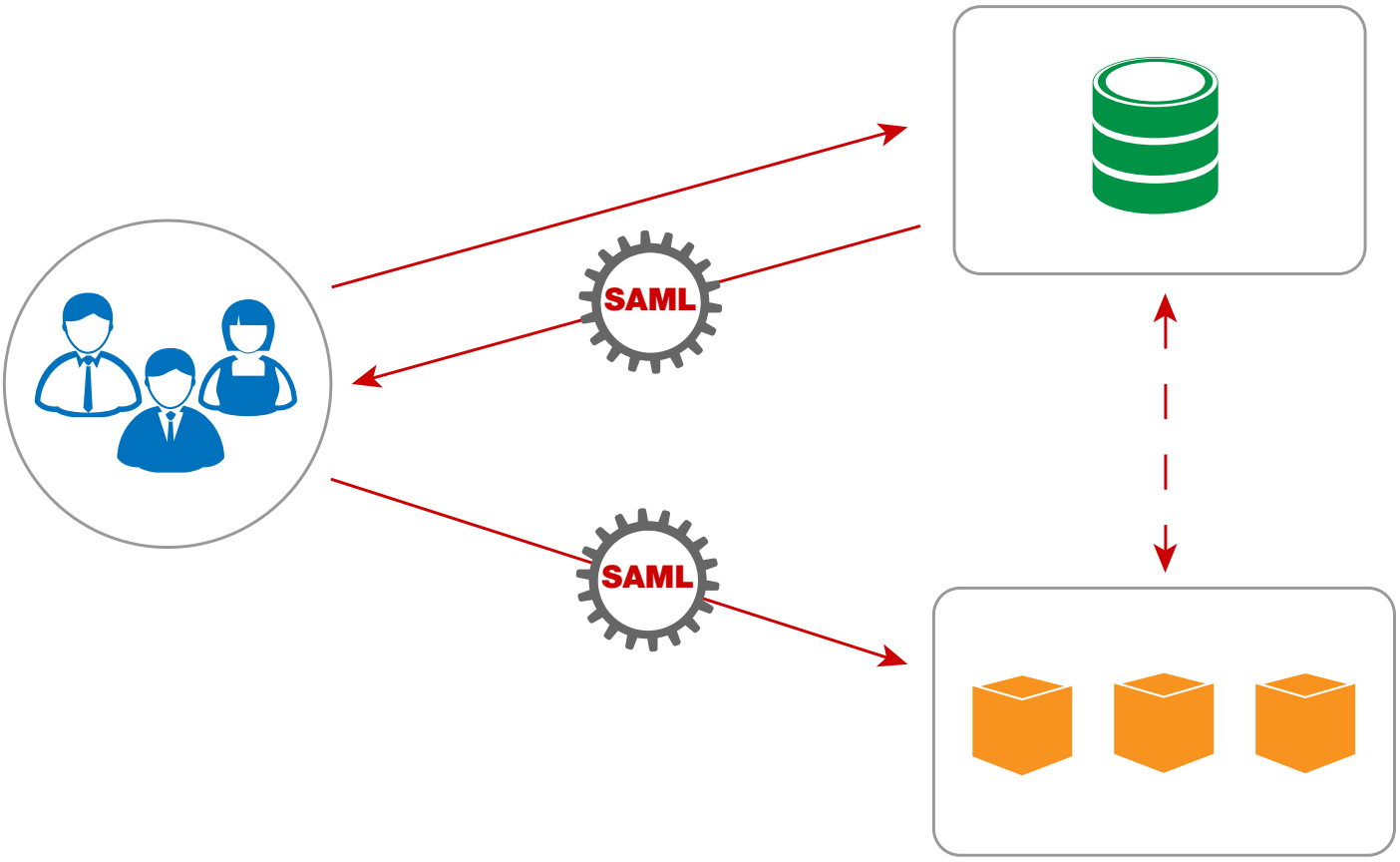
WHITE PAPER

How to Implement Enterprise SAML SSO



FORUMSYSTEMS
THE LEADER IN API AND CLOUD GATEWAY TECHNOLOGY

How to Implement Enterprise SAML SSO

SUMMARY:

In this white paper, you will learn:

- What is SAML and how it's used for Single Sign-On.
- Two methods of implementing SAML SSO.
- The advantages and challenges associated with each implementation strategy.

Introduction

Security Assertion Markup Language, or SAML, provides numerous benefits to enterprises, organizations and governments. One of its greatest assets is Single Sign-On (SSO), the ability to enable users to securely access multiple applications with a single set of credentials, entered once. With SAML, users and organizations can conduct business faster and more efficiently by seamlessly accessing multiple applications on the same domain or on multiple domains.

