Uses for identity management measures

So far, you've learned that identity management is a core component for access control in cloud environments. Identity management focuses on provisioning, managing, and deprovisioning identities like users and groups. Key identity management security measures you've learned about include Role-Based Access Control (RBAC), Single Sign-On (SSO), and Multi-Factor Authentication (MFA). You also learned how Identity and Access Management (IAM) services can help you manage these and other types of security measures.

In this reading, you'll explore scenarios where you might use identity management security measures.

RBAC scenario

Role-Based Access Control (RBAC) is a method of controlling access to resources based on the roles assigned to users. This method helps ensure users only have permissions and access to the resources necessary for their job. RBAC also lets you grant access to a collection of users via groups. Different cloud providers have their own RBAC systems, and in many cases, different products have their own RBAC features.

In this RBAC example, you'll learn how you might provide users limited access to a Google Cloud project by assigning RBAC roles, or sets of permissions to users and / or groups.

A company plans to provide a new service for its customers and needs to give temporary network access to a group of external, third party developers to help build the new service.

At the start of the project, the company's internal stakeholders provision external resources, like the tools and documentation they need to help develop the service. They also provide oversight of the external resources. The external developers leverage the resources to help them build the service, but they can't create new resources on the company's network. Only the internal users can do that.

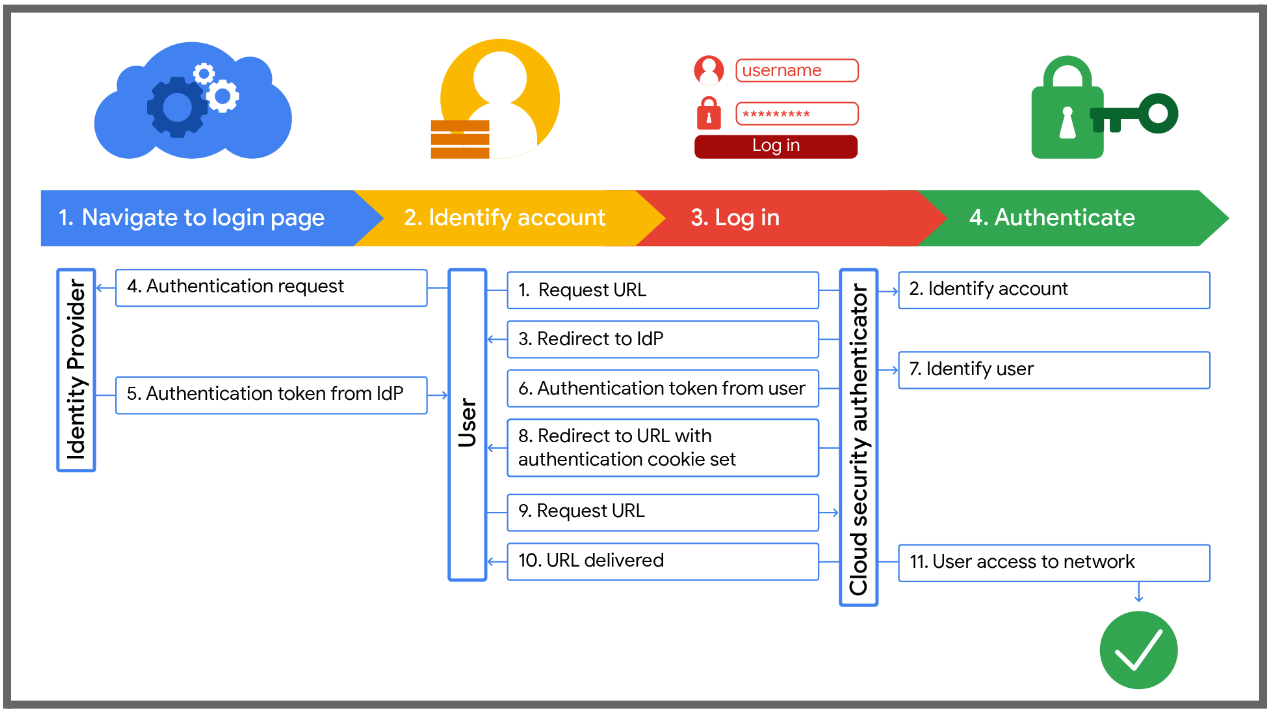
Keep in mind, those with super admin roles should stay logged in long enough to do their tasks, then sign out immediately. Only use the super admin account when you need it. You should delegate administrator tasks to user accounts that have limited admin roles. Users should only have access to the tools and resources they need for their specific tasks.

