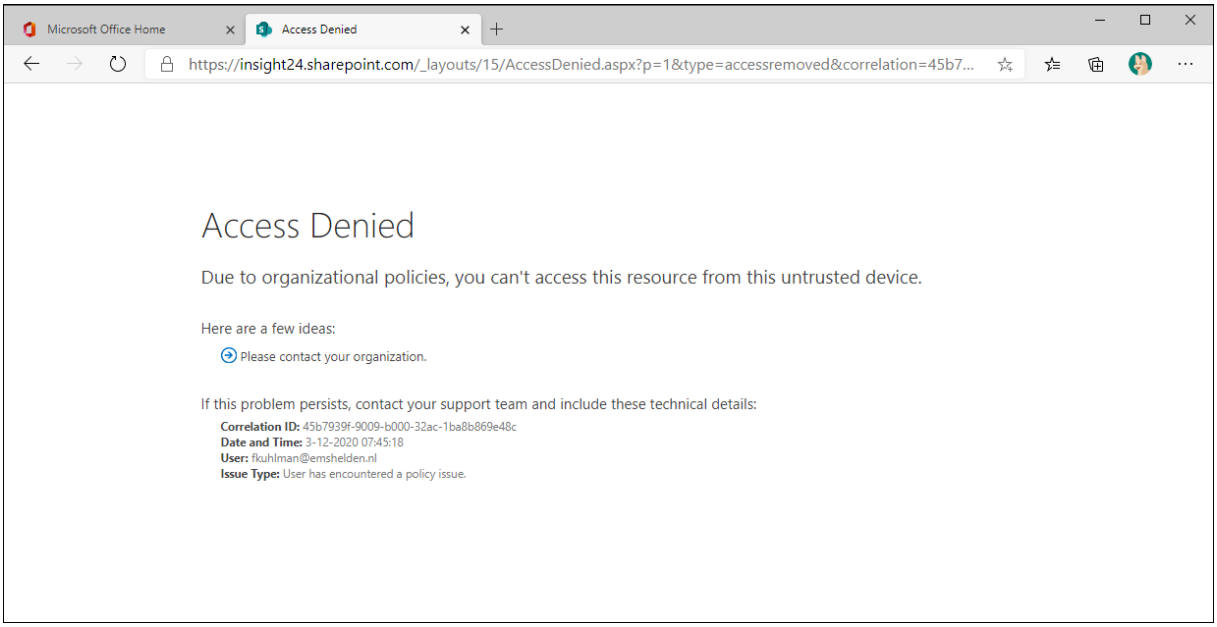
Luckily, there are options.

6.2.2.3.1 Option 1: Block or limit access to a specific SharePoint site or OneDrive

This option is explained in the following article: Control access from unmanaged devices, the article explains that by using PowerShell you can limit access by using the Set-SPOSite commandlet.

Set-SPOSite -Identity https://<SharePoint online URL>/sites/<name of site or OneDrive account> -ConditionalAccessPolicy AllowLimitedAccess

When using this option, you must remove the global setting, since setting another setting on a subsite only works when its less restrictive. If you would for example set the global policy to "Allow limited, web-only access" and use the Set-SPOSite commandlet to set the Conditional Access Policy for a specific site to "Allow full access from desktop apps, mobile apps, and the web" using the AllowFullAccess parameter, the access will still be limited. If however you would set the Conditional Access Policy for a specific site to "Block Access" using the BlockAccess parameter, the access to the site will be blocked.



Using this method, has major disadvantages, since you have to execute the necessary PowerShell command for each SharePoint and OneDrive after its created. This can easily be forgotten and can lead to inconsistency.

6.2.2.3.2 Option 2: Use Sensitivity labels for Containers

There is far more to tell when it comes to Sensitivity labels then explained in this blogpost. For this blogpost we are going to make use of Sensitivity labels for contains, which can be used to define certain settings when creating a Teams or SharePoint environment.

By using sensitivity labels for containers we can control the following settings:

- Privacy and external user access settings
    - Use the label to determine whether the site privacy is set to Public, Private or None
    - Define whether Microsoft 365 Group owners can add Guest users to the group.
- Device access and external sharing settings
    - You can determine if external sharing settings already on the site will be replaced or respected.
    - You can determine the access from unmanaged devices (same options as on global level - Full, Limited and Block)

So, by defining Sensitivity labels for contains we can actually determine the access from unmanaged devices setting that will be used when the Conditional Access policy which enforces the App Enforced Restrictions will be hit.

The settings of this policy is explained in my article: "Conditional Access demystified: My recommended default set of policies", the name of the policy providing this functionality is called: "CAD006-O365: Session block download on unmanaged device when All users when Browser-v1.0"

CAD006-O365: Session block download on unmanaged device when All users when Browser-v1.0				
Assignments			Access Controls	
Users	Cloud Apps	Conditions	Grant	Session
All Users	Office 365	Client Apps: Browser		Use App Enforced Restrictions
Except		Device state: All except Device marked as Compliant		
AAD_AA_ConAcc-Breakglass				
AAD_AA_CAD006-Exclude				

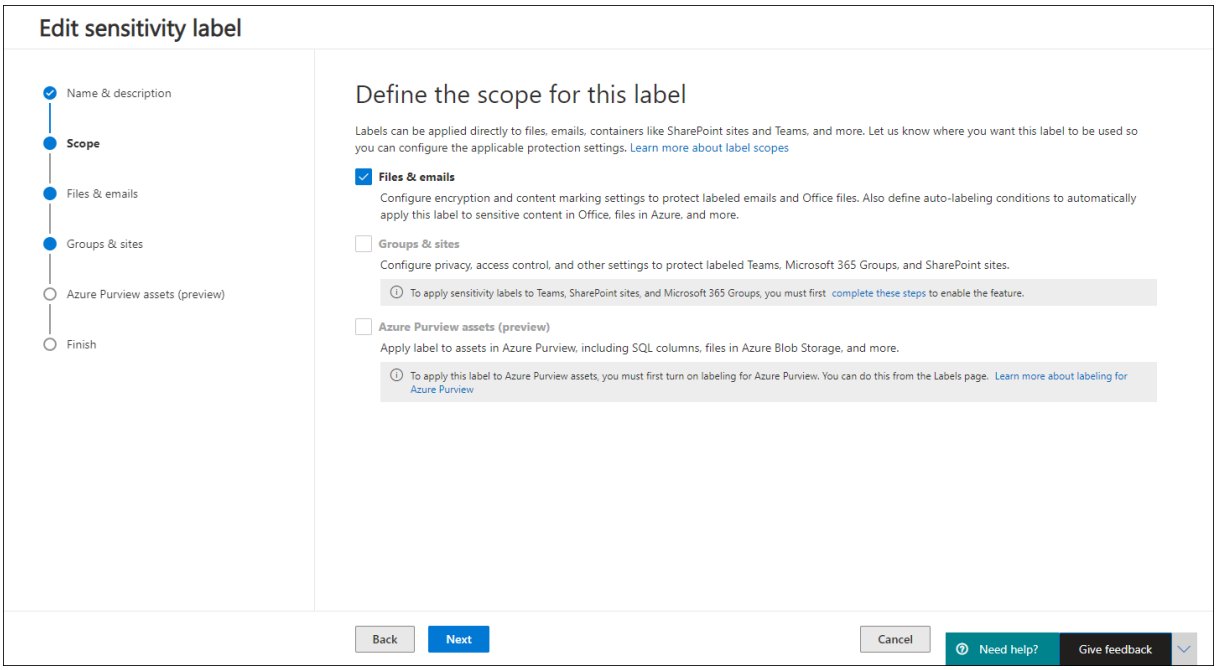
After creating the sensitivity labels, you can use them for each new Teams/SharePoint site created, and based on the defined Sensitivity label the correct access from unmanaged devices setting will be applied.

This is not a perfect solution, since it will not solve the issue for all SharePoint sites already created, and you still must build a solution for OneDrive sites which are created automatically and do not have the option to define a "default" sensitivity label at creation.

Let's go a little bit more into detail in how to build this from scratch, let's walk through the steps.

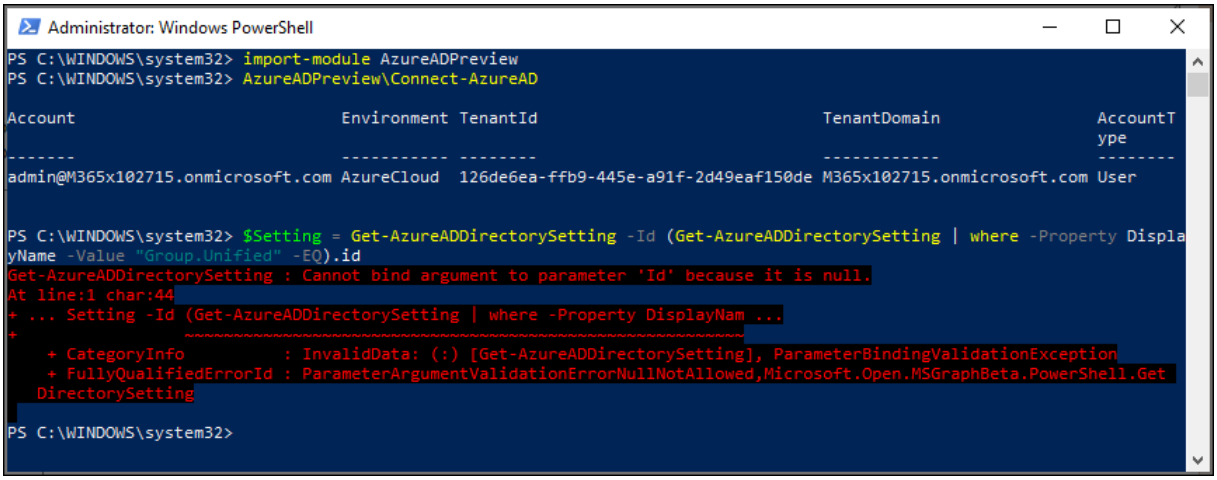
Step 1: Enable sensitivity labels for containers.

By default, support for sensitivity labels for containers is not enabled, you can easily determine this by creating a new sensitivity label or by trying to modify an existing one. If the option to select Group & sites is greyed out, you first must execute some steps to enable this functionality.

