1Password

EBOOK

Why SSO is not enough for identity security

Contents

Introduction

Today, every company is a technology company, and every employee is a tech decision-maker. Employees work from anywhere, adopt their own tools, and increasingly interact with AI agents as much as human coworkers. This freedom has fueled a generational leap in productivity, but it's also increased the spread of the Access-Trust Gap. The Access-Trust Gap refers to the security risks posed by unfederated identities, unmanaged devices, applications, and AI-powered tools accessing company data without proper governance controls.

Tools like single sign-on (SSO) integrate with a company's identity provider (IdP) to allow users to authenticate to multiple applications via a single login. By reducing the number of access points and employee credentials, SSO reduces a company's attack surface. SSO also makes it easier for IT and security teams to provision and revoke access to applications via the IdP instead of manually managing access for every individual app.

SSO grew to prominence in an era of static applications, managed devices, and perimeter-based trust. However, despite SSO's adoption, companies continue to struggle with untrusted forms of access and an overall widening of the Access-Trust Gap, which SSO cannot stop.

Companies of every size are grappling with how to use SSO most effectively – to take advantage of its strengths while accounting for its weaknesses and costs. SSO solutions still play a role, but on their own, they do not adequately address the security risks of the modern workforce.

This ebook explores how today's work environment has outpaced traditional access management and why organizations need to rethink where and how SSO fits into their security strategy.

SSO leaves many applications unmanaged

Companies implement SSO to secure and manage access to applications that interface with critical corporate data. Unfortunately, there are many applications that SSO solutions are unable or unsuited to secure.

The SSO tax puts security out of reach

The greatest obstacle to SSO's efficacy is its cost. SSO solutions require a subscription cost – usually as part of an IdP's package – but this is hardly the primary concern. SSO's chief expense comes from SaaS vendors charging vastly higher rates for the ability to secure their applications behind SSO.

Vendors often force customers who want SSO to upgrade to an "enterprise tier," which can cost exponentially more per user than a basic tier without SSO. This practice even has an unofficial nickname: the SSO tax.[1]

Naturally, it's not unreasonable for vendors to charge more for added features. However, the SSO tax can make apps prohibitively expensive. Hubspot, for instance, charges a 7,828% price increase for plans with SSO, and this is far from an aberration.[2]

This practice is particularly burdensome for smaller companies with smaller security budgets, that are also less likely to benefit from the other "enterprise" features that often come bundled with SSO functionality.

CISA recently published a report detailing the barriers to SSO adoption for small and medium-sized businesses. In it, they bluntly argue that "There is an inherent incentive to convince SMBs to adopt technologies at the level of service that may not necessarily benefit the SMBs."[3]

CISA further points out, "In addition to a higher cost per user, this premium pricing model typically requires a minimum number of users." Depending on a company's size, they may be unable to secure a given application behind SSO.

The average company has hundreds of applications, and the end result of the SSO tax is that it's simply not realistic for teams to secure every one of those applications behind SSO.

For enterprises, this may mean locking only their most critical applications behind SSO and finding supplementary solutions to secure and manage their other apps. For SMBs and midmarket companies, SSO often functions more like a luxury tax than a security solution – too expensive to deploy widely and not dynamic enough to protect their most-used apps.

For SMBs and midmarket companies, SSO often functions more like a luxury tax than a security solution.

[1] https://blog.1password.com/explaining-the-backlash-to-the-sso-tax/ [2] https://ssotax.org/
[3] https://www.cisa.gov/sites/default/files/2024-06/Barriers-to-SSO-Adoption-for-SMB-508c.pdf

Many applications are not supported by SSO

SSO operates as a go-between across four parties:

1 **The user** authenticates to their company's IdP – hopefully through strong, phishing-resistant authentication.

2 **The IdP –** often the SSO provider itself – issues a signed authentication token (e.g., SAML or OIDC) that the application uses to verify the user.

3 **The SSO** provider encodes that verified user identity into an authentication token.

4 **The service provider** (the third-party cloud application) uses the token to authenticate the user.

SSO enables users to authenticate to multiple applications through a single token, which is issued by their IDP. This is an essential outline of how SSO works, though there's some variation across different solutions.