SSO scenario

SSO is a technology that enables users to log into multiple services with one set of credentials, also called logins. With SSO, you can ensure that your existing identity provider (IdP) remains the system of record you use to authenticate users.

Consider the earlier example of the company creating a new service with external developers. In this scenario, you'll want to have the external developers create an SSO authentication option so the company's customers can use their existing credentials to authenticate. Once a customer logs in to the company website, they can update their account settings, request services, check their statement, pay a bill, and perform other banking tasks.

This graphic details the steps each external user, or customer will go through to log in through your company's sign in page, designated external IdP, and endpoint.


username
****
Log in
1. Navigate to login page
2. Identify account
3. Log in
4. Authenticate
Identity Provider
4. Authentication request
5. Authentication token from IdP
User
1. Request URL
3. Redirect to IdP
6. Authentication token from user
8. Redirect to URL with authentication cookie set
9. Request URL
10. URL delivered
Cloud security authenticator
2. Identify account
7. Identify user
11. User access to network

Here's a breakdown of these steps:

1. The user navigates to the page with the console they want to log into from their browser.
2. The authenticator identifies that the user is not logged in.
3. The authenticator sends the user a redirect to the IdP.
4. The user's browser sends an authentication request to the IdP.
5. The IdP sends an authentication token to the user.
6. The authentication token requests the user to log in to a sign-in page with their username and password, and select the Login button to submit the form.

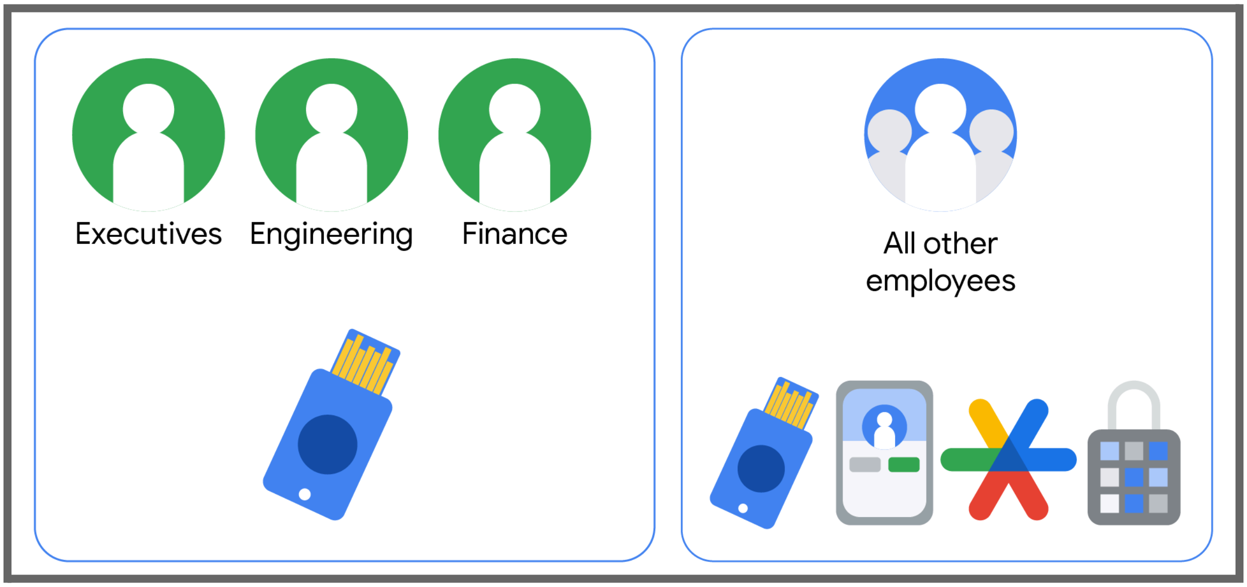
7. The authenticator identifies the user.
8. The authenticator redirects the user's browser to a URL with an authentication cookie set.
9. The user requests a URL from the authenticator.
10. The authenticator delivers the user to the URL.
11. The authenticator redirects the developer to that original network URL.

This SSO login example demonstrates the value of federation. When you use SSO, you don't need to recreate duplicate identities for external users in your own IdP. With SSO, users sign in with a single set of credentials, and can access multiple applications without having to re-enter their credentials each time.

