International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024

SSO - A Secure Login Flow Using SAML

[1]Amandeep Kaur, [2]Amjan Shaik

[1]Assistant Professor, Dept of CSE, IIMT Greater Noida,India, [1]amandeepk911@gmail.com

[2] Professor of CSE and Dean R&D, St.Peter's Engineering College, Maisammaguda, Hyderabad, Telangana, India, [2]amjansrs@gmail.com

Abstract- Single sign on technology (SSO) is a boon for the users and the enterprise as users need to remember only one set of credentials to login into different applications/websites/service providers. In this paper the overview and working of the SSO protocol SAML i.e Secure Assertion markup language has been discussed.

Keywords- Single Sign On (SSO), SAM, Identity Provider, Service Provider, Security, Authentication, Authorization

I. INTRODUCTION

SSO simplifies the user experience by reducing the need for users to remember and enter multiple user names and passwords. It is hard to manage multiple user name and passwords and it weakens the security also..[1] This leads to password fatigue. [7]Single sign-on (SSO) allows a person to authenticate once at their home domain to obtain a "token", which is stored in the browser (cookie) or mobile device, and can be presented to websites as evidence of authentication .

Fig1: A single sign On Process

II. KEY FEATURES

Key features and concepts of SSO include:

1. **User Authentication:** Users authenticate themselves once, usually at the beginning of a session, by providing their credentials to an Identity Provider (IdP). The IdP verifies their identity.
2. **Session Management:** Once authenticated, the IdP generates a session token or assertion, which is used to indicate that the user is authenticated. This token is securely shared with the Service Providers (SPs) for access to specific resources.
3. **Service Providers (SPs):** SPs are applications or services that trust the IdP's assertions and rely on it for user authentication and authorization.
4. **Federated Identity:** SSO can be used in federated identity scenarios where multiple organizations or service providers trust a common IdP for user authentication. It enables seamless access to resources across different security domains.
5. **Security and Access Control:** SSO helps centralize and strengthen security measures. Users and administrators can manage access permissions more effectively through the IdP.
6. **Single Sign-Off (SLO):** SSO often includes Single Sign-Off, which allows users to log out from one application, triggering a logout from all other SSO-enabled applications (if supported).

III. BACKGROUND

The history of Single Sign-On (SSO) dates back several decades and has evolved alongside advancements in computer networking and security.

1. **Early Days of Authentication (Pre-Internet Era):** Before the widespread use of the internet, computers and networks typically had their own authentication systems. Users had to log in separately to each system or application.
2. **Kerberos Authentication (1980s):** Developed at MIT in the 1980s, the Kerberos protocol introduced the concept of centralized authentication.
3. **Web SSO Protocols (Late 1990s):** As the World Wide Web gained popularity, the need for a more streamlined authentication process for web-based applications became apparent. Technologies like HTTP cookies and HTTP basic/digest authentication provided some early solutions, but they were not truly SSO.
4. **Security Assertion Markup Language (SAML) (Early 2000s):** SAML emerged as a significant development in SSO for web applications. It provided a framework for exchanging authentication and authorization data between an Identity Provider (IdP) and Service Providers (SPs). SAML-based SSO allowed users to log in once and access multiple web applications without re-entering credentials.
5. **OpenID (2005):** OpenID was introduced as an open standard for single sign-on on the web. It allowed users to use a single digital identity to log in to multiple websites. OpenID 2.0 became a widely adopted version.
6. **OAuth (2006):** OAuth (Open Authorization) was created as an authorization framework that allowed third-party applications to access a user's data without exposing the user's credentials. While OAuth was not initially designed for authentication, it was later

extended to support authentication use cases, leading to the development of OpenID Connect.

7. **OpenID Connect (2014):** OpenID Connect (OIDC) was introduced as an identity layer on top of OAuth 2.0. It provided a standardized way to implement SSO and identity verification for web and mobile applications. OIDC offered a more user-friendly and secure SSO experience.