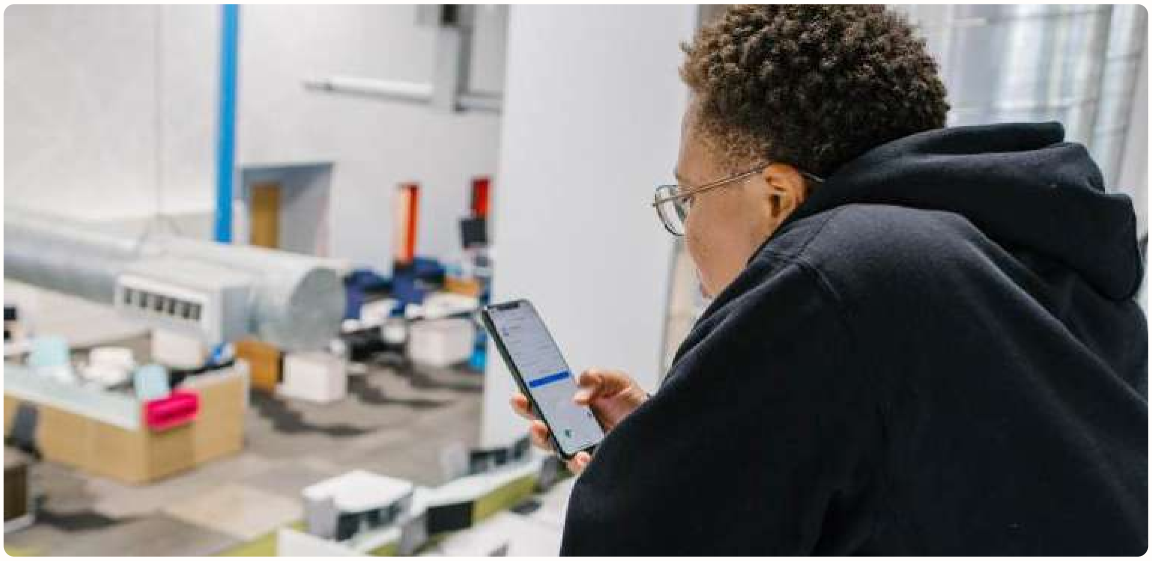
Regardless, for this process to succeed, your SSO has to be able to interface with every third-party application that your company wants to secure behind it. Unfortunately, there are many different ways that this communication can fail.

For instance, authentication tokens can be encoded through various security protocols and standards. SAML and OpenID Connect (OIDC) are two commonly used protocols today. If your SSO provider generates OpenID tokens, but an application can only read SAML tokens, the SSO can't secure that application. There are ways to bridge SSO across different frameworks, but they'll add complexity and cost.

That's just one example of how any application might not work with a specific SSO provider. Chetan Honnenahalli, engineering lead at HubSpot, explains that even in cases where the SSO and the application both operate through SAML, "…different vendors employ different methods for signing the SAMLResponse token…. If the signature expectations of the Service Provider and Identity Provider don't align, SSO transactions can fail."[4]

And, of course, SSO can't support legacy applications that predate modern authentication standards – at least, not without a lot of work being put into integrating them.

[4] https://www.cyberdefensemagazine.com/deep-dive-unveiling-the-untold-challenges-of-single-sign-on-sso-management/



35% of breaches involve data stored or shared through unauthorized applications or cloud buckets.[6]

SSO can't secure shadow IT or shadow AI

Companies frequently deploy additional security tools to manage access to any of their apps that SSO can't secure – whether due to cost or integration issues. But even in an ideal world where every approved application could be secured behind it, SSO still has no ability to discover or manage unapproved apps, shadow AI, and shadow IT.

These unapproved shadow IT applications present unacceptable security risks. In a 1Password survey, more than a third of workers admitted to using unapproved apps or tools during work.[5] Meanwhile, IBM reports that 35% of breaches involve data stored or shared through unauthorized applications or cloud buckets. Furthermore, IBM reports that breaches involving shadow IT last longer and cost more for the impacted company.[6]

Modern corporations also need to contend with the evolving risks posed by AI tools. Without proper governance, even approved AI applications can become a security liability. Teams need to ensure that these applications are being used correctly – for instance, that users are logged in through their secure enterprise ChatGPT account, rather than their personal one.[7] Unfortunately, SSO has no insight into how applications are used at a granular level.

SSO's inability to discover shadow IT can lead to data breaches, fines for non-compliance, and overspending on redundant or unauthorized SaaS tools.