MFA scenario

MFA is a security measure that requires a user to verify their identity in two or more ways to access a system or network, like a password combined with a fingerprint scan.

Consider the earlier example of the company creating a new service with external developers. Keeping the new development work secret from the general public is critical to help maintain an edge over competitors. Right now, the company's network security only requires one form of authentication, and if a password is compromised, the malicious actor now has the same access and permissions as the worker who owns that password. So, you plan to add an extra layer of security by requiring MFA, where all users in the company prove their identity using multiple methods. The form of MFA you and your security team implement is 2-step verification (2SV).



This graphic details various multi-factor authentication options with two subsets of users. The all other employees group includes one subset that can access network resources using a physical key or a digital key, like a 2SV code to their handheld device, Google Authenticator app, or generated access codes. The second subset includes an *exception* group of **Executives, Engineering,** and **Finance** employees who can only use a physical key to gain network access.

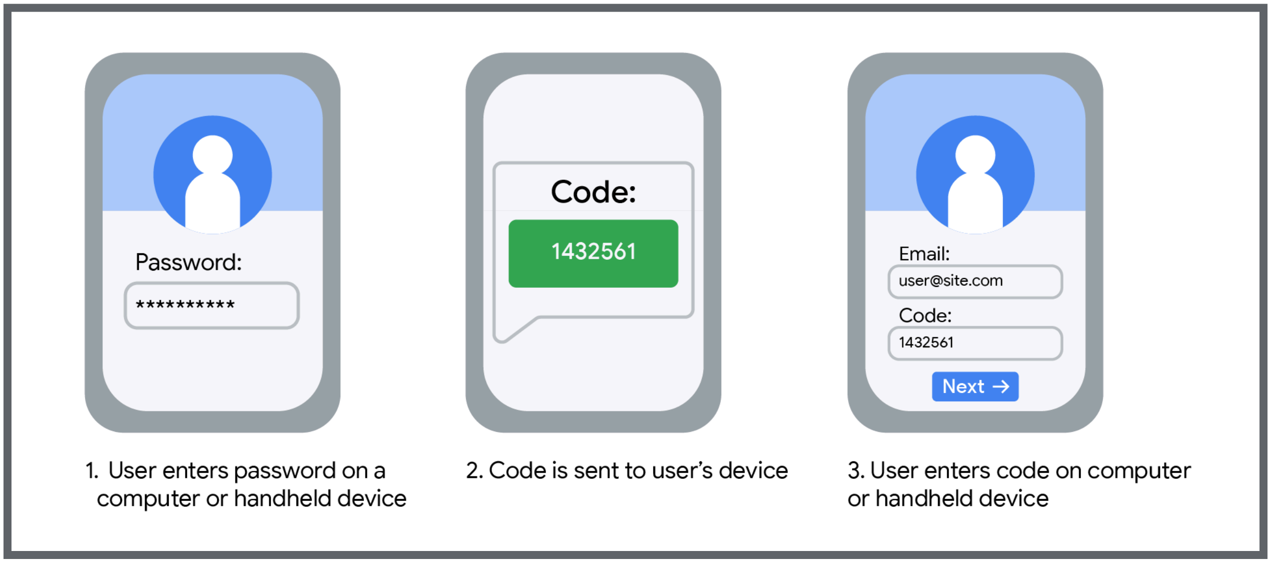
When designing the MFA, your security team decides to take a risk-based approach. The team differentiates the second factor based on the criticality of the access a user has to the system and the sensitivity of the data a user has access to.

Top level users will need to use security keys. These users include those who have executive responsibilities; work directly with the most sensitive data, like security; access administrative systems; and work directly with sensitive employee information, like human resources.