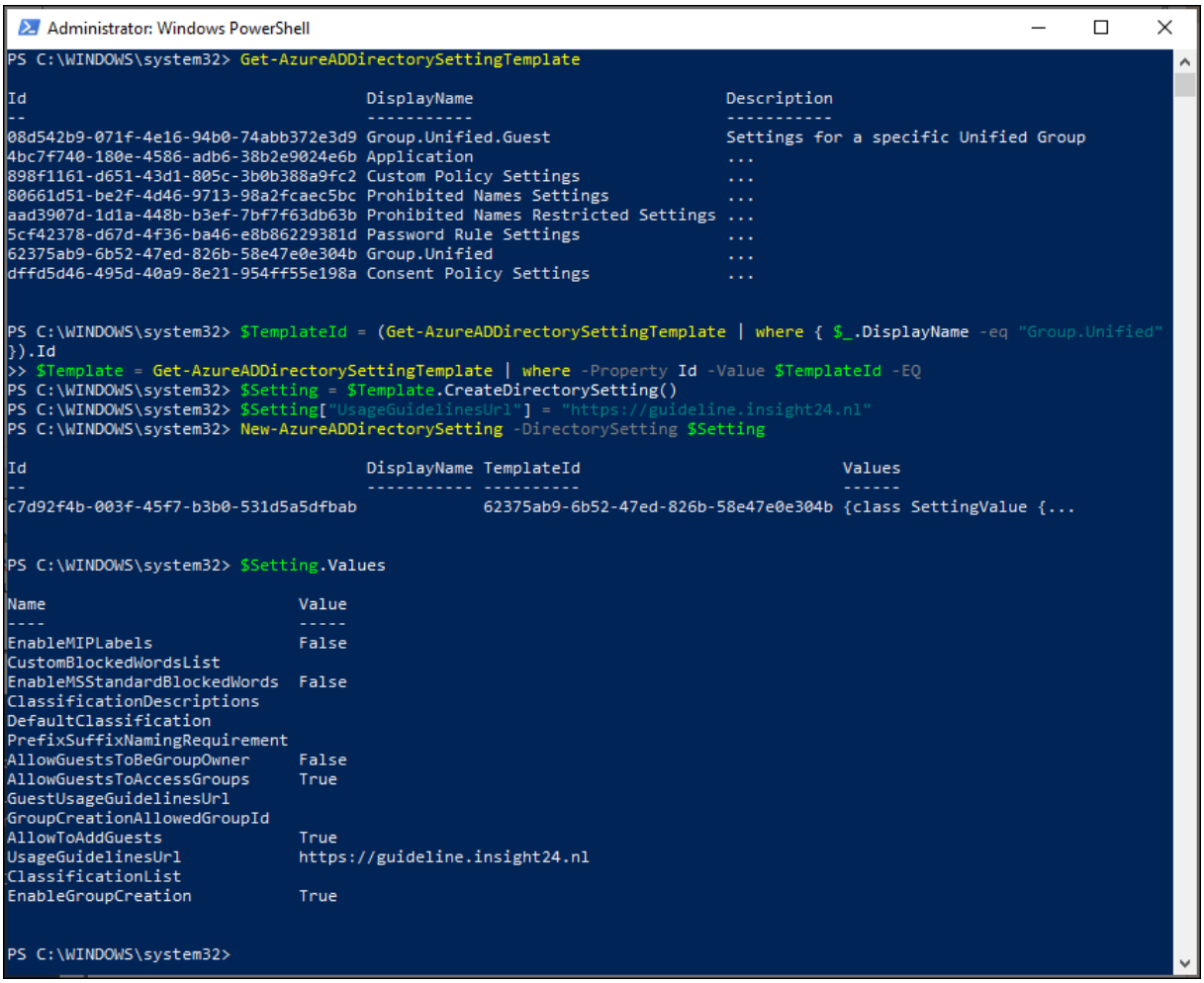


The following procedure is explained in the following article: [How to enable sensitivity labels for containers and synchronize labels] and one of the first steps is to enable the feature to apply sensitivity labels to groups as explained in this article: [Assign sensitivity labels to Microsoft 365 groups in Azure Active Directory]

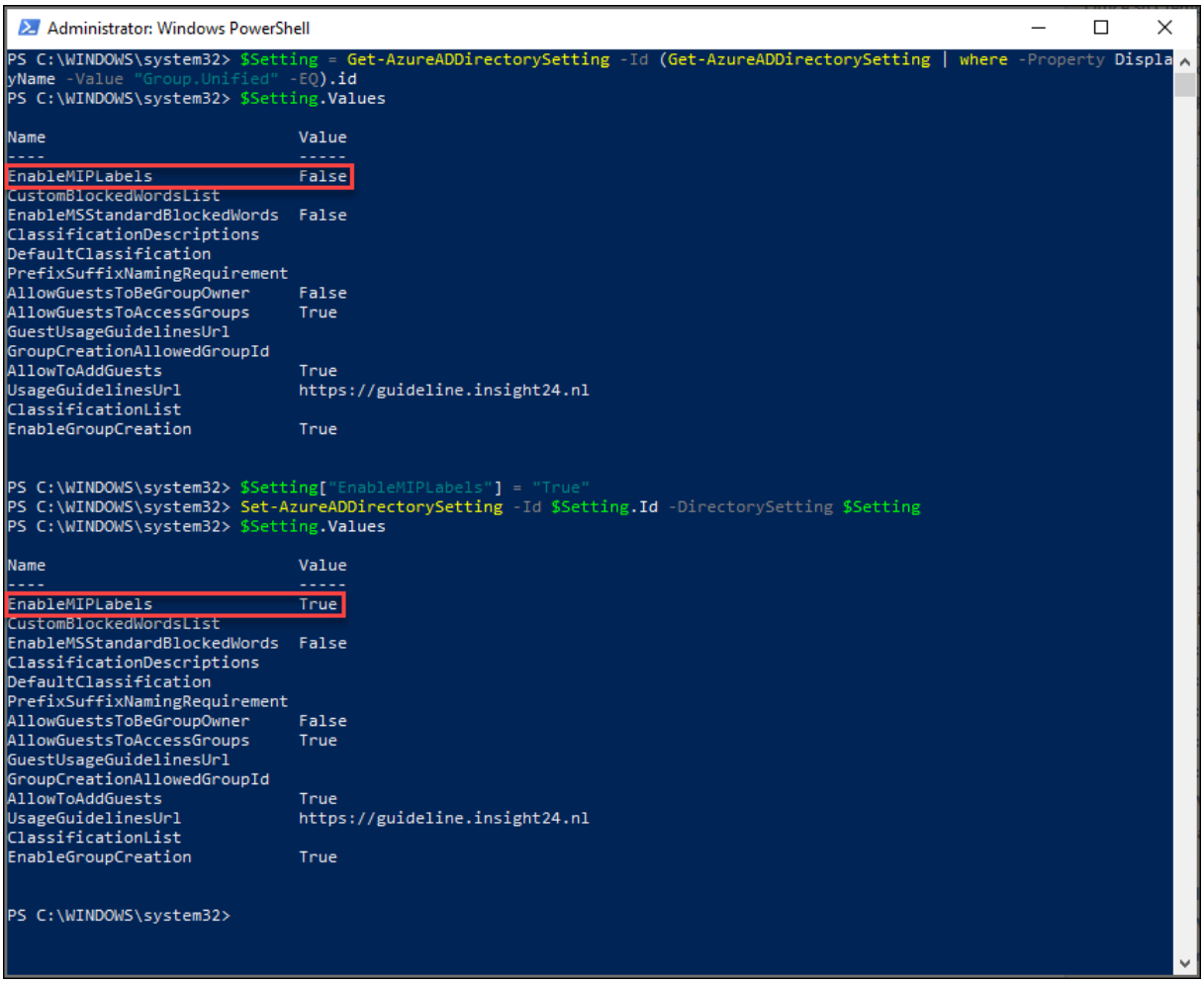
The first thing we need to do, is to import the AzureADPreview module using the import-module AzureADPreview and connect using the Connect-AzureAD commandlet. Make sure that you use the Connect-AzureAD commandlet from the Azure AD Preview module by putting AzureADPreview\ in front of it. Once connected verify if group settings have been set for the Azure AD organization. If no group settings are applied, you'll get the same error as in the picture below.



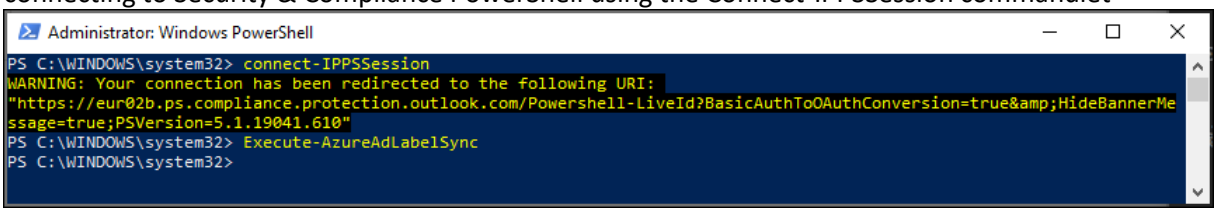
If this is the case you first have to create the settings, as explained in the following article: [Azure Active Directory cmdlets for configuring group settings]



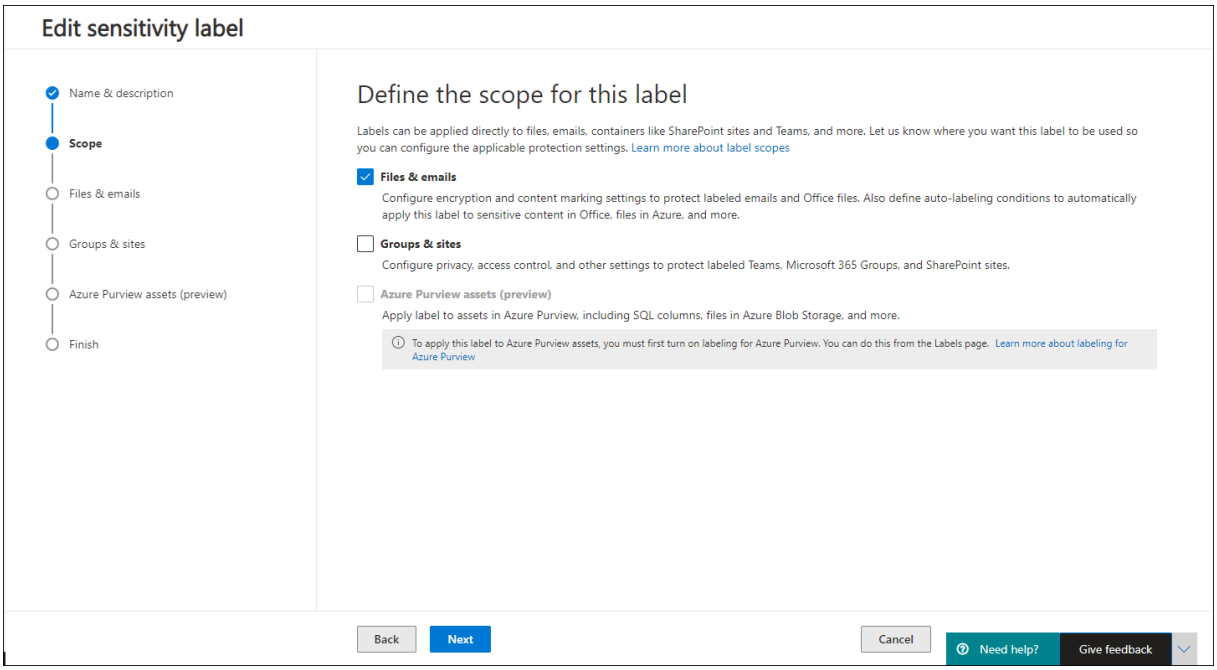
After performing these steps, you can continue with enabling the Microsoft Information Protection labels functionality as shown in the figure below. You can see that the EnableMIPLabels value is set to True.



After this is done, you must synchronize your sensitivity labels to Azure AD. You can do this by connecting to Security & Compliance PowerShell using the Connect-IPPSSession commandlet



Once finished, the result should be that you are able to specify the "Groups & Sites" option when modifying an existing Sensitivity label or creating a new one.