In summary, there are various reasons why an SSO solution might not support a given SaaS application, from cost to technological dependencies. Teams will likely have to put a great deal of effort into exploring the complexities of their intended app integrations and have a separate solution for shadow IT discovery.

[5] https://blog.1password.com/productivity-ai-cybersecurity-report/. [6] https://www.ibm.com/downloads/documents/us-en/107a02e94948f4ec [7] https://blog.1password.com/two-checks-chatgpt-macos-app/

SSO does not manage access for every user

Companies often assume that if SSO manages an app, it can't be accessed any other way. Unfortunately, this is far from true, and many users can still access applications outside of SSO's oversight.

Superadmins have alternate means of accessing applications

In 2022, SSO vendor OneLogin suffered an outage of about four hours. During that time, most users could not connect to the SSO portal. As Sumeet Wadhwani reported for Spiceworks, "The outage was profound because it barred employees from logging in and accessing applications, some of which could be business-critical."[8]

SSO solutions, like any software, can suffer outages or interruptions to their service. However, since SSO is the single point of access guarding company apps, it can also be the single point of failure. Any disruption to it can represent a considerable risk to company productivity.

As such, most companies with SSO will also have superadmin and "break-glass" accounts. These users can typically access critical applications outside the SSO through more standard login flows. This is a necessary safeguard to enable some access to those critical applications in emergencies. The issue is that superadmin users may be authenticating through notoriously insecure methods like passwords, which are vulnerable to compromise. And if a superadmin needs to be offboarded, then the offboarding flow afforded by SSO will not work.

Superadmins will likely have a great deal of access to sensitive data within company applications. It's imperative that those accounts be managed and that IT and security can oversee who needs that kind of access.

> "
>
> The outage was profound because it barred employees from logging in and accessing applications, some of which could be business-critical.[8]

Partner or contractor accounts can't always be added to SSO

Users outside the company – like external partners or independent contractors – cannot always be easily added or offboarded from SSO.

For one thing, SSO solutions are often tied to companies' specific email domains. Email accounts outside that domain can't be verified or provisioned through the SSO.

[8] https://www.spiceworks.com/tech/tech-general/news/onelogin-single-sign-on-outage/

Issues like this are common across SSO and IdP solutions. Some vendors allow for the creation of guest users and accounts, but these can pose their own risks, with one report finding that “over 37% of Azure accounts have at least one overly permissive guest user.”[9]

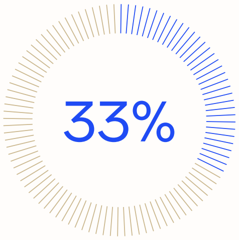
Provisioning SSO access can also be overly complex for managing contractor identities, who may need to be able to enter and leave an organization reasonably quickly. WorkOS advises, “For cases where external contractors can not be added to the organization’s IdP, an MFA requirement can be set as an enforcement fallback. This is sometimes a reasonable compromise if many contractors rotate in and out of the organization.”[10]

However, this compromise invites risk, as evidenced by the 2022 Uber hack, where the attacker was able to bypass the MFA safeguards on a set of stolen contractor credentials. From there, they could leverage that access to steal administrator credentials with full access to Uber’s cloud environments.

SSO doesn’t see legacy users

When a company rolls out SSO, IT and security admins can typically set authentication policies requiring that users can only access certain apps through the SSO. Unfortunately, SSO doesn’t oversee legacy user accounts – especially those created before SSO enforcement or those tied to unintegrated systems. These accounts often live on in overlooked SaaS tools, unmanaged cloud environments, or shadow IT that wasn’t included in the initial rollout.

There may also still be active accounts from users who have already left the company before SSO adoption, whose accounts are unlikely to be included in any enforcement policies. These accounts rarely trigger alerts, fall outside SSO’s governance policies, and are easily missed in traditional audits, yet they often retain access to sensitive data. The result is a growing surface of unseen, unmanaged risk. Addressing it requires more than federation; it demands a shift toward continuous access visibility, app discovery, and lifecycle oversight – capabilities that go beyond what SSO was ever designed to deliver.



33% of enterprise apps will integrate agentic AI by 2028, with 15% of daily work decisions made autonomously.[11]

SSO can’t secure non-human identities

According to Gartner, 33% of enterprise apps will integrate agentic AI by 2028, with 15% of daily work decisions made autonomously.[11] Indeed, agentic AI represents a new generation of AI-powered software. Autonomous systems not only generate outputs but also make decisions and independently interact with enterprise systems.