Many organizations are already utilizing SAML to create a better user experience for their customers and employees. However, there are countless enterprises and organizations who have yet to implement SAML because it can be a difficult and expensive to implement. This white paper will show how SAML SSO works, provide examples of two different methods of implementing SAML and showcase the advantages and disadvantages that are associated with each method.

What is SAML

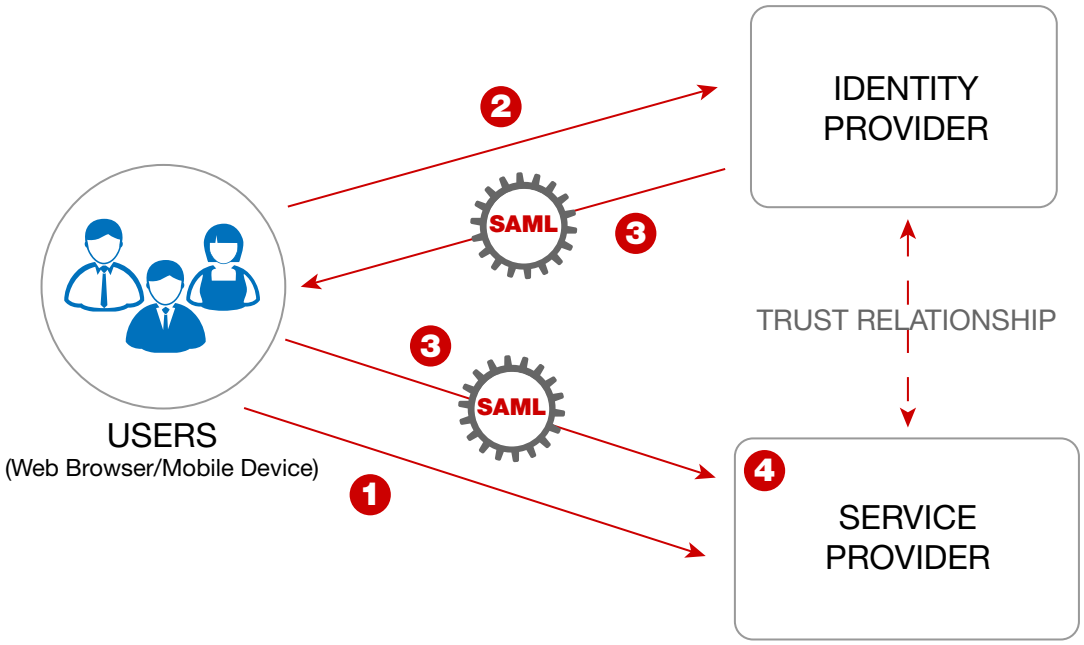
SAML is an XML-based framework used to authorize, authenticate and communicate attributes and privileges of a user. SAML provides the ability to send information about an authenticated user and does not require a third-party system to ever have access to the user's credentials.

To properly understand how SAML SSO works, there are three participants that need to be established: the User, the Identity Provider (IdP) and the Service Provider (SP). The User is the client attempting to access an application. The Service Provider is the owner of the application (may also be referred to as the SAML relying party). The Identity Provider is the holder of record of the user's authentication credentials such as username, password, x509, roles, etc. (may be referred to as the SAML asserting party). As the name SAML (Security Assertion Markup Language) suggests, it is a secure means to assert the identity of a user to another system.

SAML SSO

Now that the participants have been established, let's take a closer look at each of their roles in SSO. There are two primary methods of SAML SSO: Service Provider Initiated (SP-Initiated) and Identity Provider Initiated (IdP-Initiated). SP-Initiated SSO is the more common use case of the two scenarios and it occurs when a user attempts to access the Service Provider directly. Refer Figure 1 on the next page to follow the necessary steps to complete SP-Initiated SAML SSO.