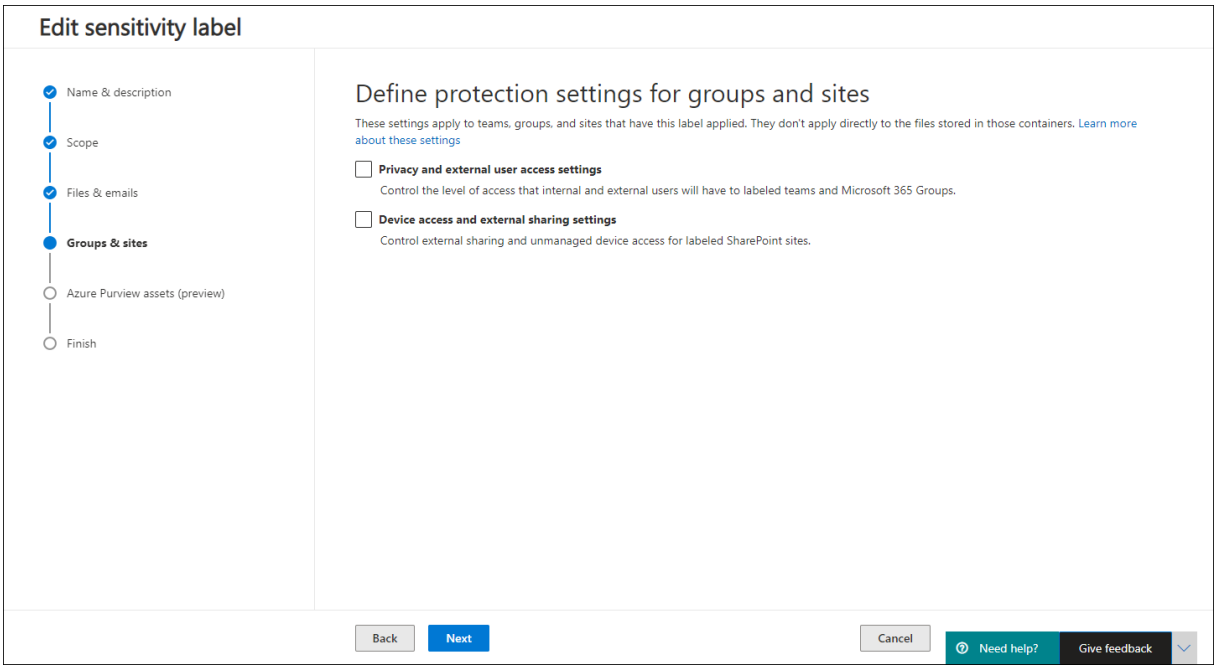


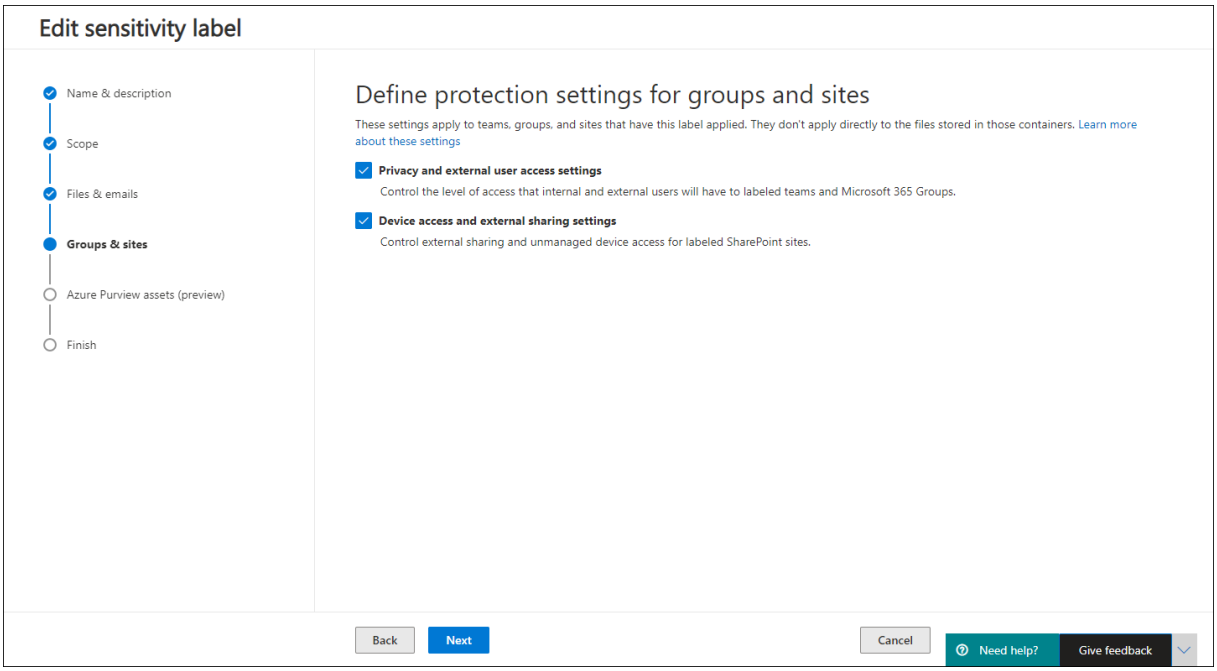
Step 2: Create or Modify your Sensitivity Labels.

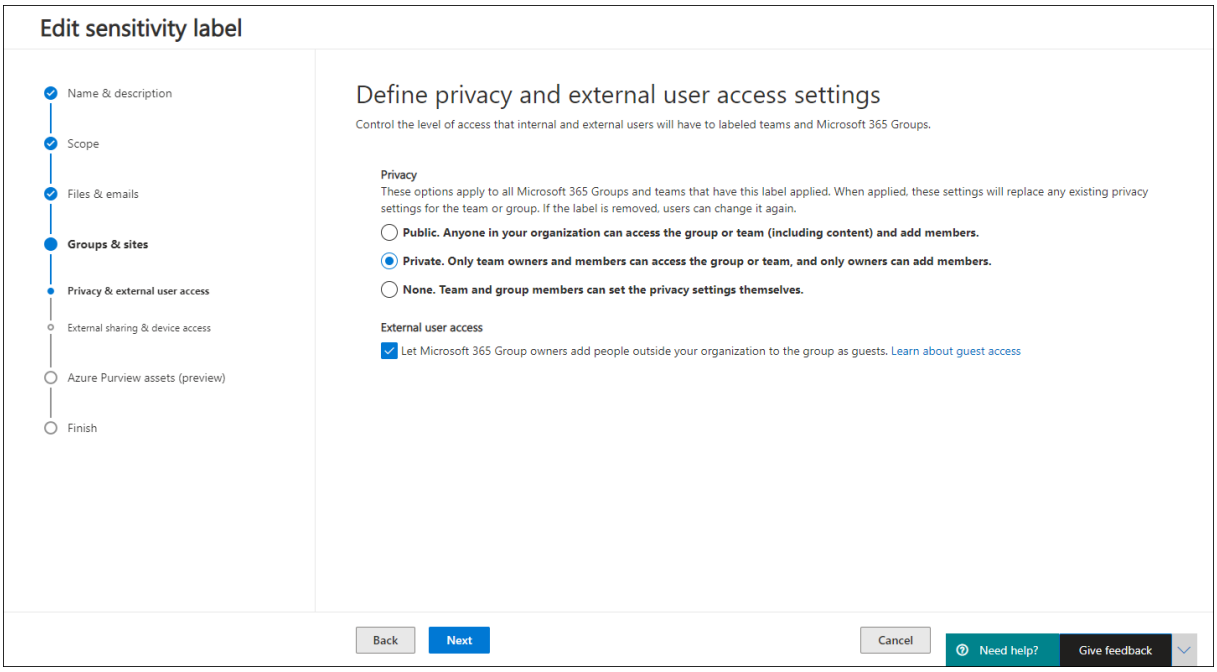
So, now that we have the functionality available, we can define our settings. In this blogpost I used the following settings

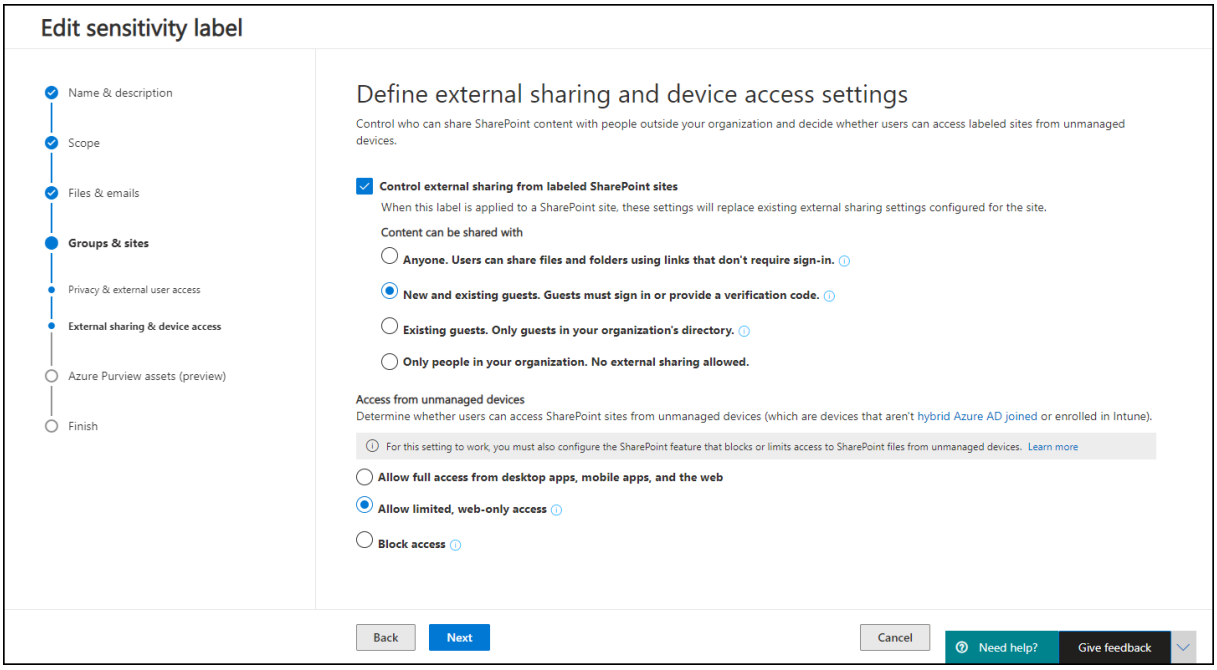
Label name	Privacy and external user access settings	Device access and external sharing settings
Public	Privacy: Public External user access: Enabled	Content can be shared with: Anyone. Users can share files and folders using links that don't require sign-in Access from unmanaged devices: Allow full access from desktop apps, mobile apps, and the web
General	Privacy: Private External user access: Enabled	Content can be shared with: New and existing guests. Guests must sign in or provide a verification code Access from unmanaged devices: Allow full access from desktop apps, mobile apps, and the web
Confidential	Privacy: Private External user access: Enabled	Content can be shared with: New and existing guests. Guests must sign in or provide a verification code Access from unmanaged devices: Allow limited, web only access
Highly Confidential	Privacy: Private External user access: Disable	Content can be shared with: Only people in your organization. No external sharing allowed Access from unmanaged devices: Block access

Next some figures showing the configuration of the Groups & Sites setting of my Confidential label