A real-world example is the increasing use of AI-driven financial automation in enterprises. An AI-powered expense management system may need secure and controlled access to corporate banking data, payroll records, and approval workflows.

[9] https://orca.security/resources/blog/detect-guest-user-account-exploited/ [10] https://workos.com/docs/user-management/sso-with-contractors/adding-an-authentication-policy [11] https://www.gartner.com/en/articles/intelligent-agent-in-ai

Provisioning that access poses many complications. SSO was built for human users on trusted devices, operating within well-defined network perimeters. But agentic AI does not operate like a human, and it can't be secured like one. These agents don't log in through browsers, can't use FIDO2 keys or biometric MFA, and don't support SAML flows. In practice, developers often hardcode secrets into scripts, disable MFA, or borrow employee credentials just to make the agents function. All of this creates massive blind spots and compliance risks that SSO can't prevent or even see.

Just as unsanctioned SaaS tools gave rise to shadow IT, the rise of agentic AI is driving a surge in shadow AI. AI agents operate continuously outside IT's visibility, often with access to sensitive data and systems. Without centralized identity-aware credential governance, they access systems using shared credentials and create data exposure risk.

Security risks tied to agentic AI:

1. **AI agents connecting to Salesforce or payroll systems using shared tokens stored in plaintext.**
2. **Agents reading entire databases to complete a single task – violating least-privilege principles.**
3. **Companies unable to revoke access to agents that are no longer needed or that were built by departing employees.**
4. **Audit logs that can't distinguish between human and AI actions – making breach investigations nearly impossible.**

Bad actors can steal SSO session tokens

SAML tokens are certainly more challenging to steal than standard passwords. However, one token can also represent the keys to the kingdom, as it provides access to multiple company applications. That makes them extremely valuable targets to bad actors, who have found various ways of compromising SSO tokens.

Tokens are typically stored as cookies in users' browsers, and therefore can be stolen through methods like adversary-in-the-middle and pass-the-cookie attacks.

As CSO John A. Smith points out, employees' personal devices pose a particular risk when allowed to access corporate SaaS apps. “Personal devices also have browser caches but do not have to pass the security rigor of corporate systems. They are more easily compromised by threat actors who can capture tokens directly from poorly secured personal devices.”[12]

Another method of token theft is through compromising the IdP or SSO providers themselves. For example, a 2023 Okta breach resulted in the exposure of their customers’ session tokens.[13]

In an even more harrowing 2023 incident, bad actors stole a Microsoft signing key used to validate authentication tokens. As the U.S. Cyber Safety Review Board reported, “...when combined with another flaw in Microsoft’s authentication system, the key permitted... full access to essentially any Exchange Online account anywhere in the world.”[14] The bad actors were able to compromise the email accounts of high-ranking government officials in multiple countries.

In summary, SSO is not a silver bullet for preventing unauthorized application access. Many users fall outside its management, and there are various ways that bad actors can compromise SSO authentication.

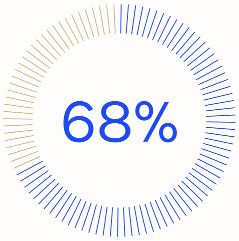
[12] https://www.darkreading.com/cyberattacks-data-breaches/why-tokens-are-like-gold-for-opportunistic-threat-actors [13] https://www.darkreading.com/application-security/more-okta-customers-hacked-through-support-service [14] https://www.cisa.gov/sites/default/files/2025-03/CSRBReviewOfTheSummer2023MEOIntrusion508.pdf

SSO doesn’t eliminate password risk, it can obscure it

For many organizations, SSO is not accelerating passwordless adoption but is actually slowing it down. That’s because SSO relies on each application supporting federated login standards like SAML or OIDC, and passkey adoption in those apps often requires significant integration work. Even when passkeys are supported, they’re usually tied to the SSO provider itself, making it hard to enforce consistent passwordless experiences across the entire application ecosystem. Organizations find themselves in a fragmented environment where only a subset of users and tools benefit from stronger authentication, while the rest remain stuck in the password era.