IV. SSO PROTOCOLS

Single Sign-On (SSO) is a mechanism that allows users to access multiple applications or services with a single set of credentials after they have initially logged in. There are several protocols and standards that facilitate SSO in different scenarios. Here are some of the most commonly used SSO protocols:

1. **SAML:** SAML is a protocol used for exchanging authentication and authorization data between an Identity Provider (IdP) and a Service Provider (SP).
2. **OpenID:** OpenID is a decentralized SSO protocol that allows users to log in to multiple websites with a single OpenID identifier. It is commonly used for web-based SSO and online authentication.
3. **OpenID Connect:** OpenID Connect is an extension of the OAuth 2.0 protocol. It provides authentication services and is often used for SSO in modern web and mobile applications. It allows for identity information to be passed in a secure and standardized manner.
4. **OAuth 2.0:** While OAuth 2.0 is primarily an authorization framework, it is often used in combination with OpenID Connect to achieve SSO. OAuth 2.0 enables delegated access and can be used for single sign-on use cases when integrated with OpenID Connect.
5. **Kerberos:** Kerberos is a network authentication protocol that is commonly used in Windows environments. It provides strong authentication and is used for SSO in Windows-based networks.
6. **CAS (Central Authentication Service):** CAS is a protocol and framework for single sign-on and secure authentication. It is often used in educational institutions and is technology agnostic, meaning it can be used with various programming languages and platforms.
7. **WS-Federation:** WS-Federation is a protocol that allows different organizations to establish trust and enable SSO across their applications. It is often used in corporate and government settings.
8. **JWT (JSON Web Tokens):** JWT is a compact, self-contained means of representing information between parties. It is often used in token-based authentication and can facilitate SSO when integrated with other protocols.
9. **Shibboleth:** Shibboleth is an open-source project and a SSO protocol used in the education and research sector. It's based on SAML and provides federated access control.
10. **Browser Cookies:** While not a standard protocol, browser cookies are sometimes used for simple SSO. A user's authentication state is stored in a cookie, allowing them to remain authenticated across multiple web applications within the same domain.

The choice of SSO protocol depends on the specific use case and requirements of the organization or application. Different industries and scenarios may favor one protocol over another. Additionally, some modern SSO solutions combine multiple protocols to provide a comprehensive and adaptable authentication and authorization system.

V. SAML

SAML (Security Assertion Markup Language): SAML is one of the earliest and most widely adopted SSO protocols. The SAML specification defines three roles: the Principal (typically a human user), the identity provider (IdP) and the service provider (SP)[6]. It is XML-based and primarily used for web-based SSO. SAML enables the exchange of authentication and authorization data between an IdP and an SP. The user is redirected to the IdP for authentication, and upon successful login, the IdP sends a SAML assertion to the SP to grant access.

A. *Service provider* – A service provider is **an individual or entity that provides services to another party**. A service provider is a federation partner that provides services to the user.

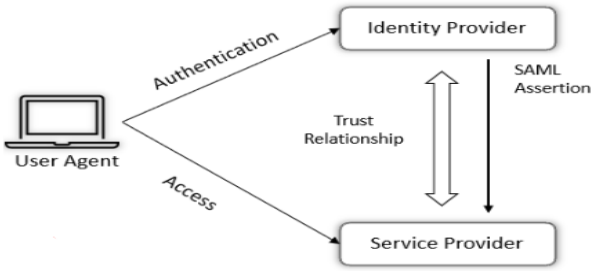

Fig2: A process between a User Agent, Identity Provider and Service Provider

B. *Identity provider* - The Identity Provider authenticates the user and provides an authentication token (that is, information that verifies the authenticity of the user) to the service provider.

C. *User Agent* - A User, basically a browser.

The security of a SAML SSO solution depends on many assumptions (e.g. the trust relationships among the involved parties) and security mechanisms (e.g. the secure transport protocols used to exchange messages).[1]

SAML-based SSO services provide better security and flexibility, as applications do not need to store user credentials on their system.