Edit sensitivity label
Name & description
Scope
Files & emails
Groups & sites
Azure Purview assets (preview)
Finish
Define protection settings for groups and sites
These settings apply to teams, groups, and sites that have this label applied. They don't apply directly to the files stored in those containers. Learn more about these settings
Privacy and external user access settings
Control the level of access that internal and external users will have to labeled teams and Microsoft 365 Groups.
Device access and external sharing settings
Control external sharing and unmanaged device access for labeled SharePoint sites.
Back
Next
Cancel
Need help?
Give feedback

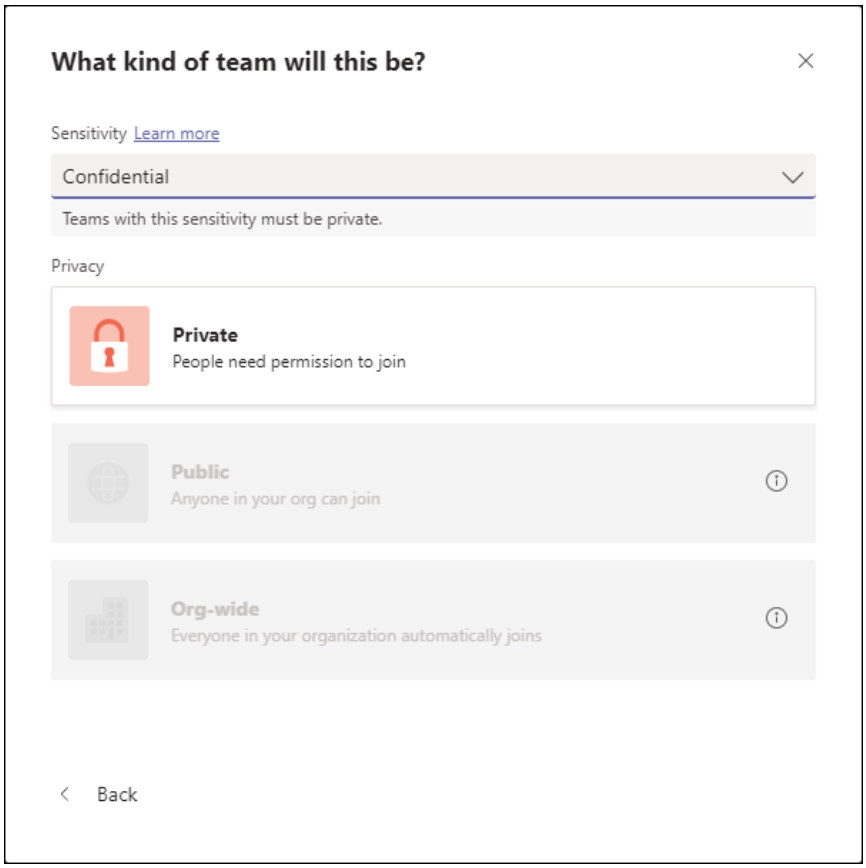

Edit sensitivity label
Name & description
Scope
Files & emails
Groups & sites
Azure Purview assets (preview)
Finish
Define protection settings for groups and sites
These settings apply to teams, groups, and sites that have this label applied. They don't apply directly to the files stored in those containers. Learn more about these settings
Privacy and external user access settings
Control the level of access that internal and external users will have to labeled teams and Microsoft 365 Groups.
Device access and external sharing settings
Control external sharing and unmanaged device access for labeled SharePoint sites.
Back
Next
Cancel
Need help?
Give feedback

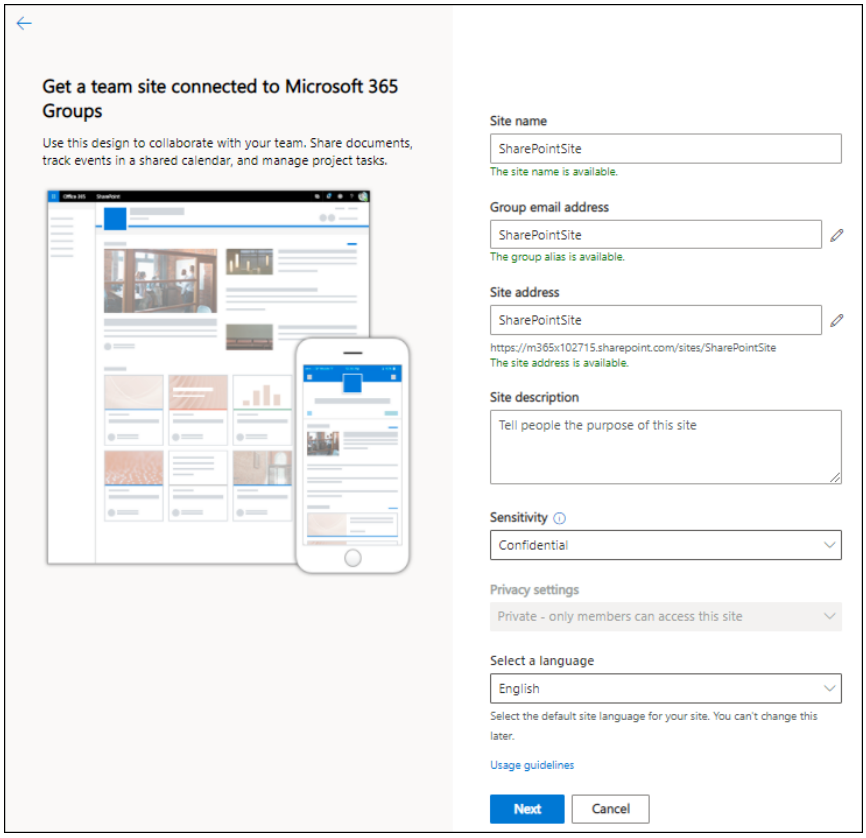


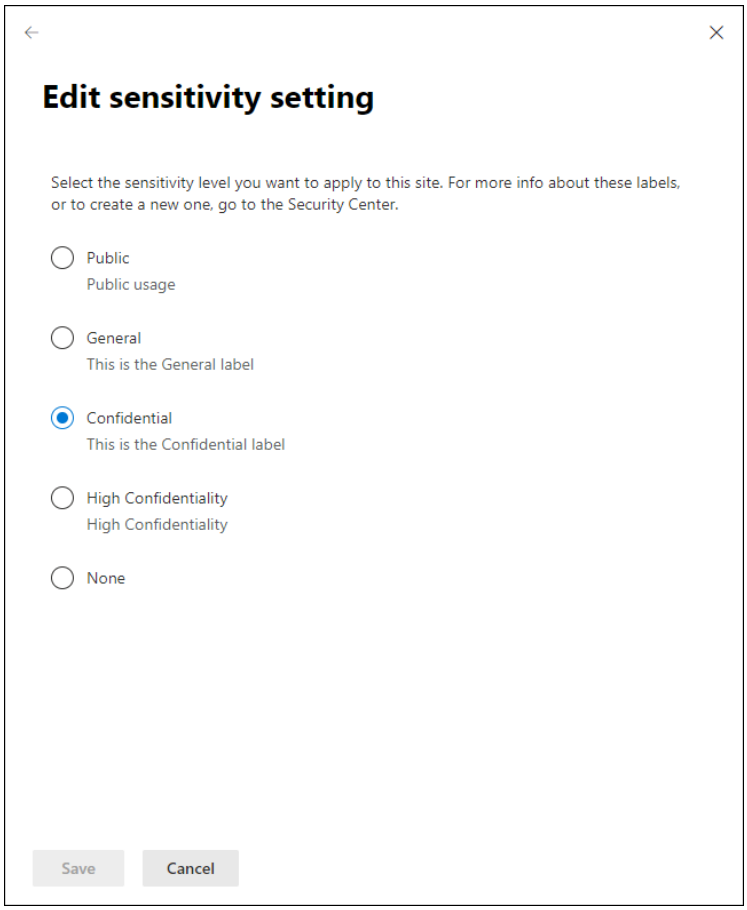


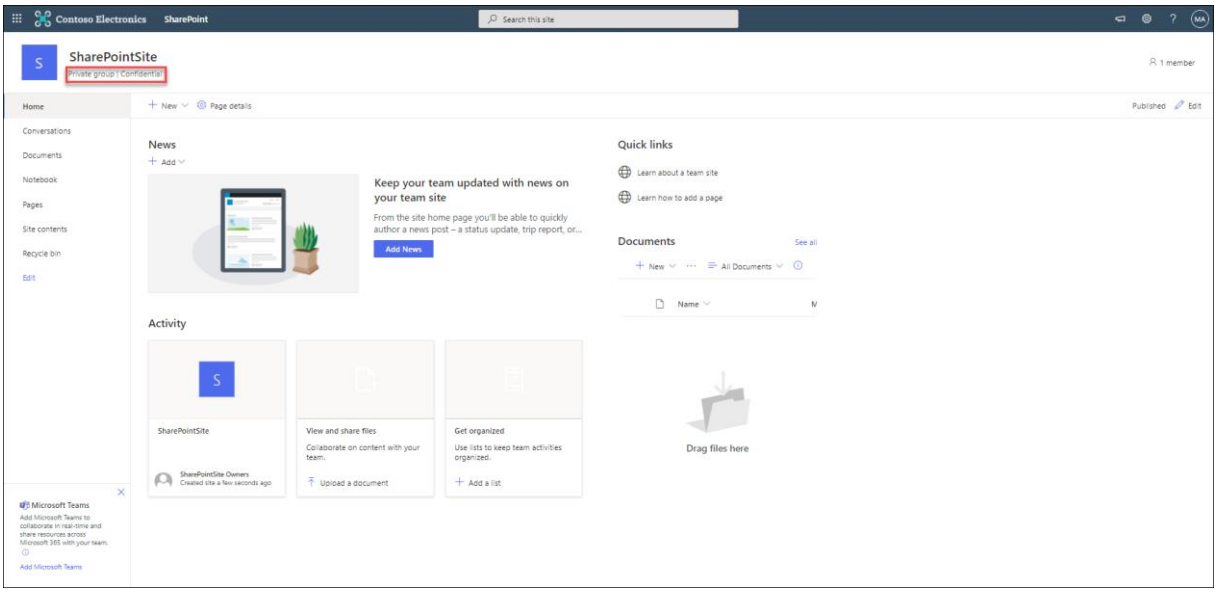
So now that the labels have been created, you will have the Sensitivity label options available when creating SharePoint and Teams environments as detailed in the slideshow below.

Note: It can take a while before you are able to use the sensitivity labels.


