



www.cybersecuritycohort.com

Cyber Security Cohort Podcast hosted by Heather Holliday

Season 02 : Episode 04 | Access Management Solutions

In today's episode, we'll explore several common Access Management Solutions. In this episode we'll compare and contrast options such as Federated Identity Management, Same Sign-On, Multifactor Authentication and Single Sign-On. We'll take a look at what makes each option unique so you can better understand why there is more than one option.

We'll also talk about the power of combining these solutions and using them in tandem. It's important to note that identity access management solutions don't have to be used in isolation. Oftentimes the best strategy for access management is to combine different types of solutions to add complexity, and improve security.

First, let's talk about Federated Identity Management, or FIM. For more information on FIM, I turned to an article from Onelogin. [According to this article](#), Federated Identity Management works on the basis of a mutual trust relationship between service providers and identity providers. Think about it this way: the service provider is your application. The identity provider is the middleman who manages the user credentials, and basically serves as the gatekeeper for determining whether or not a user has the correct authorization to access the given application. As we've seen with password managers, if the identity provider becomes compromised in any way, this can mean bad news for all of the service providers that have been doing business with the identity provider. So, while FIDs are commonly used, they aren't without their own set of security concerns.

Another option is called **same** sign-on. While **same** sign-on carries the same acronym, SSO, as another access management solution, called **single** sign-on, these two solutions are actually not identical. The differentiating factor is actually right in the name. Same sign on means that you're going to use the same credentials over, and over, and over again to sign in across different systems and applications. Single sign on, however, is going to use a one time sign in of your credentials to allow access which propagates across all of the applications and systems as you use them.

And finally, there's multifactor authentication. We talked about multifactor authentication and season two episode two, "Passwords, Passphrases and PINs." So if you didn't catch that episode you'll want to make sure to review it. Or, you can always catch the short version on my YouTube channel, @h2comm, where I provided a short definition of multifactor authentication in the video titled, "Multifactor Authentication - CAM 2023." As a side note, this is part of the video series that I completed for cyber security awareness month in 2023.

As for multifactor authentication, the point I'll make here is that it can actually be combined with a single sign on (or other access management solutions) to strengthen security and simplify what a user needs to do in order to gain access to their systems and applications.

So, now that we've covered a simple overview on access management options, let's take a deep dive into single sign-on. This access management solution is one of the most commonly used, particularly among corporations.

What is single sign-on you wonder? According to Tech Target, "Single sign-on is a session and user authentication service that permits a user to use one set of login credentials." For example, a user provides their credentials once in order to access multiple applications.

While single sign-on is commonly used in enterprise situations, small and midsize businesses often use single sign on as well. What's interesting about single sign on is that it makes it easy for users to utilize multiple applications. In modern day business, this is very important. After all, most business users aren't just using a word processing program or reviewing email. Chances are, they also have to access different systems that are uniquely accessed in order to prevent unauthorized use or access to the data housed within.

So there is a high value proposition for using a single sign-on method as it provides two key features:

- 1) It's more productive. Users can flow in and out of different systems and applications seamlessly, with their credentials being managed "behind the scenes." After all, time is money, and the faster we can move from one task to the next, the more we can get accomplished in a day.
- 2) It's a better user experience. Many users aren't as passionately enthusiastic about cyber security as we are. They just want to be able to get work done and when they have to spend time key and rekeying the credentials all day long, they find it rather annoying. When users are annoyed, they tend to go looking for their own ways to "work around" the system. This can create new and different cyber security risks, AND it's also risky behavior for the employee. After all, they may (intentionally or not) bypass or ignore policies put in place...which could mean disciplinary action. So, there's a case to say that making a better user experience also helps to keep users on the right path.

According to Amazon Web Services, there are several different types of single sign-on options.

The first of these is OAuth, or open authorization. OAuth is commonly "used to provide secure access to protected resources"¹. Tech Target also gives us a definition for open authorization as, "the framework that enables the end users account information to be used by third-party services such as Facebook without exposing the users password." This is important, because keeping your password obfuscated from other applications is part of what keeps it safe.

Next, is SAML, or Security Assertion Markup Language. SAML is an Extensible Markup Language, or XML, -based format. [Onelogin has a helpful overview article](#) which provides information on how SAML works and why it's able to perform these tasks without relying on cookies, as was required in the past. "It achieves this objective by centralizing user authentication with an identity provider. Web applications can then leverage SAML via the identity provider to grant access to their users." SAML is a popular option among enterprise organizations. SAML leverages a trusted relationship with an identity provider in its authentication protocol. This trusted relationship is what allows the user to gain the access they need. So, what's the value of the SAML approach? According to AWS, one key advantage of using SAML is that it provides "better security and flexibility, as applications do not need to store user credentials on their system." It accomplishes this through the trusted relationship with the identity provider.