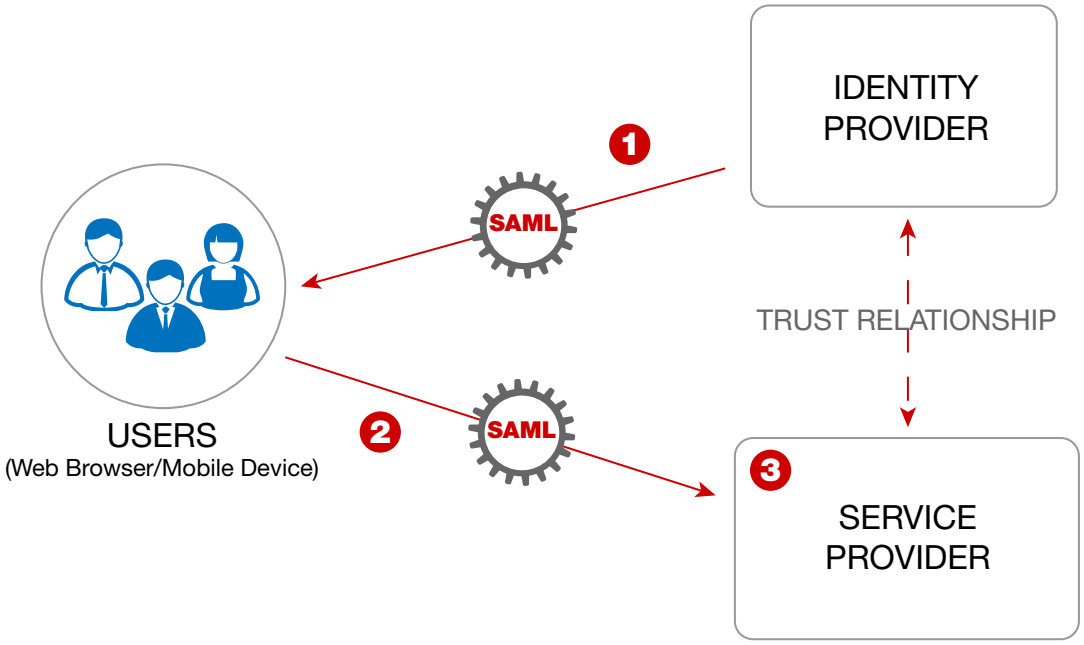
Figure 1: SP-Initiated SAML SSO



1. **The User attempts to access a service from the Service Provider via URL.**
2. **The Service Provider redirects the User's browser to the Identity Provider for authentication.**
3. **The User provides credentials to the Identity Provider. If the credentials are valid, the IdP creates a SAML token and embeds the token along with an HTTP redirect command. The HTTP redirect command sends the User's browser or mobile device back to the Service Provider to present the SAML token. This step has no impact on the client and no code is required since it uses standard HTTP redirect mechanisms.**
4. **The Service Provider inspects the inbound SAML token from the redirect and validates the SAML via the PKI signature check and the SAML contents. If valid, the user is allowed access and a session cookie is established to enable the continued use of the SSO session for some duration of time without requiring re-authentication.**

IdP-Initiated is a less common SSO use case because it requires the User to already be authenticated with the Identity Provider. Typically, this is the case with mobile devices where a User is can be authenticated by accessing their phone. See Figure 2 on the next page to see how IdP-Initiated SSO works.

Figure 2: IdP-Initiated SAML SSO



1. **The User clicks on a link to a partner Service Provider. Since the User has already been authenticated with the Identity Provider, the IdP issues a SAML token to the User.**
2. **The User's browser is redirected to the Service Provider.**
3. **The Service Provider inspects the inbound SAML token from the redirect and validates the SAML via the PKI signature check and the SAML contents. If valid, the user is allowed access and a session cookie is established to enable the continued use of the SSO session for some duration of time without requiring re-authentication.**

The examples above were designed to simplify the two SSO use cases and to do that, many of the technical details associated with each step were omitted. The next section will go into a little more detail on what is required to enable SSO.