What kind of team will this be?
Sensitivity Learn more
Confidential
Teams with this sensitivity must be private.
Privacy
Private
People need permission to join
Public
Anyone in your org can join
Org-wide
Everyone in your organization automatically joins
Back


Get a team site connected to Microsoft 365 Groups
Use this design to collaborate with your team. Share documents, track events in a shared calendar, and manage project tasks.
Site name
SharePointSite
The site name is available.
Group email address
SharePointSite
The group alias is available.
Site address
SharePointSite
https://m365x102715.sharepoint.com/sites/SharePointSite
The site address is available.
Site description
Tell people the purpose of this site
Sensitivity
Confidential
Privacy settings
Private - only members can access this site
Select a language
English
Select the default site language for your site. You can't change this later.
Usage guidelines
Next
Cancel



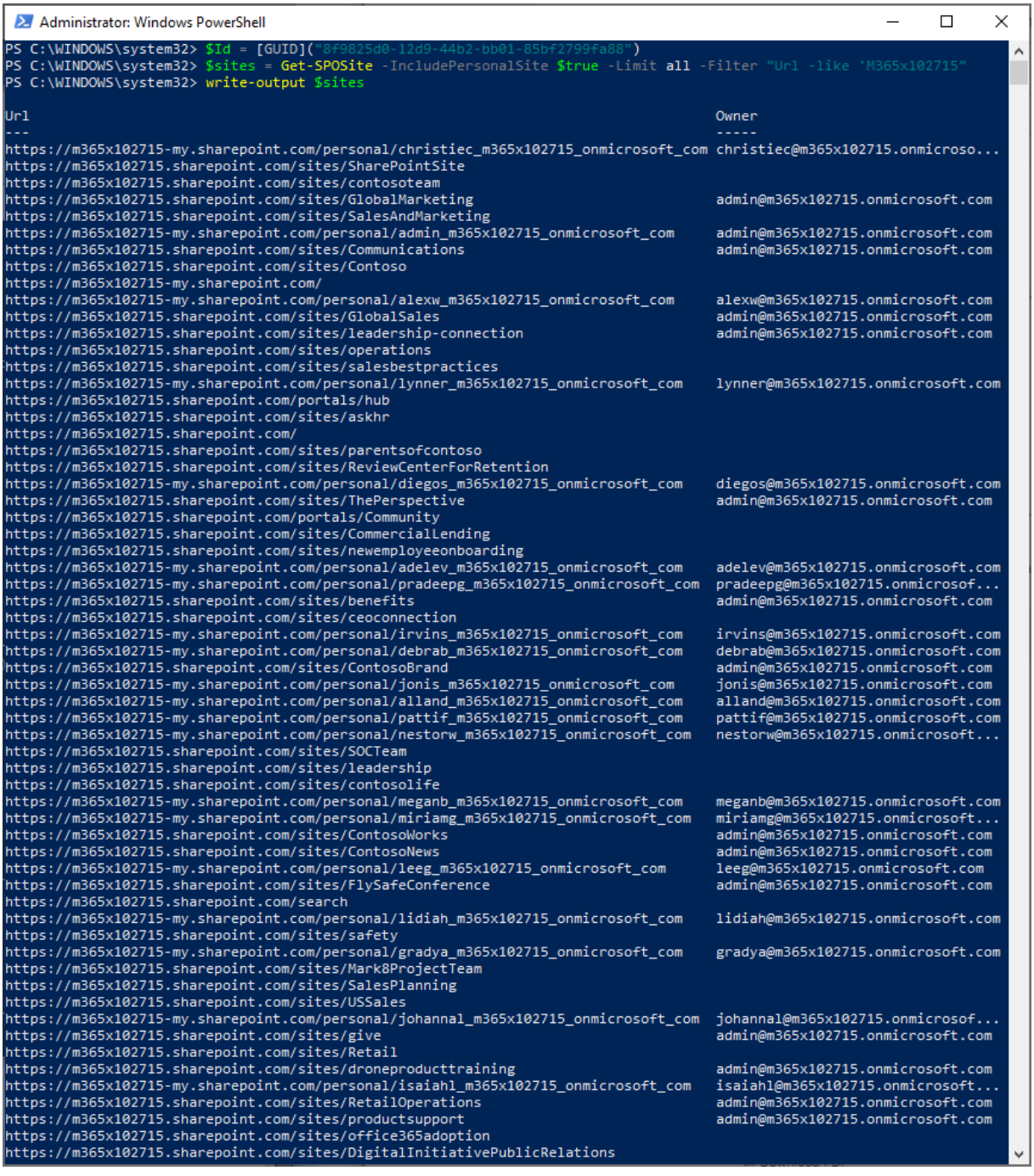


6.2.2.3.3 How to provide a Sensitivity label to your already existing Teams and SharePoint sites?

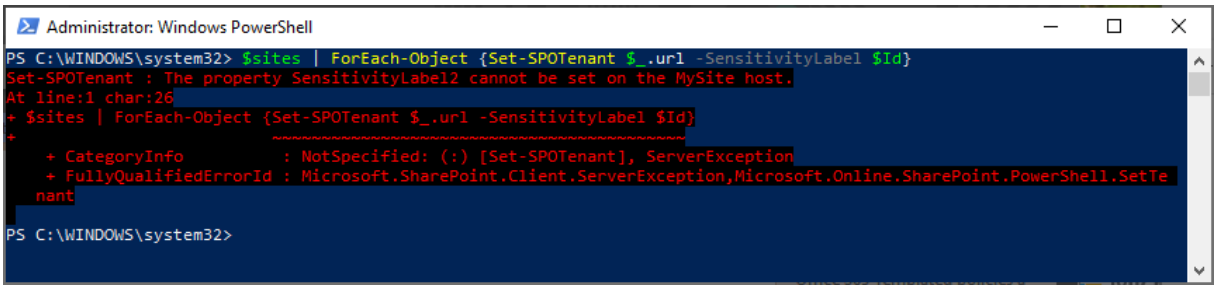
So, now that we have configured the sensitivity labels, and can use them to create new Teams or SharePoint sites, how can we handle that our current Teams and SharePoint sites are labelled as well?

For this we need PowerShell, as explained in the following article: [Use PowerShell to apply a sensitivity label to multiple sites]

Make sure that you have connected to SharePoint Online and to the Security & Compliance Center PowerShell environments by using the Connect-SPOService and Connect-IPPSSession commandlets. Retrieve the GUID of the label that you want to apply to all your existing sites using the Get-Label |ft Name, Guid command. Make sure to put the ID in a variable and enumerate the SharePoint sites by using a generic string representing your tenant. In my case this is "M365x102715"

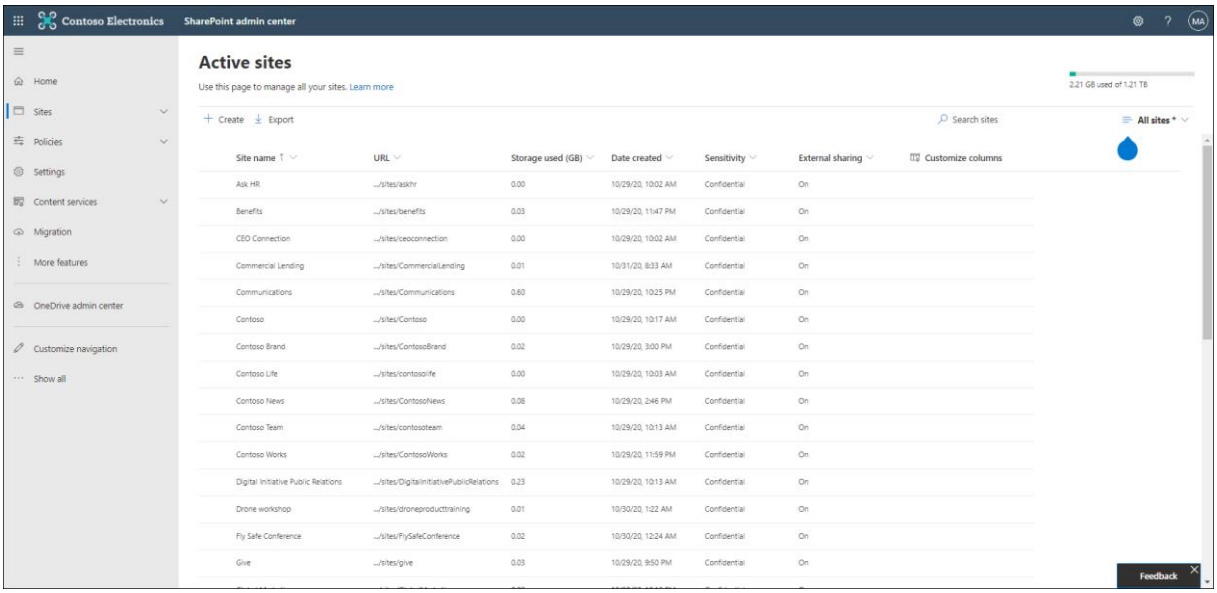


Now that we have enumerated all SharePoint sites (including OneDrive sites) we can apply the label we want, In my case I have chosen to use the Confidential label, so that by default I provide limited access and can use the GUI to make exceptions.



Apparently, you cannot set a sensitivity label on the MySite host, which is the https://m365x102715-my.sharepoint.com/ URL in my case. But this can be ignored.

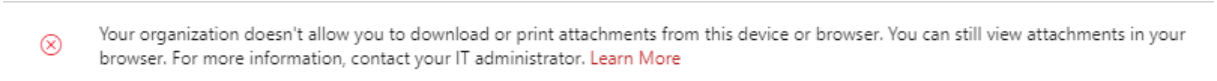
The end result is that all the SharePoint sites, and OneDrive sites will have the Confidential sensitivity label applied on the container level



Keep in mind though that you have to create a procedure from now on to make sure that the Sensitivity label gets applied to newly created OneDrive sites. Unfortunately I haven't found a way yet to set a default Sensitivity label for newly created OneDrive sites.

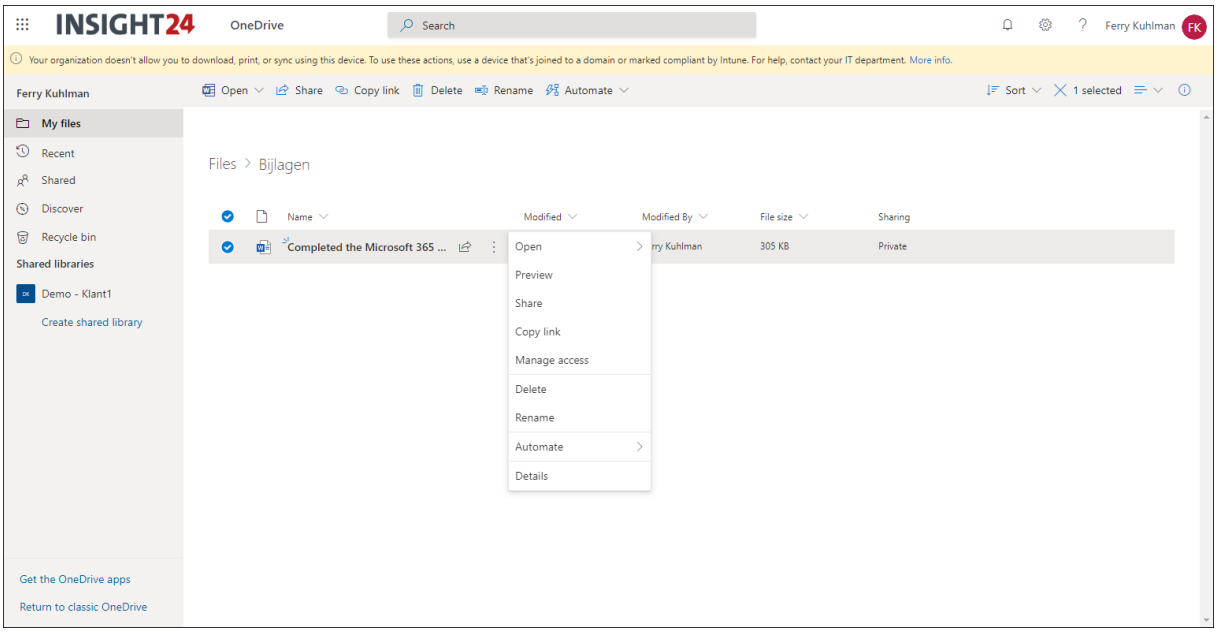
6.2.2.4 Policy behavior within Outlook Web Access

When the Conditional Access policy is applicable, the user accessing Outlook Web Access experiences the following behavior.