The third single sign-on option is Kerberos. Kerberos uses a ticketing schema to accomplish the single sign-on capability. According to simplilearn, Kerberos was developed at the Massachusetts Institute of Technology (MIT) in the 1980s and it is a common solution used by a number of operating systems today. Key elements of Kerberos are its cryptographic secret key, a trusted third party and its ticketing system. Kerberos shares the secret key with the trusted third party, known as the Key Distribution Center (KDC). The KDC then uses a ticketing system to authenticate and obtain a ticket, known as a Kerberos ticket-granting ticket (TGT). I encourage you to read [the full article from Simplilearn](#) to learn more about Kerberos and its advantages, such as limited lifetime for key tickets, mutual authentication and reusable authentication.

Finally, developers can also select OIDC or OpenID Connect. You've probably seen and used this service before – whether you knew it or not. Have you ever been asked by an application if you wanted to sign into their account by using your Google account, Facebook account or other common credentials. That, in a nutshell, is OIDC. By signing into one account, you are granted access to another. So, how does this work? It's built on the concept of tokens. In fact, [an article from Microsoft](#) provides an excellent overview of the process. Here's what they had to say:

“A typical OIDC authentication process includes the following steps:

1. A user goes to the application they wish to access (the relying party).
2. The user types in their username and password.
3. The relying party sends a request to the OpenID provider.
4. The OpenID provider validates the user's credentials and obtains authorization.
5. The OpenID provider sends an identity token and often an access token to the relying party.
6. The relying party sends the access token to the user's device.
7. The user is given access based on the information provided in the access token and relying party.”

As a user, it's important to understand the “relevant data” being shared through the tokenization process could include data such as your email address and name. This is why you should use this option with caution.

Now that we've talked about the different single sign-on options, let's consider the advantages and disadvantages of its use. Even though it's convenient to use, it can make it easy for an attacker who gains control over the single sign-on credentials to access all of the applications that user has a right to access. This could mean an increase in the potential damages and impact of those damages. So you'll often see single sign-on used with two factor or multifactor authentication, which is something that we continue to talk about because it is so important.

I also encourage you to explore the resources and references I've used to put this episode together. For example, Amazon Web Services, or AWS, has [an article about single sign-on](#) and shares several reasons why it's important. They list examples like strengthening your password, security, improving productivity, reducing cost, improving security posture and providing a better customer experience.

All of these things are really important and I can tell you as a single sign on user that I enjoy having the convenience of one to remember and that makes it easier for companies to require users to have more lengthy and complex passwords so if I only have to remember one. Because I'm using single sign-on, it's OK that my password has to be long and complex. In my world, productivity is no small thing. So, if I have to stop what I'm doing and login and remember a new password to a new application every time I need to switch from one resource to another it's definitely going to slow me down. Not only do I have to stop and pause and think about what those different passwords are but I also have to go through the authentication process with each and everyone of them. So single sign on is definitely a winner for productivity.


When it comes to security posture, we can say it's a little bit of a tossup. Single sign-on does minimize the number of passwords per user, which can reduce the risk of security events that targets, but as we said before it can create a situation where once you have the keys to the kingdom you get the entire kingdom.

Finally, I want to emphasize that when evaluating your options, it's important to fully understand both the benefits and potential pitfalls of your access management solution.

Thank you for joining this episode of the Cyber security Cohort. This is your host, Heather Holliday. Join us next time for another step in our journey of 1000 miles toward cyber security expertise.

Episode Notes & References

Information shared in this episode came from personal experience. More information on these topics can be found by searching these references.

- CompTIA Security+ Get Certified Get Ahead: SY0-601 Study Guide by Darril Gibson: <https://www.amazon.com/CompTIA-Security-Get-Certified-Ahead/dp/B096D1LGSK>
-  What is Single Sign-On Authentication? And...How Does it Work? (SSO)
- Tech Target: What is single sign-on (SSO)? <https://www.techtarget.com/searchsecurity/definition/single-sign-on>
- AWS: What is SSO (Single-Sign-On)? <https://aws.amazon.com/what-is/sso/>
- Onelogin: What is Federated ID?: <https://www.onelogin.com/learn/federated-identity>
- Onelogin: SAML Explained in Plain English: <https://www.onelogin.com/learn/saml>
- What is Kerberos?: <https://www.simplilearn.com/what-is-kerberos-article>
- Microsoft: What is OpenID Connect?: <https://www.microsoft.com/en-us/security/business/security-101/what-is-openid-connect-oidc>