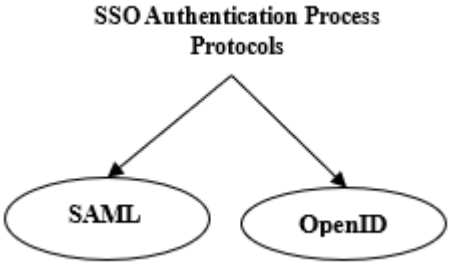

Fig3: SSO common Protocols

D.Authentication Protocol: An authentication protocol is a set of rules that define how the IDP and SP communicate with each other to establish trust and exchange authentication information. Common protocols used in SSO include SAML (Security Assertion Markup Language), OAuth 2 and OpenID Connect

VI. SAML LOGIN WORK FLOW

Here's how SAML works:

1. **User Initiates Access:** The SAML process begins when a user attempts to access a service provided by the Service Provider (SP). This could be a web application, a cloud service, or any resource that requires authentication.
2. **Redirect to Identity Provider (IdP):** The Service Provider (SP) identifies that the user is not authenticated and redirects the user to the Identity Provider (IdP). The SP sends an authentication request to the IdP.
3. **User Authentication:** The Identity Provider (IdP) handles the user's authentication process. The user enters their credentials (e.g., username and password) or uses another authentication method, such as multi-factor authentication (MFA).
4. **SAML Assertion Generation:** After successful authentication, the IdP generates a SAML assertion. A SAML assertion is an XML document that contains information about the user and their authentication status. There are two main types of assertions:
    - **Authentication Assertion (Authentication Statement):** This asserts that a user has been authenticated.
    - **Attribute Assertion (Attribute Statement):** This provides additional information about the user, such as their name, email, or roles.
5. **SAML Assertion Response:** The IdP sends the SAML assertion back to the user's browser as part of the HTTP response. The SAML assertion is typically embedded in an HTML form or sent as a URL parameter.
6. **Assertion Verification:** The SP receives the SAML assertion and verifies its authenticity and validity. This verification includes checking the digital signature of the assertion and ensuring that it came from a trusted IdP.
7. **Access Granted:** If the SAML assertion is valid and the user is authorized to access the requested resource, the SP grants access to the user without requiring the user to log in again. The user is now authenticated and can access the service.
8. **Session Management:** The SP often manages the user's session, allowing them to access additional resources or services without being prompted for credentials again.
9. **Single Sign-Off (SLO):** SAML also supports Single Sign-Off (SLO), which enables users to log out from one application or service, triggering a log-out from all other SSO-enabled applications if they support SLO.

Fig4: SAML Work Flow [6]

1. The process begins when the user (client) attempts to access a service provided by the Service Provider (SP).
2. The SP identifies that the user is not authenticated and redirects the user's browser to the Identity Provider (IdP).
3. The SP sends an Authentication Request to the IdP, initiating the SSO process.
4. The IdP handles the user's authentication and generates a SAML Assertion that contains information about the user's identity and authentication status.
5. The IdP sends an Authentication Response back to the user's browser. This response often includes the SAML Assertion.
6. The user's browser is redirected back to the SP with the Authentication Response, carrying the SAML Assertion.
7. The SP receives the SAML Assertion and verifies its authenticity and validity.
8. If the SAML Assertion is valid and the user is authorized to access the requested resource, the SP grants access to the user without requiring the user to log in again.
9. The user can now access the service, and the SP often manages the user's session.

10. Single Sign-Off (SLO) is a complementary feature that enables users to log out from one application, triggering a log-out from all other SSO-enabled applications if they support SLO.

The Service Provider never directly interacts with the Identity Provider. A browser acts as the agent to carry out all the redirections.[5]

This diagram provides a simplified visual representation of the SAML SSO workflow. In practice, various SAML protocols, bindings, and security measures may be involved, but this illustration captures the core steps in the process.

VII. SSO BENEFITS

Benefits of Single Sign-On include:

- **User Convenience:** Users only need to remember one set of credentials, reducing the risk of password fatigue and the likelihood of using weak or easily guessable passwords.
- **Security:** With SSO, it's easier to implement centralized security policies, enforce strong authentication, and monitor user activity across various applications. It can help reduce the risk of password-related security breaches.
- **Efficiency:** SSO streamlines the login process, making it faster and more convenient for users, which can increase productivity.
- **Simplified Management:** Organizations can manage user access centrally, making it easier to add or remove users, update permissions, and track user activity.