While SSO can centralize access control for federated applications, most deployments still rely on passwords behind the scenes. SSO does nothing to eliminate password-based authentication for third-party SaaS apps, legacy systems, or shadow IT – where password reuse, shared credentials, and insecure sign-ins are still the norm.

In fact, password practices often deteriorate under SSO because users assume SSO has it covered. In reality, poor password hygiene, shared credentials, and exposed secrets still put companies at risk – with 68% of breaches involving credential compromise.[15]



In reality, poor password hygiene, shared credentials, and exposed secrets still put companies at risk – with 68% of breaches involving credential compromise.[15]

In summary, SSO is not a silver bullet for preventing unauthorized application access. Many users fall outside its management, and there are various ways that bad actors can compromise SSO authentication.

[15] https://www.verizon.com/business/resources/reports/2024-dbir-executive-summary.pdf

All SSO security is not created equally

SSO solutions can leave many applications and users unsecured. But even when SSO protects something, questions may remain about how well it is being secured.

SSO offers flat security

SSO offers some centralized access controls but lacks crucial insight into the context of access at the individual application level.

In some ways, SSO replicates the shortcomings of the old “castle and moat” version of network security. It offers a protective layer of security around access. However, once a user has crossed that layer, SSO does very little to reason about what happens within its barrier.

Jason Meller
VP of Product at 1Password

“

SSO is a pretty flat form of security, and it’s missing a lot of the details that let admins make responsible decisions about managing access. It shouldn’t be a yes/no binary where SSO grants everyone the same level of access – you need to understand what someone will be doing in an app. When was the last time they used it? Why are they assigned this role? Who granted them permission? Signing in with SSO and centralizing permissions has some benefits, but they're minimal compared to this larger access problem.

Even for managed identities, SSO doesn't have much ability to reason about who is accessing applications and why. For instance, SSO can show certain log events, like when users last accessed an application.

But to find this information, the admin will typically need to have specific users or apps in mind to monitor. There are few systems within SSO that can alert when, for instance, a particular user hasn't logged in for an excessive amount of time, which can be invaluable for tracking improperly offboarded users and detecting unused licenses.

SSO can also be used to provide access to apps only as needed, according to a user's group or level. However, within the apps themselves, SSO has no real oversight over what access that user has or how they use it. Furthermore, most SSO providers have limited ability to tailor access based on context like device health or app sensitivity.

For instance, admins may want to allow developers to access Slack through any personal device but want to require that GitHub is only accessed from the user's secure, managed device. SSO cannot allow that level of nuance.

Authentication protocols vary in their security

> “
>
> SAML integration can be difficult to implement and check… if every layer isn't thoroughly checked, malicious actors can misuse the payload.” [16]

As previously mentioned, a few authentication standards are in use today, and each one offers a different level of security. Unfortunately, calculating the relative security provided by any given option can depend on many factors.

In comparing SAML and OIDC, for instance, teams must consider various factors. SAML has been a long-held standard of SSO, meaning it's built more directly into the infrastructure of web applications. It's also broadly considered the more stringent security standard for enterprise environments.

However, SAML is also highly complex and difficult to manage. As Michael Grinich and Zeno Rocha of WorkOS explained, “SAML integration can be difficult to implement and check… if every layer isn't thoroughly checked, malicious actors can misuse the payload.”[16] Teams without ample time and resources may have trouble configuring it properly, which could easily offset any security gains.

OIDC is lighter weight, simpler to manage, and better suited to mobile authentication. However, it also works by adding an authentication layer to the OAuth 2.0 authentication protocol. OIDC offers flexibility and is better suited to mobile and modern apps, but it can inherit OAuth 2.0's complexities or vulnerabilities if not implemented carefully.

These are only examples of some of the complexities of comparing SSO standards. Overall, teams have to consider many factors when deciding what protocol will offer the best security gains.

[16] https://stackoverflow.blog/2022/09/12/the-many-problems-with-implementing-single-sign-on/

SSO can be difficult to manage

Regardless of the protocol, SSO's overall security depends heavily on the size, skillset, and capacity of the team maintaining it. Even basic tasks – like integrating apps, managing token settings, or troubleshooting failed logins – require specialized knowledge and ongoing oversight.

CISA explicitly names this complexity as a barrier to SSO adoption for SMBs and commercial companies, where teams are often smaller and less specialized. CISA writes, "setting up the advanced SSO features often requires specialized technical knowledge and expertise as well as a time commitment."[17]

In other words, SSO is often only as effective as the team behind it. In many organizations, that team is already stretched thin.