6.2.2.5 Policy behavior within SharePoint Online and OneDrive

When the Conditional Access policy is applicable, the user accessing SharePoint or OneDrive experiences the following behavior.



6.3 Enable SharePoint and OneDrive integration with Azure AD B2B

By default, when you share a document coming from SharePoint and OneDrive, a SharePoint B2B guest account will be created for the user you are sharing with. When links shared through this mechanism are used, they bypass Conditional Access.

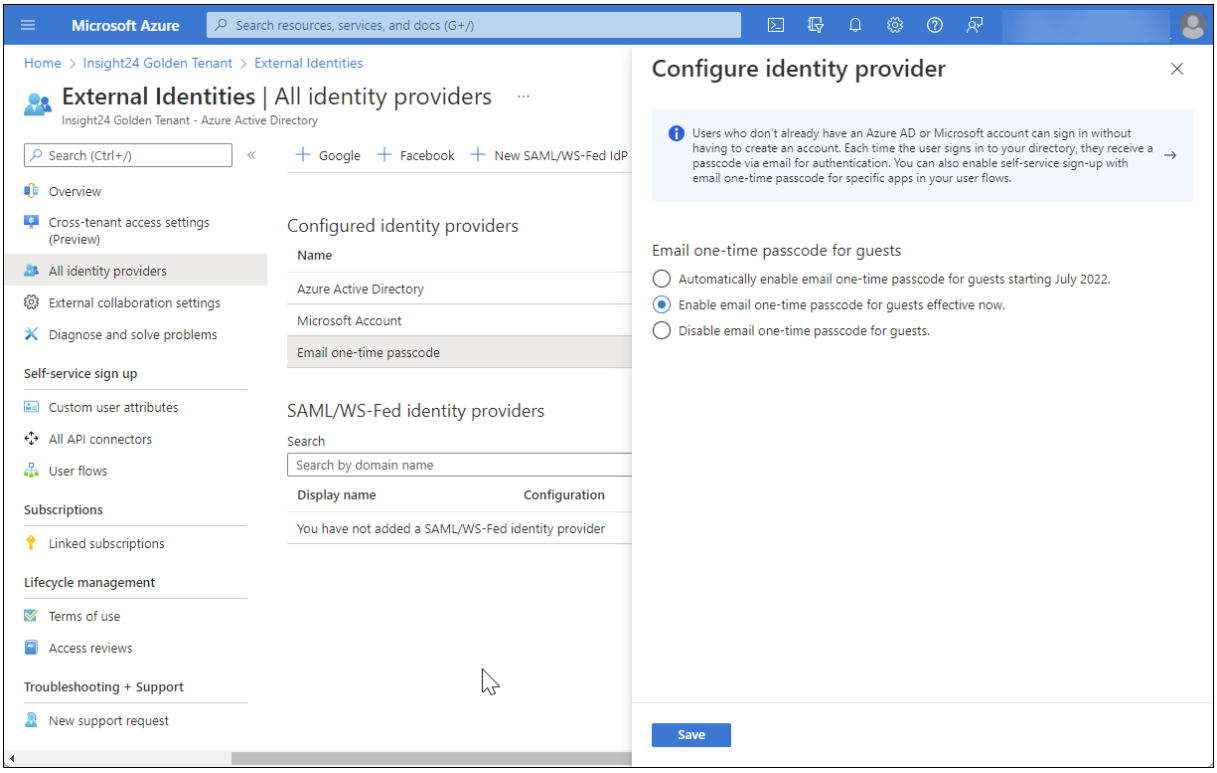
It's therefore recommended to enabled SharePoint and OneDrive integration with Azure B2B, as explained in the following article:

[SharePoint and OneDrive integration with Azure AD B2B]

The screenshot below gives an idea of the necessary steps in order to execute this change.

```
Administrator: Windows PowerS
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\KennethvanSurksum> Install-Module -Name Microsoft.Online.SharePoint.PowerShell -Force
PS C:\Users\KennethvanSurksum> Connect-SPOService -Url https://[redacted]-admin.sharepoint.com
PS C:\Users\KennethvanSurksum> Get-SPOTenant | Select *B2B*

EnableAzureADB2BIntegration
---------------------------
                      False


PS C:\Users\KennethvanSurksum> Set-SPOTenant -EnableAzureADB2BIntegration $true
WARNING: Make sure to also enable the Azure AD one-time passcode authentication. If it is not enabled then SharePoint
will not use Azure AD B2B even if EnableAzureADB2BIntegration is set to true. Learn more at
http://aka.ms/spo-b2b-integration.
PS C:\Users\KennethvanSurksum> Get-SPOTenant | Select *B2B*

EnableAzureADB2BIntegration
---------------------------
                       True


PS C:\Users\KennethvanSurksum>
```

Keep in mind that besides enabling SharePoint and OneDrive integration with Azure AD B2B, also the Email one-time passcode must be enabled. You can find this option under Azure AD\External Identities\All Identity Providers\Email one-time passcode and select the radio button "Enable email one-time passcode for guests effective now"



Some more context can be found here: [Securing SharePoint Online guest users with the Azure AD B2B experience – Identity Man (identity-man.eu)]

7 Testing and troubleshooting conditional access

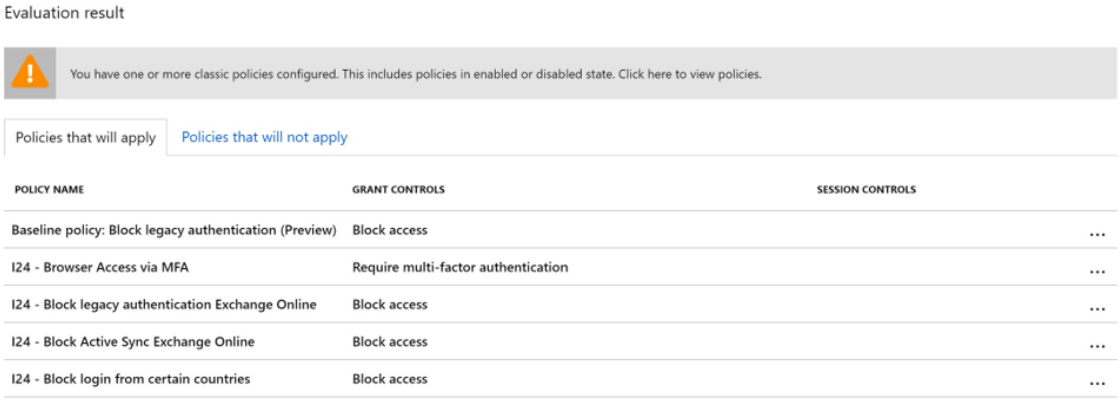
In this chapter we will go into more detail on where we can find information which can help us to test and troubleshoot Conditional Access policies.

7.1 What if tool

You can find the What if tooling by clicking on the What If icon on the Conditional Access policy overview page.

With the What if tooling you can determine which policies are applicable for a certain scenario. If you run the tooling it will give you an overview of which policies will apply, and which policies will not apply including the condition that has not been met.

A possible outcome can be:



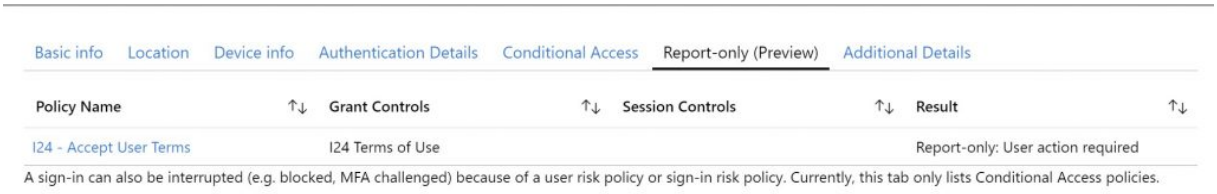
There are some things the What if tooling is not capable of displaying though, that is that the effective outcome for the user, take for example the outcome of the example above - I'm still granted access to the cloud apps even though some "Block access" controls apply.

7.2 Report-only Mode and Workbooks

By enabling the Report-only mode the conditional access is evaluated on the client instead of enforced. By using the Azure AD sign-in logging functionality we can then determine the expected behavior of the Conditional Access Policy. This can be done in two ways:

1. Using Azure Active Directory Sign-in logging

Go to the Azure AD administration portal | Monitoring | Sign-ins and select one of the listed sign-ins. Once selected, within the sign-in logging, a tab titled: “Report-only” is available. Here you can see, in this example that the “I24 – Accept User Terms” conditional access policy reports the result: ” Report-only: User action required”

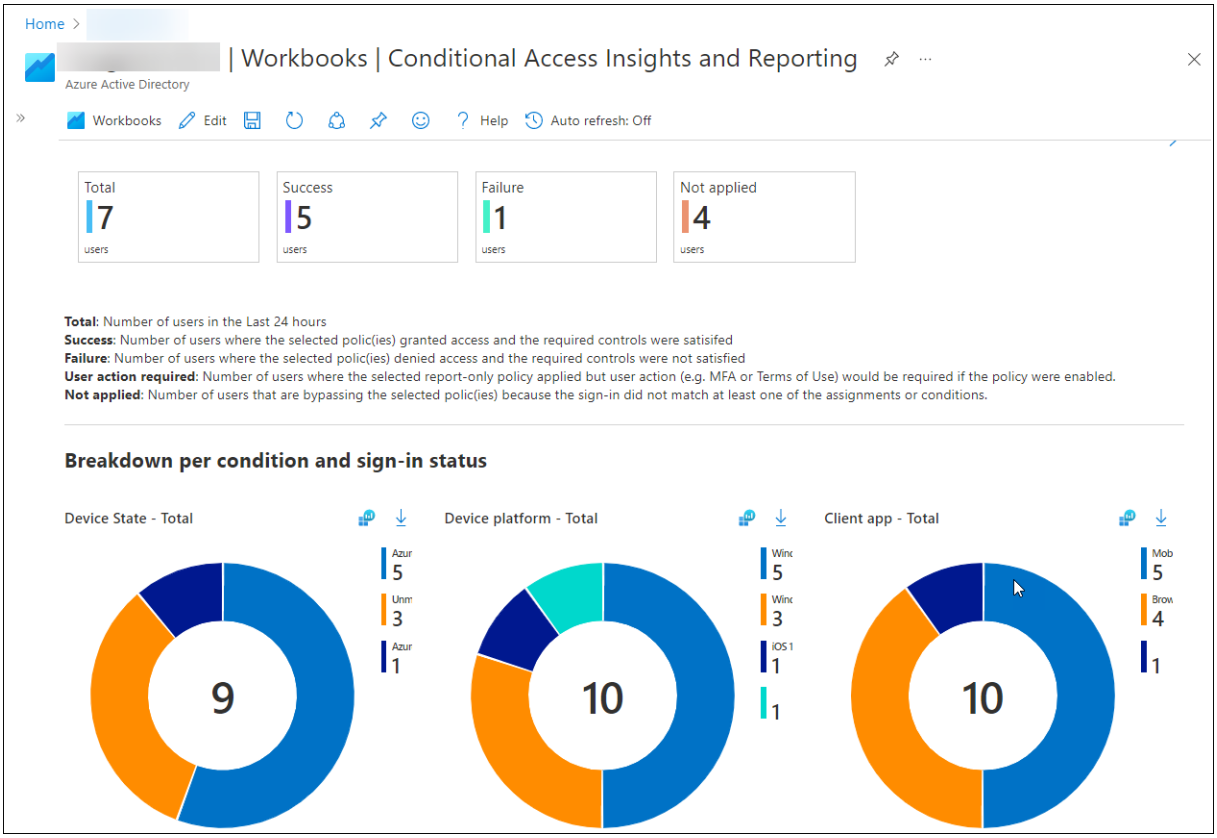


2. Using Azure AD Workbooks

Using Azure AD workbooks requires that you have setup Log Analytics and forward your Azure AD sign-in logging to a Log Analytics Workspace.