SSO solutions improve user experience by avoiding the interruptions caused by password requests to access essential IT tools.[8]

Common examples of SSO implementations include using services like Google Sign-In, Facebook Login, or Microsoft Azure AD for authentication across various websites and applications. Additionally, organizations often implement SSO for their employees, enabling them to access company resources, such as email, CRM systems, and internal portals, with a single login.

VIII. MODERN SSO SOLUTIONS (PRESENT) AND CHALLENGES:

Today, various SSO solutions are widely used, both in enterprise settings and on the internet. Major companies and organizations have adopted SSO to improve user experience and enhance security. Google provides a single sign-on (SSO) service based on SAML, enabling partner companies to exert comprehensive control over the authorization and authentication processes for user accounts accessing web-based applications like Gmail or Google Calendar. Adhering to the SAML model, Google takes on the role of the service provider, delivering services like Gmail and Start Pages. Meanwhile, Google's partners serve as identity providers, overseeing usernames, passwords, and other pertinent information essential for identifying, authenticating, and authorizing users for the web applications hosted by Google. [2]

Commercial Identity Providers like Microsoft Azure AD, Okta, and Google Identity Platform offer SSO solutions for businesses.

SSO has also evolved in terms of authentication methods. While traditional username and password-based SSO remains prevalent, there is a growing adoption of multi-factor authentication (MFA) and other advanced authentication methods to enhance security.

1. **Challenges and Ongoing Developments:** Despite the benefits of SSO, it is not without its challenges, such as security and privacy concerns. As technology evolves, SSO solutions continue to adapt to meet new challenges, with ongoing developments in the SSO space.While Single Sign-On (SSO) offers significant advantages in terms of user convenience, security, and productivity, there are also challenges associated with its implementation and maintenance. These challenges can vary depending on the specific SSO solution and the organization's requirements. Here are some common challenges

Complex Implementation: Setting up SSO can be complex, especially in large organizations with diverse IT environments. It may require significant changes to existing authentication systems and applications.

Integration Issues: Some legacy systems and applications may not support modern SSO protocols or may require custom integration work, making integration challenging and time-consuming.

User Experience: While SSO is intended to improve user experience, it can lead to frustration if not implemented correctly. Users may experience difficulties in accessing services or understanding how SSO works.

User Adoption: Users may resist SSO if they perceive it as a significant change to their routine or if they have concerns about privacy and data security.

Security Risks: If not properly configured, SSO can introduce security risks. Weak authentication methods, inadequate session management, or misconfigurations can lead to unauthorized access.

Lack of Standardization: There are multiple SSO protocols (e.g., SAML, OAuth, OIDC), and not all applications and services support the same protocols. This lack of standardization can complicate SSO deployments.

Maintenance and Updates: SSO solutions require ongoing maintenance, monitoring, and updates to stay secure and compatible with evolving security threats and standards.

Identity Lifecycle Management: Managing user identities, provisioning, de-provisioning, and managing

permissions across a variety of applications and services can be challenging, especially in larger organizations.

Session Management: Managing SSO sessions, including single sign-off (SLO) and session timeouts, can be complex, leading to security and usability issues if not handled correctly.

Performance and Scalability: SSO solutions can introduce latency if not optimized for performance. Ensuring that SSO systems can handle a growing number of users and services can also be a challenge.

Password Policies: Integrating SSO with existing password policies can be complex, as some users might still need to maintain separate passwords for certain applications.

Regulatory Compliance: Depending on the industry and geographical location, organizations must ensure that their SSO solutions comply with various data protection and privacy regulations.

User Lockout: In some cases, if an SSO solution experiences issues, it can lock users out of multiple services, which can be disruptive to business operations.

Vendor Lock-In: Organizations that rely on a specific SSO vendor may find it challenging to switch providers, potentially leading to vendor lock-in and dependency.

Education and Training: Users and IT staff may require education and training to understand the SSO system fully, which can be time-consuming and costly.