This results in a troubling dynamic. The companies most in need of centralized access control often lack the resources to implement it securely. When SSO is misconfigured, it can create a false sense of protection, leaving unmanaged access hidden behind a security façade.

SSO requires strong MFA

Though SSO is considered compatible with a Zero Trust security architecture, it remains true that if the SSO is compromised in any way, multiple applications are put at risk.

SSO is only as effective as the team behind it.

To mitigate that risk, SSO needs to be protected by strong authentication.

However, much like SSO, not all MFA solutions are created equally. Email and SMS MFA are notoriously insecure, for instance. Meanwhile, the previously mentioned Uber hack provides one example of how MFA prompt fatigue – combined with a bit of phishing – could breach systems with MFA enabled.

Companies hope to use SSO to enable passwordless authentication. Passwordless authentication is critical to security, using phishing-resistant authentication factors like biometrics or FIDO2 keys.

Unfortunately, AI agents and bots will be unable to use many passwordless factors – with biometrics being one obvious example. Teams will need some way of securely provisioning them with credentials and MFA codes. Meanwhile, FIDO2 and WebAuthn – the cryptographic standards behind most passwordless factors – both operate independently of SSO. This means they can suffer from many integration issues plaguing SSO solutions. Okta, for instance, provides a list of cases where WebAuthn may be limited or unsupported.[18]

[17] https://www.cisa.gov/sites/default/files/2024-06/Barriers-to-SSO-Adoption-for-SMB-508c.pdf

[18] https://support.okta.com/help/s/article/webauthn-limitations?language=en_US

Session timeouts and limitations are a critical component of SSO security.

Individual applications have varying security within SSO

It falls upon third-party SaaS vendors to design and implement their app's integration with SSO. This means that many of SSO's security abilities depend on the application providers, who have been known to misconfigure various settings when designing SSO implementation.

In an article for *CSO Online*, Joe Sullivan and Atul Tulshiibagwale discuss some common misconfigurations in SSO integrations. Session limitations are one particularly concerning example. "While SSO providers can set restrictions on token usage, it ultimately depends on the application provider to implement and enforce these limitations… many applications incorrectly configure session tokens by failing to enforce expiration, properly validate authentication tokens, terminate sessions when users log out, or implement or validate binding."[19]

This came into play in the aforementioned 2023 Okta breach that resulted in losing their customers' session tokens. As CSO John A. Smith reported on *Dark Reading*, "…on Nov. 23, 2023, Cloudflare detected a threat actor targeting its systems using session tokens from the Okta breach. This indicates that these session tokens were not expired a full 30 to 60 days following the Okta breach – not as a routine course of business, and not in reaction to the breach itself."[20]

Session timeouts and limitations are a critical component of SSO security. If these are misconfigured, they can put a company's security at risk while also jeopardizing its ability to comply with regulations like HIPAA, PCI DSS, and others.

In summary, SSO's security capabilities depend highly on a team's size and ability to manage its complexities. Furthermore, security varies depending on the protocols used and the individual SSO integration designed by third-party SaaS vendors.

[19] https://www.csoonline.com/article/1248700/the-sso-tax-is-killing-trust-in-the-security-industry.html