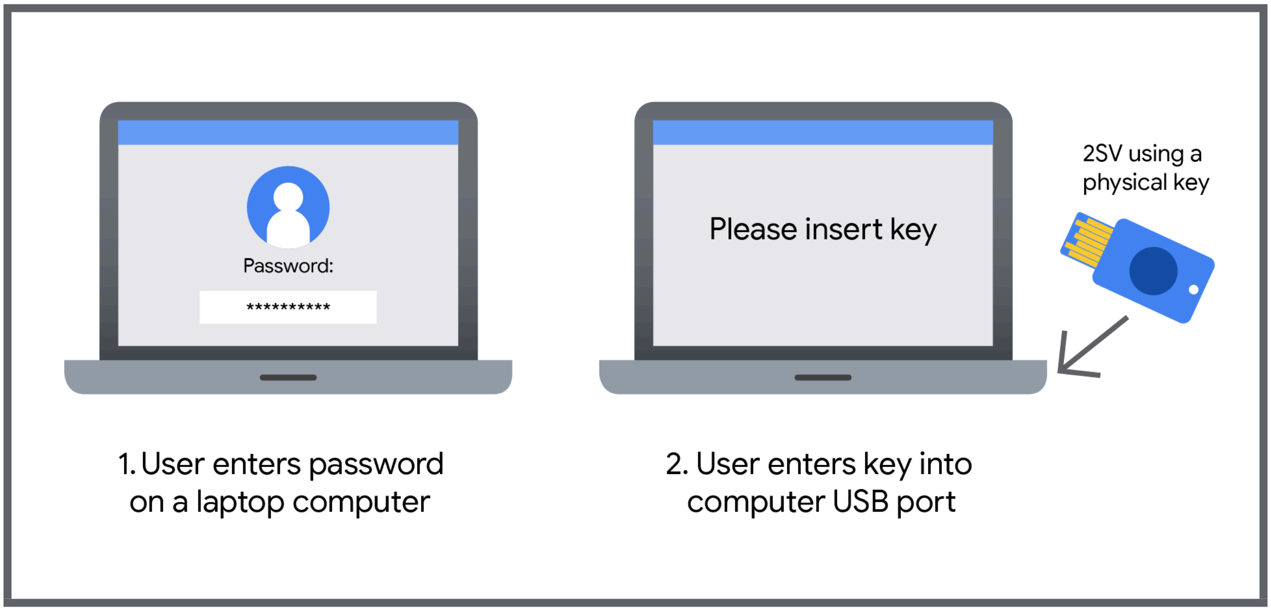
A security key is a physical device that works alongside your username and password. The key verifies your identity to a site or app and uses physical connections. For example, you can plug the key into a USB port on a computer and it performs a cryptographic handshake with the locked site. The login is stopped if the site can't validate your login credentials. Security keys may also use biometric authentication, like checking fingerprints.

These physical keys are distributed as soon as the authentication is set up. They provide the strongest security of all the 2SV methods. For other users, your security team sets up a system that sends them codes directly to their email app or mobile device's texts app. By using MFA through 2-step verification, you make it much less likely anyone who should not have access will be able to get it.

Next, you'll learn about the 2-step verification process with two different login methods: using a handheld device and using a physical key.


Password:
****
Code:
1432561
Email:
user@site.com
Code:
1432561
Next →
1. User enters password on a computer or handheld device
2. Code is sent to user's device
3. User enters code on computer or handheld device

This graphic represents the 2-step verification process using a handheld device and digital key. The left part of the graphic features a device's monitor with a site's input for a user to enter their password on the screen. In the middle is the user's mobile device featuring a code, sent from the site to authenticate the login. The right side of the graphic features the next screen the app displays on the user's device with a prompt to enter their full email and the input code. To complete the verification, the user can then click the Next-> button and log in to the app.


Password:
****
Please insert key
2SV using a physical key
1. User enters password on a laptop computer
2. User enters key into computer USB port

This graphic represents the 2-step verification process using a physical key. The left part of the graphic features a laptop device with a site's input for a user to enter their password on the screen. In the middle is the user's full laptop, with the app prompting the user to insert their physical key into their laptop USB port to authenticate their login. Once the user plugs the key into their computer, the verification process will be complete.

Auditing for accountability and compliance

Finally, you'll learn how IAM services can help you audit your security measures with respect to authentication, authorization, and auditing (AAA) credential handling. This reading's scenarios covered authentication and authorization activities, like logging into your network using MFA. The user either successfully completes the MFA challenges or fails the MFA challenges.

All login attempts are logged in the admin log or activity logs and serve as records you can audit to ensure accountability. You can also audit the role assignment activities in these logs as part of your organization's need to comply with annual review and attestation. Updating your

IAM policies for least privilege access can help ensure that the users who have access, and still need it, continue to have access.

Key takeaways

You can grant access to resources in the cloud using identity management methods like RBAC, SSO, and MFA. These security measures help you control what resources an authenticated user has access to, so the group or user only has access to system resources they need to get their work done. These measures will also help prevent malicious actors from getting access to user accounts, even if passwords have been compromised.