To address these challenges, organizations need to carefully plan and implement their SSO solutions

Applications :

SSO is widely used in various domains, including enterprise environments, cloud-based applications, and educational institutions, to enhance security and user convenience. It simplifies the user experience, reduces the risk of password-related security breaches, and improves overall access control and identity management.

. Here are some common SSO applications and their use cases:

1. **Enterprise SSO:**
   - **Microsoft Azure Active Directory:** Provides SSO for Microsoft 365 and other Microsoft services.
   - **Okta:** Offers SSO, multi-factor authentication, and identity management for enterprise applications.
2. **Cloud Services:**
   - **Google Workspace:** Allows users to access Google services, including Gmail and Google Drive, with a single login.
   - **Salesforce:** Provides SSO for Salesforce and other integrated cloud services.
3. **Identity and Access Management (IAM):**
   - **OneLogin:** Offers SSO, MFA, and identity management for businesses of all sizes.
   - **Ping Identity:** Provides SSO, identity and access management, and API security.
4. **Customer-Facing Applications:**
   - **Login with Facebook/Google:** Many websites and apps offer social logins for user convenience.
   - **Apple Sign-In:** Allows users to sign in to apps and websites using their Apple ID.
5. **Educational Institutions:**
   - **Shibboleth:** Used in the education and research sector for federated identity and access management.
   - **Canvas (Instructure):** Provides SSO for students, teachers, and administrators.
6. **Healthcare:**
   - **Epic (MyChart):** Offers SSO for patient portal access and healthcare information.
   - **Cerner:** Provides SSO solutions for healthcare providers and systems.
7. **Government and Public Services:**
   - **US Government SSO (Login.gov):** Allows citizens to access various government services with a single login.
   - **eIDAS (European Union):** Provides cross-border digital identity and SSO for EU citizens.
8. **Web and Mobile Apps:**
   - **Auth0:** Offers identity as a service, providing SSO and authentication for web and mobile apps.
   - **Firebase Authentication (Google):** Provides SSO for mobile and web apps, often integrated with other Firebase services.
9. **Web Hosting and Content Management Systems (CMS):**
   - **WordPress:** Offers SSO plugins and solutions for websites and blogs.
   - **Drupal:** Provides SSO modules for user authentication and access control.
10. **Financial Services:**
    - **Plaid:** Enables secure access to financial data through SSO, often used by fintech applications.
    - **Fiserv:** Offers SSO solutions for online banking and financial institutions.

These SSO applications cater to various industries, organizations, and user needs. They simplify the login experience, enhance security, and improve user management for both enterprises and consumers. The choice of an SSO application depends on the specific requirements and use cases of the organization or service provider.

IX. SAML DRAWBACKS

Here are some of the common drawbacks of SAML:

1. **Complexity:** SAML can be complex to implement, especially for organizations with a large number of applications and services. Configuring the SAML infrastructure, including Identity Providers (IdPs) and Service Providers (SPs), can require significant technical expertise.
2. **Initial Setup Costs:** Implementing SAML solutions can involve substantial initial costs for software, hardware, and professional services, making it less accessible for smaller organizations with limited budgets.
3. **Interoperability Challenges:** Achieving seamless interoperability between various SAML implementations can be challenging. Differences in SAML profiles, configurations, and vendor-specific extensions can lead to integration difficulties.
4. **IdP Dependency:** SAML-based SSO solutions are heavily reliant on the availability and reliability of the Identity Provider (IdP). If the IdP experiences downtime or technical issues, it can disrupt access to all connected services.
5. **Single Point of Failure:** SAML introduces a single point of failure, as a breach or compromise of the IdP can have far-reaching consequences, potentially exposing access to multiple applications and services.
6. **User Experience:** Users may experience a delay when accessing SAML-protected applications due to the redirection to the IdP for authentication and back to the SP. This can affect the user experience, particularly for applications with high latency.
7. **Session Management:** SAML-based SSO systems require careful session management to maintain security and privacy. Inadequate session management can lead to unauthorized access if a user does not log out properly.
8. **Data Privacy Concerns:** Some users and organizations have privacy concerns related to the exchange of user attributes and information between the IdP and SPs, even though SAML is designed to protect privacy.
9. **Complex Revocation:** Revoking access for a user can be complex in a SAML environment. If access to one application is revoked, other connected services may still grant access, leading to potential security risks.
10. **Legacy System Compatibility:** Some older or legacy systems may not support SAML, requiring custom development or workarounds to enable SSO.
11. **User Training:** Organizations need to provide user training and support to ensure users understand how to use SAML-based SSO and adhere to security best practices.
12. **Complex Identity Management:** Organizations may find managing user identities and attributes in a federated environment challenging, especially when users access services across different organizations or domains.