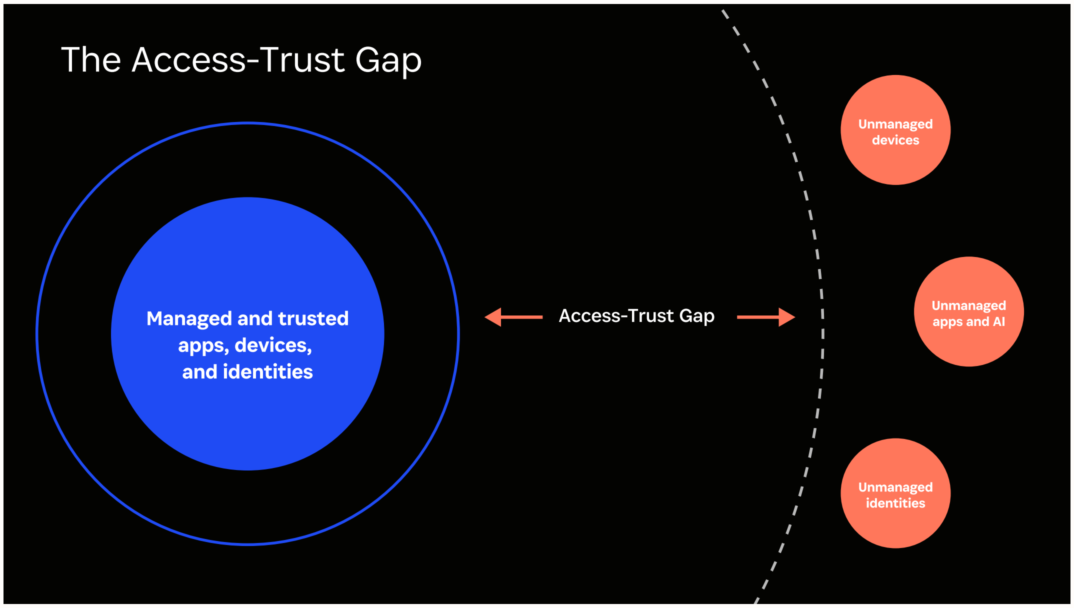
[20] https://www.darkreading.com/cyberattacks-data-breaches/why-tokens-are-like-gold-for-opportunistic-threat-actors

SSO supplements and alternatives

For companies that already have SSO, there's little reason to retire it. When properly configured, SSO can bring many security benefits to corporate environments.

Even so, companies with SSO need to consider supplementary solutions to secure the various applications and users it leaves unmanaged. Companies without SSO will benefit from exploring alternative solutions that better serve their needs until they can afford the investment of budget and technical expertise.

A new category of security solutions, Extended Access Management (XAM) is designed from the ground up to close the Access-Trust Gap. It complements tools like SSO but goes far beyond their limitations.



Application discovery and management

It's been established that shadow IT and app sprawl are critical IT and security concerns, and SSO cannot discover and manage them.

However, containing their spread across an enterprise is no easy task, and efforts to control shadow IT often only drive it further into the shadows. Some companies, for example, try to route all work through enterprise browsers, virtual desktops, or other technologies that don't allow for the use of unapproved tools. Not only do workers often find workarounds, but this effectively inhibits productivity.

To avoid this scenario, containing app sprawl and shadow IT will require teams to follow these best practices:

1. Discover and inventory: Teams need to have a complete picture of all work-related SaaS apps – both managed and unmanaged – in use across their organization.
2. Understand user engagement: IT and security need avenues to communicate with employees about the value they're getting from unsanctioned apps and either bring those apps under management or present alternatives.
3. Identify application risks: Teams must have a reasonable understanding of the risks posed by different apps so they can manage them accordingly.

Trelica by 1Password is a SaaS management platform that enables teams to discover all of the SaaS apps that employees use, both sanctioned and unsanctioned. The solution achieves continuous app discovery through Identity Provider logs (OAuth/OIDC, SAML, SWA), finance systems, a browser extension, and 350+ direct integrations with leading SaaS vendors.

Trelica by 1Password provides teams with an inventory of all apps employees use for work, whether the app is downloaded to their computer or accessed via browser. It shows the number of licenses and how often individual employees use applications. It also provides pre-populated app profiles that grade risks and compliance issues according to the permissions granted to each app.

This provides a simple method for reducing app sprawl and removing unused or risky licenses without locking down employees and harming productivity. Furthermore, these features enable spend management and optimization on SaaS licenses. Teams can reduce unnecessary costs by analyzing app usage data to identify and eliminate unused licenses and redundant apps. They can even leverage integrations from procurement tools and AI models to extract metadata from PDF contracts in order to identify active usage against paid licenses.

Lifecycle management

It's critical for IT teams to give users and AI agents the level of access they need, when they need it, and to revoke access when it is no longer appropriate. This requires regular audits of user permissions and visibility into all access.

The 1Password Extended Access Management platform secures every sign-in to every application from any device, which enables it to manage access to applications and identities that are invisible to SSO.

For instance, Trelica by 1Password provides visibility to all forms of authentication – including superadmins and legacy accounts – and provides a truly centralized way to grant and revoke access.

1Password Extended Access Management also allows teams to enforce security policies requiring that employees register any AI agents, along with policies to distinguish between human and non-human access.