From there the following workbooks related to Conditional Access are available:

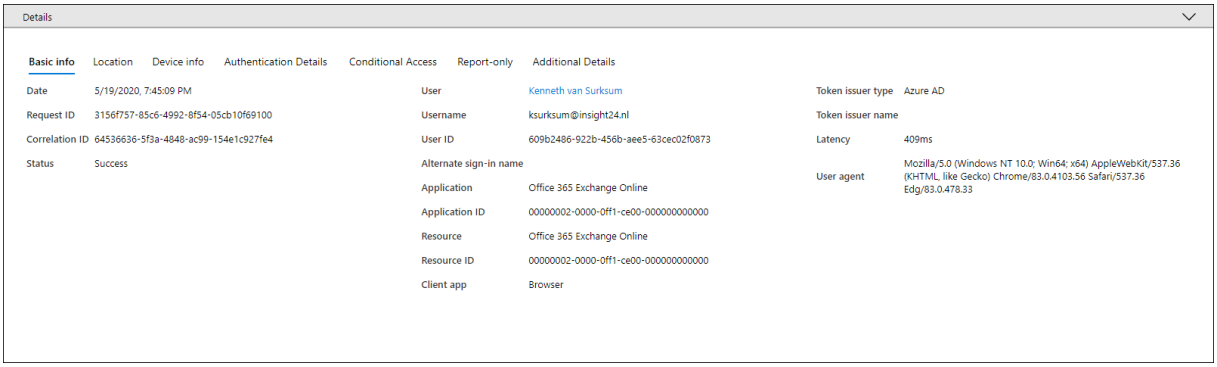
- Conditional Access Insights and Reporting
- Continuous access evaluation insights
- Sign-ins by Conditional Access Status (Deprecated)
- Sign-ins by Grant Controls (Deprecated)
- Conditional Access Gap Analyzer



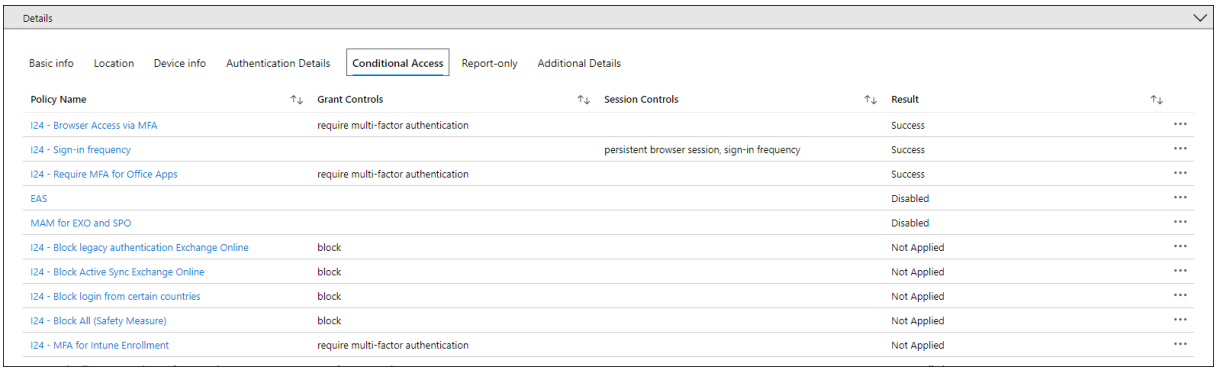
7.3 Azure Active Directory sign-in logging

Once the policy is implemented you can use sign-ins logging from Azure Active Directory. Sign-ins logging is available under the monitoring section of Azure Active Directory, in the overview you can see all the sign ins and once a sign in is selected you can find more information about the circumstances under which the sign-in took place. On the Conditional Access tab, you can find all the Conditional Access related information.

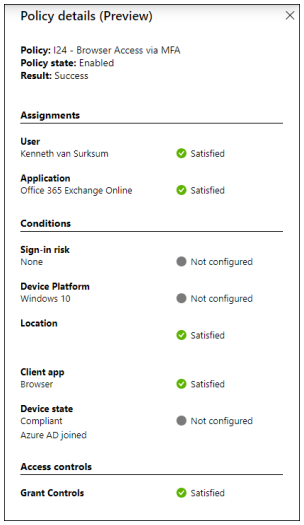
Below is an example of the outcome. On the basic page you can see the user involved, the date and time the sign in took place and what client app (Chrome browser) was used to access the cloud app (in this case Office 365 Exchange Online).



On the tab Conditional Access you can see which policies are applied for this login, whether the policy blocked access or denied access and what result was.



If you click on one of the Conditional Access policies, a new pane will pop up giving you additional information about this specific Conditional Access Policy



7.4 Where to find help and provide feedback

There are many resources where you can find help on Azure Active Directory Conditional Access, when troubleshooting at first Google/Bing is your best friend here. If those search engines don't give the expected result, you can always ask at the following forums or reach out using twitter and other social media channels.

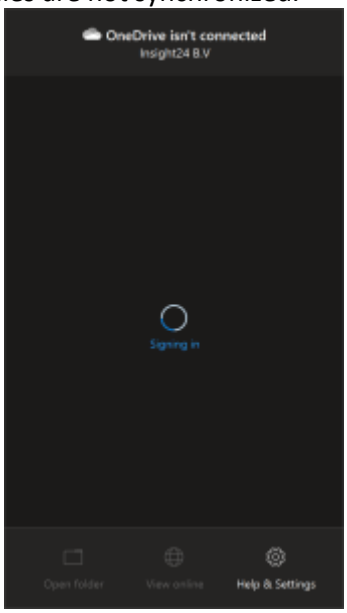
- Azure Active Directory @ MSDN - https://social.msdn.microsoft.com/Forums/en-US/home?forum=WindowsAzureAD
- Azure Active Directory @ Stack Overflow - https://stackoverflow.com/questions/tagged/azure-active-directory
- Microsoft EM+S community on discord - https://discord.gg/PjgUEkdp2E
- Modern Endpoint Management (MECM | SCCM | Intune | AzureVD | Security | MacOS | iOS) Group on LinkedIn - https://www.linkedin.com/groups/8761296/

If something does not work as expected another good source could be to check the Azure Active Directory user voice page, where a certain functionality might already be noticed by somebody else who requested the product team to solve it. You can find the UserVoice page for Conditional Access here: https://feedback.azure.com/forums/169401-azure-active-directory/category/167259-conditional-access

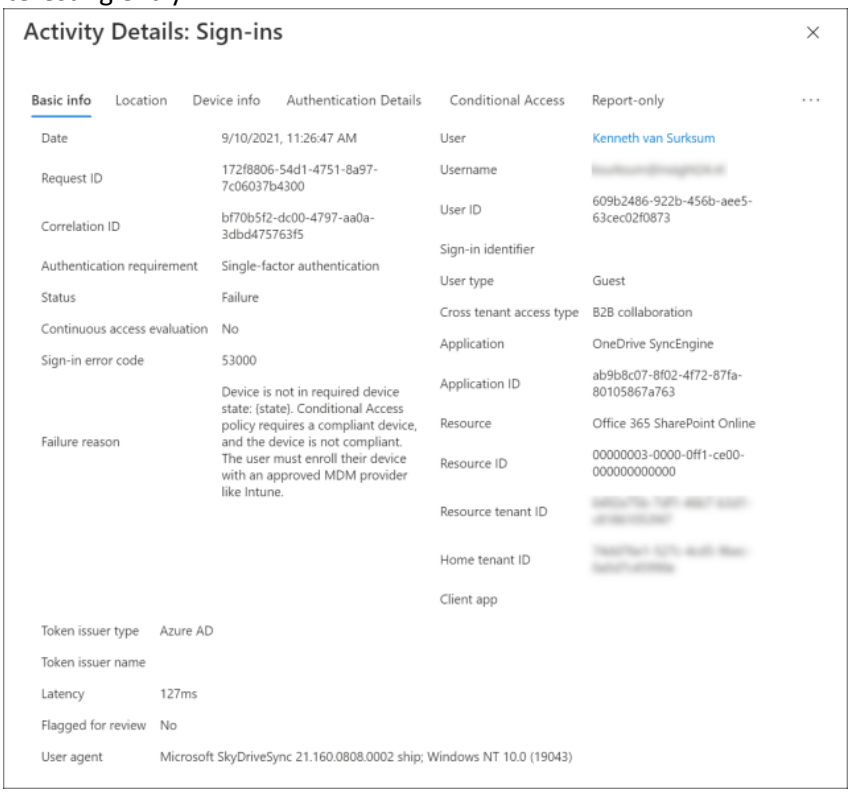
7.5 OneDrive client sign-in issues due to Conditional Access policies in Azure AD tenant where you are a guest user

This is an example of what can happen to Guest users, If you configure your own Conditional Access policies but don't take your Guest users into account.

The issue I encountered was related to the fact that I couldn't sign-in into the OneDrive client anymore. When you are not able to sign-in, you cannot open documents which are cloud hosted for example, and of course changes in files are not synchronized.



So in order to troubleshoot the issue, I started digging into the Azure AD sign-in logs, where I found the following interesting entry.



The first thing I noticed is that the User Type property was set to Guest, while normally it states “Member”. It kept me wondering why my account was a Guest user and then it all became clear. For a project which I was working on at another customer, the customer asked me for my IP address so that they could allow that IP address to access the Teams environment being created. I have setup a sync of one of the SharePoint folder within my OneDrive client. Everything was working fine until I went to the office and got another IP address which broke the ability of the OneDrive client to sign in.

7.5.1 So, what was going on?

The client defined a Conditional Access policy, which either requires a compliant device or a trusted location before access to SharePoint Online is granted. This caused my own OneDrive syn client to not be able to login and therefore was not usable until my customer changed something in their Conditional Access policy, or until I remove the sync to the SharePoint Online library hosted in the tenant of my customer, but I can only do that when I’m back in the home office where my IP is whitelisted.

7.5.2 Lessons learned

When creating a Conditional Access policy, this can have an unexpected impact even on Guest users. And in my opinion using IP addresses to exclude in a Conditional Access policy is asking for issues sooner or later. Go for Zero trust always.