Despite these drawbacks, SAML remains a widely adopted and trusted protocol for SSO and federated identity management. Organizations considering its implementation should carefully assess their specific requirements, conduct thorough risk assessments, and take steps to mitigate potential challenges while benefiting from its advantages.

X. CONCLUSION

In conclusion, Single Sign-On (SSO) and Security Assertion Markup Language (SAML) play pivotal roles in revolutionizing the way users access digital services, offering a more streamlined, secure, and user-friendly approach to authentication and authorization. Together, they address many of the challenges associated with managing multiple credentials and provide solutions for achieving robust identity and access management.

The authentic Single Sign-On (SSO) system is categorized into distinct types, including cookies-based, agent-based, and gateway-based systems. These categories adhere to fundamental principles. A dedicated SSO Web Authentication Center is established to forge a trust relationship with all business systems. Comprehensive logging occurs exclusively through the SSO authentication center, ensuring the legitimacy of SSO certificates for current access to web applications via various validation methods. [4]

Federated identity systems "are the next logical evolution in authentication and entitlement system." however only after any company has the technology, infrastructure, and processes in place to effectively manage internal resources can your company begin to share and manage identity information with other companies [3].

Together, SSO and SAML provide a myriad of benefits.

However, it is crucial to recognize that implementing and maintaining SSO and SAML solutions come with their own set of challenges. Organizations must carefully plan and execute deployments, address user adoption and security concerns, and ensure that SSO systems remain up-to-date and aligned with evolving industry standards.

Furthermore, future research in SSO and SAML is essential to tackle emerging challenges and opportunities. These may include their role in securing the Internet of Things, integration with Zero Trust security models, enhanced interoperability, and the exploration of innovative privacy-preserving solutions.

In the rapidly evolving digital landscape, SSO and SAML continue to evolve and adapt, staying at the forefront of identity and access management. Their ability to offer secure and user-friendly authentication experiences, coupled with their potential for innovation and growth, underscores their significance in shaping the future of digital identity and access control.

REFERENCES

[1] Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps.
[2]SAMLbased SSO technical Overview https://support.google.com/a/answer/6262987?hl=en
[3] Single Sign-On and Single Log Out in Identity Management
[4] Jian Hu, Qizhi Sun, Hongping Chen Application Of Single Sign-On (Sso) In Digital Campus.
[5]SAMLhttps://developer.okta.com/docs/concepts/saml/#planning-for-saml
[6]https://en.wikipedia.org/wiki/Security_Assertion_Markup_Language
[7] Sun, S. T., Pospisil, E., Muslukhov, I., Dindar, N., Hawkey, K., and Beznosov, K. What Makes Users Refuse Web Single Sign-On? An Empirical Investigation of OpenID. J. Symposium on Usable Privacy and Security (SOUPS), (Jul. 2011), 1-3.
[8] "The Advantages and Disadvantages of Single-Sign-On (SSO) Technology", Secure Connexion, 2012. [Online] https://secureconnexion.wordpress.com/2012/08/24/theadvantages-and-disadvantages-of-single-sign-on-ssotechnology-mini-whitepaper/.
[9] "Does single sign-on (SSO) improve security?" SearchSecurity, 2016. [Online] http://searchsecurity.techtarget.com/answer/Does-singlesign-on-SSO-improve-security.
[10] Davis, M. 2013 "The Pros and Cons Of Single Sign-On for Web Services", Future Hosting, [Online] https://www.futurehosting.com/blog/the-pros-and-consof-single-sign-on-for-web-services/.