Continuous trust verification

Traditional SSO is built on a single binary event: you log in once, and then you're trusted. But in today's environment, that one-time trust model breaks down. Users switch devices, connect from personal endpoints, and use AI agents that never "log in" in a traditional sense.

Trust can no longer be a single event. It must be continuously earned. That's why 1Password Extended Access Management replaces static login-based trust with real-time context-rich verification of user and device status, ensuring that sensitive resources are only being accessed by devices and users that are currently trusted to do so. This helps to manage the risks posed by legacy users and compromised credentials.

SSO does not enable a path to passwordless

SSO adoption often gets mistaken with passwordless maturity. In reality, the two are not synonymous. It's critical to demand strong authentication factors for every app at a company whether that app is secured behind SSO or outside of its protection. In either case, there's an undeniable security mandate to phase out passwords in favor of more phishing-resistant forms of authentication.

A true transition to passwordless security requires more than SSO federation. It demands visibility into where passwords are still used, prioritization of credential risks, and structured support to help employees and IT move to stronger, phishing-resistant authentication methods.

1Password Extended Access Management secures every sign in to every application from every device.

For example, 1Password Device Trust also provides a phishing-resistant authentication factor in the form of device identity. Users can't access applications from any device that isn't trusted and associated with that user. This means that bad actors, even with phished credentials, won't be able to authenticate to systems as long as they're using an unknown device – offering a means beyond SSO to govern access to critical applications.

1Password Extended Access Management also enables your team to see where passwords are still being used at your company. It then uses the 1Password Enterprise Password Manager (EPM) to facilitate the transition from passwords to passwordless authentication factors.

EPM can, therefore, secure authentication for apps where SSO is too expensive or incompatible. 1Password's EPM also allows developers and administrators to securely provision the necessary credentials to AI apps and agents.

Unlike SSO, which is limited to apps that support modern identity provider infrastructure, 1Password brings passwordless capabilities to every application and identity, including those that SSO can't reach. By helping organizations identify where password risks still exist and providing the tools to replace them, Extended Access Management paves the way to a future where mature passwordless security is achievable.

Build security into applications

Solving the issues of secure authentication to SaaS apps also requires looking more closely at the applications themselves. It's necessary to ensure that application developers have the tools they need to build secure access controls into their offerings.

For example, as AI agents become increasingly embedded into enterprise workflows, developers need a way to securely manage credentials, secrets, and private information within each application.

The 1Password Developer SDK provides a solution for AI developers to seamlessly integrate credential storage, authentication, and access management into their apps. Through secure vaults and end-to-end encryption, the 1Password SDK ensures that AI agents only retrieve the credentials and secrets they are explicitly authorized to use. This enables app developers to ensure that their applications can only access sensitive information within a secure, policy-driven environment.

SSO won’t suit every team

SSO best served its purpose when work was centralized. But in a decentralized world of SaaS, AI, and unmanaged personal devices, it can no longer stand alone. Extended Access Management closes the gaps that SSO leaves behind – because every app matters, every identity counts, and every sign-in must be secure.

All the applications and identities that SSO can’t protect (or doesn’t protect adequately) fall into the Access-Trust Gap. The Access-Trust Gap refers to the security risks posed by unmanaged devices, applications, and AI-powered tools accessing company data without proper governance controls. That doesn’t mean these tools lack value; they are merely insufficient and require supplementary solutions.

> “The bottom line is that SSO is no less secure than an infrastructure without it, and is almost always more so.”[21]
>
> –Josh Fruhlinger, CSO Online

The above quote is accurate to a degree. The issue is that this isn’t a compelling enough security promise when weighed against SSO’s high cost and complexity – particularly for smaller organizations.

The Access-Trust Gap refers to the security risks posed by unmanaged devices, applications, and AI agents accessing company data without proper governance controls.

Companies of any size need to be aware of SSO’s limitations and costs when they choose when and how to invest in it. Every company should consider alternative or complementary solutions to address SSO’s shortcomings and close the Access-Trust Gap.

Want to learn more about how 1Password Extended Access Management can enhance or replace traditional security solutions? Reach out for a demo!

©2025 AgileBits Inc. All Rights reserved. 1PASSWORD, the Key Hole Logo and other trademarks owned by AgileBits Inc.

[21] https://www.csoonline.com/article/510713/sso-explained-single-sign-on-definition-examples-and-terminology.html