Implementing SAML SSO

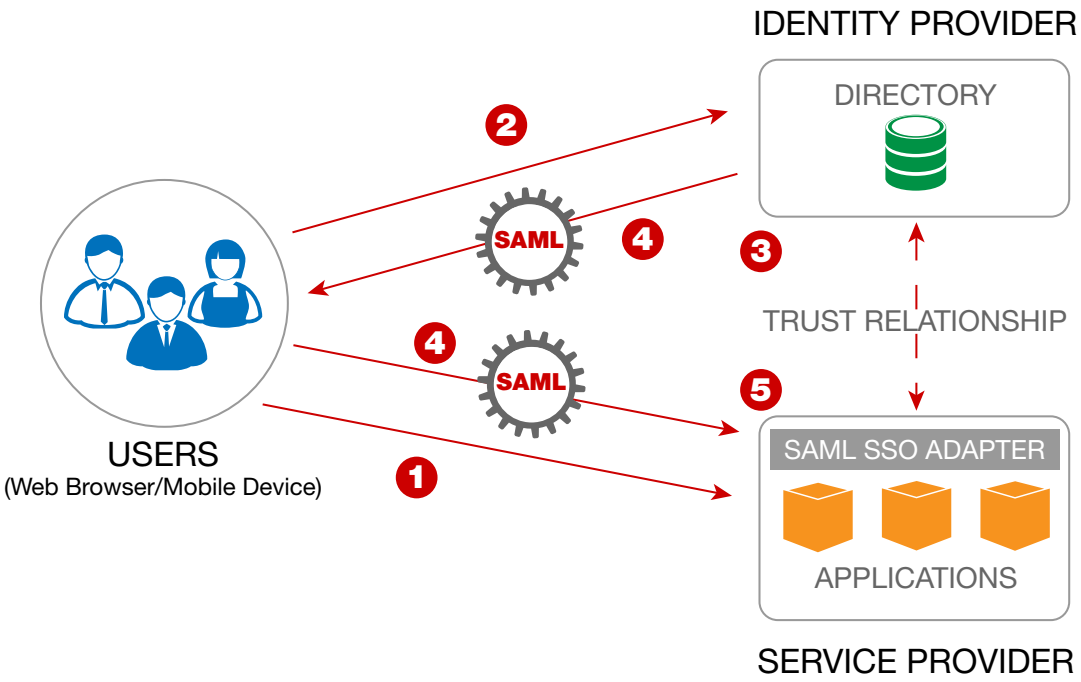
Implementing SAML SSO can be challenging for an organization. SAML consumption and generation can be complex and the PKI digital signature (DSIG) and verification components of SAML trust relationships require PKI frameworks, key generation, and several OASIS standards-based parsing capabilities. The implementation strategy of SAML is dependent on multiple factors: architecture design, legacy system enablement, development team skillset, and cost of building individual embedded SAML adapters.

Now, let's assess two different methods of implementing SAML SSO. The first option is to build or purchase pre-built SAML SSO adapters for each Service Provider that will be used for SAML SSO (sometimes these are referred to as SAML agents). The second option is to deploy an API gateway to seamlessly facilitate SAML SSO with no coding or impact on back-end services. It's important to note that these scenarios assume that the IdP is SAML 2.0 enabled.

SAML SSO Adapters

Standards-based SAML adapters have the advantage of providing a vendor-agnostic approach to identity and SSO, which is an essential part of a modernization strategy. The implementation of SAML may provide challenges at the end-mile applications simply due to framework and environment limitations, legacy components, and lack of in-house expertise. The purpose of the adapters is to embed these SAML-aware components into the infrastructure so that you can achieve scenarios represented in Figure 3 below.

Figure 3: SAML SSO without an API Gateway



1. **The User attempts to access an application from the Service Provider.**
2. **However, the User has not been authenticated and is redirected to the Identity Provider for authentication.**
3. **The IdP challenges the User for credentials. The User then authenticates at the IdP with his/her credentials.**
4. **If the credentials are valid, the SAML SSO Adapter creates a SAML token and redirects the User's browser back to the Service Provider.**
5. **The SAML SSO Adapter at the SP inspects the SAML token. If valid, the user is allowed access and a session cookie is established to enable the continued use of the SSO session for some duration of time without requiring re-authentication.**

On paper, this may seem like an easy and straightforward approach. However, there are several challenges to consider:

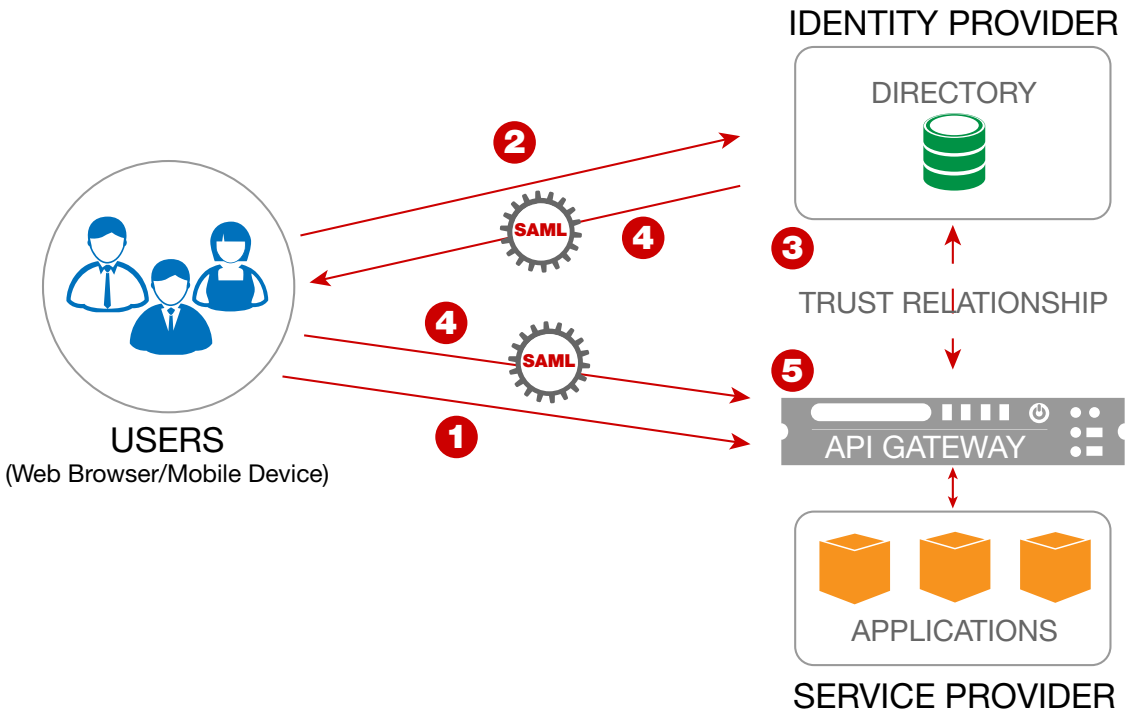
- Building your own adapters may require time and resources of your development team that you can't afford to spare. Additionally, your development team may not have the bandwidth to manage the ongoing maintenance and troubleshooting required.
- Purchasing adapters would still require proper integration and configuration. Each new or updated SP would require a thorough security audit, because each SP introduces a new point of vulnerability.

- Adapters do not shield the Service Providers from threats to the underlying platform, resulting in additional points of vulnerability, e.g. insecure SSL implementation.
- As new Service Providers are added, scalability becomes an issue. Each new SP must be coded and tested, requiring time and resources.
- The distributed adapters and potential inconsistencies in different environments could greatly complicate testing, debugging, monitoring, auditing, maintenance, and security policy changes.

SAML SSO with an API Gateway

Alternatively, you can SSO enable your enterprise architecture with an API gateway. Instead of using adapters, the API gateway takes on the responsibility of facilitating SAML SSO. Figure 4 illustrates an architecture deployment with an API gateway performing SAML SSO functions.

Figure 4: SAML SSO with an API Gateway



1. **The User attempts to access an application from the Service Provider.**
2. **However, the User has not been authenticated and the API gateway at the Service Provider redirects the User's browser to the Identity Provider for authentication.**
3. **The IdP challenges the User for credentials. The IdP validates the customer's credentials.**
4. **If the credentials are valid, the IdP creates a SAML token and redirects the User's browser back to the API gateway at the Service Provider.**
5. **The API Gateway at the SP inspects the SAML token and its contents. If valid, the API gateway grants the User access and a session cookie is established to enable the continued use of the SSO session for some duration of time without requiring re-authentication.**

Utilizing an API gateway to enable SAML SSO alleviates many of the challenges mentioned in the previous scenario:

- No coding is required. Your development team doesn't need to make any modifications to your Service Provider(s). The management and maintenance of gateway policies requires less time and resources.
- Deploying new API gateways is easy and doesn't require development team resources.
- The API gateway provides hardened security with a single point of enforcement.
- Scalability is no longer a concern. Configuring new applications and Service Providers is easy with the API gateway's point-and-click interface.
- The API gateway centralizes testing, debugging, monitoring, auditing, maintenance, and security policy changes to one centralized location, eliminating the headache of having to deal with each individual adapter.

When evaluating the best approach for your organization, you must consider your current infrastructure and future development roadmap. If you are a small organization with limited IT budget and no plans to expand beyond one or two service providers, building SAML SSO adapters may be the best route. On the other hand, if you're a company that is constantly adding new applications and Service Providers, the API gateway will suit your organization best.

About Forum Systems

Forum Systems is the global leader in API and Cloud Security technology with industry-certified and patented products that secure enterprise infrastructure. Forum Systems has been an industry leader for over 13 years and has built the core architecture of its technology on the foundation of FIPS 140-2 and NDPP. Our security-first mindset enables trusted, network edge deployments for protecting critical enterprise transactions.

Forum Systems supports enterprise customers across commercial, government, and military sectors. Forum Systems technology provides leading-edge security with identity and SSO features that enable out-of-the box business solutions with code-free configuration.

References:

Security Assertion Markup Language (SAML) 2.0 Technical Overview
https://www.oasis-open.org/committees/download.php/11511/sstc-saml-tech-overview-2.0-draft-03.pdf