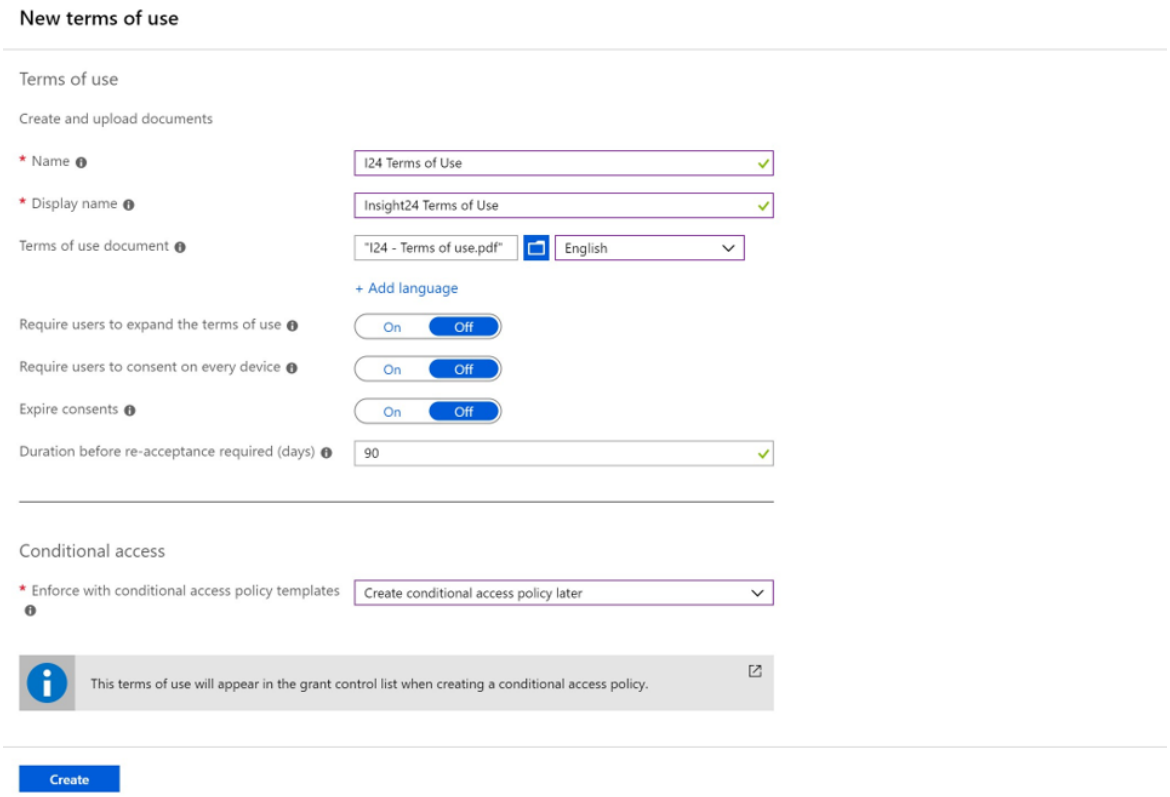
8 Modifying Conditional Access to suit your special needs.

When you want to integrate other products into your Conditional Access environment you can use "Custom controls" to include products from other vendors into your Conditional Access conditions. If a custom control is used the browser is redirected to the external service, performs any required authentication or validation activities, and is then redirected back to Azure Active Directory. If the user was successfully authenticated or validated, the user continues in the Conditional Access flow.

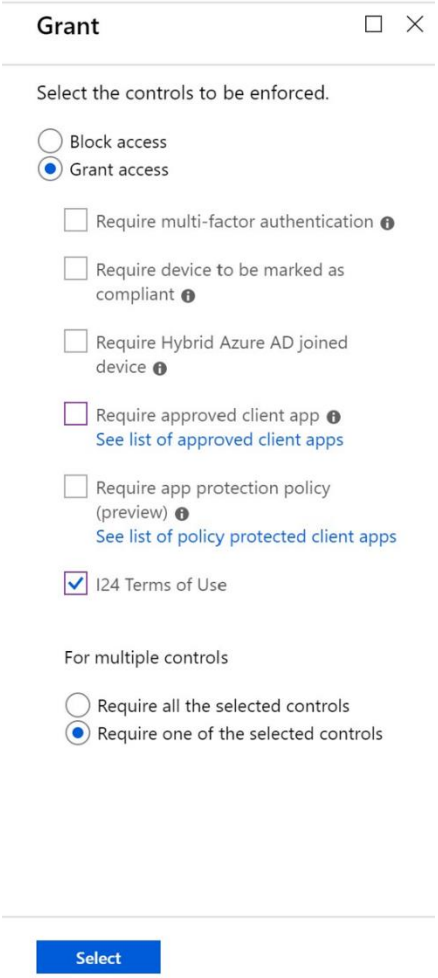
More information and some samples can be found here: Custom controls (preview)

Another thing you can do to extend the grant control with Terms of Use which users must consent with before they can access the cloud app. More information about creating the terms of use can be found here: Azure Active Directory terms of use - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/terms-of-use

In the example below I have created the terms of use for my tenant Insight24



Once created if I open any Conditional Access policy, I have an extra control available which I can select in the Grant control. In this case the user is granted access to the cloud app if the I24 Terms of Use are accepted by the user.



9 Resources and further references

9.1 Microsoft documentation

- What is Conditional Access? - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/overview
- What is Azure Active Directory? - https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis
- Azure Active Directory pricing - https://azure.microsoft.com/en-us/pricing/details/active-directory/
- Azure AD Adoption kits: https://www.microsoft.com/en-us/download/details.aspx?id=58321
- Microsoft 365 Business Service Description - https://docs.microsoft.com/en-gb/office365/servicedescriptions/microsoft-365-business-service-description
- QuickStart: Block access when a session risk is detected with Azure Active Directory conditional access - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/app-sign-in-risk
- Infographic: Control access to your data with intelligence using Microsoft EMS - https://gallery.technet.microsoft.com/Infographic-Control-access-81e7d79e
- Infographic: Comprehensive protection of Office 365 data on any device with EMS - https://gallery.technet.microsoft.com/Infographic-Comprehensive-e9a6c8c3
- Enable combined security information registration (preview) - https://docs.microsoft.com/en-us/azure/active-directory/authentication/howto-registration-mfa-sspr-combined
- Enabling limited access with SharePoint Online - https://aka.ms/spolimitedaccessdocs
- Enabling limited access with Exchange Online - https://aka.ms/owalimitedaccess
- Use app enforced restrictions - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/concept-conditional-access-session#application-enforced-restrictions
- Protect apps with Microsoft Defender for Cloud Apps Conditional Access App Control - https://docs.microsoft.com/en-us/cloud-app-security/proxy-intro-aad#how-it-works
- User sign-in frequency - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/howto-conditional-access-session-lifetime#user-sign-in-frequency
- Configure authentication session management with Conditional Access - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/howto-conditional-access-session-lifetime
- What are service dependencies in Azure Active Directory Conditional Access? - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/service-dependencies
- Reduce your attack surface - https://docs.microsoft.com/en-us/azure/security/azure-ad-secure-steps#step-2---reduce-your-attack-surface
- Tutorial: Secure user sign-in events with Azure Multi-Factor Authentication - https://docs.microsoft.com/en-us/azure/active-directory/authentication/tutorial-enable-azure-mfa
- Quickstart: Block access when a session risk is detected with Azure Active Directory Conditional Access - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/app-sign-in-risk
- Quickstart: Require terms of use to be accepted before accessing cloud apps - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/require-tou
- Control access from unmanaged devices - https://docs.microsoft.com/en-us/sharepoint/control-access-from-unmanaged-devices

- Use sensitivity labels to protect content in Microsoft Teams, Microsoft 365 groups, and SharePoint sites - https://docs.microsoft.com/en-us/microsoft-365/compliance/sensitivity-labels-teams-groups-sites?view=o365-worldwide#enable-this-preview-and-synchronize-labels
- Assign sensitivity labels to Microsoft 365 groups in Azure Active Directory - https://docs.microsoft.com/en-us/azure/active-directory/enterprise-users/groups-assign-sensitivity-labels
- Use sensitivity labels to protect content in Microsoft Teams, Microsoft 365 groups, and SharePoint sites - https://docs.microsoft.com/en-us/microsoft-365/compliance/sensitivity-labels-teams-groups-sites?view=o365-worldwide
  - Optimize reauthentication prompts and understand session lifetime for Azure Multi-Factor Authentication - https://docs.microsoft.com/en-us/azure/active-directory/authentication/concepts-azure-multi-factor-authentication-prompts-session-lifetime
  - What is a Primary Refresh Token? - https://docs.microsoft.com/en-us/azure/active-directory/devices/concept-primary-refresh-token
  - Configure the 'Stay signed in?' prompt for Azure AD accounts - https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/keep-me-signed-in
  - Configure authentication session management with Conditional Access - https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/howto-conditional-access-session-lifetime
  - Azure AD registered devices - https://docs.microsoft.com/en-us/azure/active-directory/devices/concept-azure-ad-register
  - Accessing Conditional Access protected resources in Microsoft Edge - https://docs.microsoft.com/en-us/deployedge/security-overview#accessing-conditional-access-protected-resources-in-microsoft-edge
  - Enable passwordless sign-in with the Microsoft Authenticator app (preview) - https://docs.microsoft.com/en-us/azure/active-directory/authentication/howto-authentication-passwordless-phone

9.2 Other interesting blogs

- Conditional Access posts by Peter van der Woude - https://www.petervanderwoude.nl/post/category/microsoft-intune/conditional-access/
- Conditional Access posts by Peter Daalmans - https://www.configmgrblog.com/tag/conditional-access/
- Conditional Access posts by Per Larsen - https://osddeployment.dk/tag/conditional-access/
- Conditional Access posts by Daniel Chronlund - https://danielchronlund.com/
- Conditional Access posts by Thomas Naunheim - https://www.cloud-architekt.net/
  - Thomas has an amazing project related to using Azure DevOps to implement Conditional Access which you can find here: https://www.cloud-architekt.net/aadops-conditional-access/
- Conditional Access posts by Christian Decker - https://www.derdecker.at/
- Conditional Access posts by Alex Fields – https://www.itpromentor.com/tag/conditional-access/
- CA Optics – Azure AD Conditional Access Gap Analyzer, by Joosua Santasalo - https://github.com/jsa2/caOptics
- Terraform Azure AD recommended Conditional Access policies, by Robert Brandsø - https://github.com/robertbrandso/terraform-azuread-recommended-conditional-access-policies
- How to get started with Conditional Access, by Per Larsen - https://osddeployment.dk/2018/07/01/how-to-get-started-with-conditional-access/

- Conditional Access - are you really getting the most out of it?, by Joni Nieminen - https://bloggerz.cloud/2019/01/02/conditional-access-are-you-really-getting-the-most-out-of-it-part-2-of-2/
- Implementing Modern Security Tools – Part 3 – Conditional Access, by Maurice Daly - https://www.scconfigmgr.com/2019/02/19/implementing-modern-security-tools-part-3-conditional-access/
- Azure Active Directory and Office 365: Conditional Access, by Jethro Seghers - https://regarding365.com/azure-active-directory-and-office-365-conditional-access-8bc616a392b2
- My favorite Conditional Access Policies for the SMB, by Alex Fields - https://www.itpromentor.com/conditional-access-faves/
- Conditional access (zero trust) is the most important EUC movement since mobile and cloud, by Jack Madden - https://www.brianmadden.com/opinion/Conditional-access-zero-trust-is-the-most-important-EUC-movement-since-mobile-and-cloud
- Diverse articles on Conditional Access from the Practical 365 team - https://practical365.com/tag/conditional-access/
- Bypassing Conditional Access Device Platform Policies, by Nicola Suter - https://tech.nicolonsky.ch/bypassing-conditional-access-device-platform-policies/