The Anatomy of Single Sign On

Ewa Anna Szyszka · 2020-09-24

What is the actual cost of forgetting a password? To an individual user having to reset a password might be yet another mundane task. Still, at scale, it can be a reasonably expensive venture. To online market services, forgotten passwords mean lost business and lost productivity. When the number of forgotten passwords across the organization is multiplied by the time it takes to reset them, we realize that the gross harmful effects can be staggering. This is where federated authentication comes into the picture. Federated authentication is a way of managing an electronic identity by means such as encryption, digital signatures, and public key infrastructure.

SSO

Single Sign-On (SSO) is a subset of federated authentication management. We can distinguish two main tasks that it achieves: federation and delegation. Federation happens, for example, when you log in to another application using Google or Facebook. Delegation is the process in which one application is trying to access data from another application. For instance, your website is trying to access Facebook friends. Thanks to SSO, users can use external identity providers to authenticate and log into multiple services without having to remember numerous passwords. Using SSO, your application can also access data from other services such as Google Drive or Facebook. There are many standards available for SSO, such as SAML and OAuth/OpenID Connect. Understanding what the differences and similarities between those standards are is key to choosing the appropriate solution.

SAML

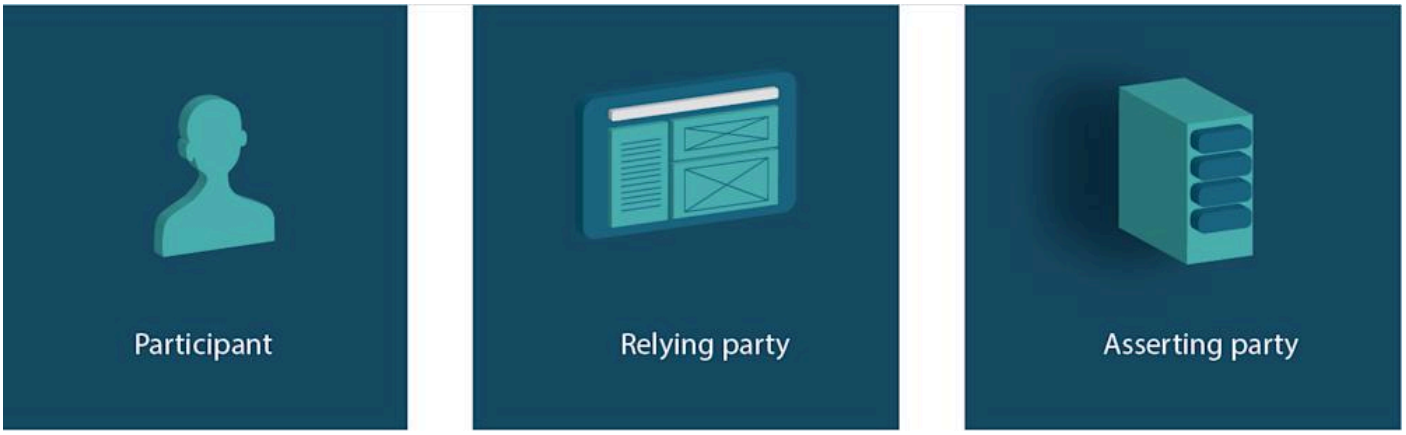
SAML is an authentication and authorization protocol based on XML most commonly used for enterprise SSO. It is important to stress the difference between authentication and authorization, as not all protocols support both like SAML.

Authentication verifies the digital identity of a user and is analogous to an online passport. In other words, authentication makes sure the user is actually who the user claims to be.

On the other hand, authorization defines what resources a user can access. Think of an intern in a company that needs access to marketing documents but who should not have access to your companies financial statements. Authorization is often implemented by roles or groups, but is not limited to that.

When speaking of the SAML standard, we can identify three parties involved: The Participant (client), the Relying party (Service provider) and an Asserting party (Identity provider). The goal of the participant is to access the service without having to remember multiple passwords. The asserting party is the middle man between the user and the service. It's role is to authenticate the user's identity by sending three types of assertions to the relying party: authentication, attribute, and authorization decision assertions [1].

The authentication assertion provides information about the type of authentication used (LDAP, Kerberos, OAuth2). The attribute assertion shares a series of characteristics (attributes) about the user that wants to be granted access. Finally, the authorization decision assertion informs whether the user was given permission or not. Once the user's identity is verified using the trusted party, the connection between service provider and the user can be established. Because the external identity provider is responsible for checking who the client is, further communication can happen using SSO.



Oauth & OpenID Connect

OAuth is a JSON based intermediary standard used solely for authorization between services. To understand what that means, imagine that you are running an online business that turns bad quality photos into high-resolution images. To do so, a user needs to log in to your website and give the service access to photos stored on their Google Drive account.

OAuth is the protocol that allows Google Drive and your website to "talk to each other." Google Drive is a service to which your client is logged in, and your website is your service that needs access to Google drive. Hence, authorization is necessary between services.

Under the hood, OAuth provides an access token that allows the account information to be passed on. By itself, OAuth does not allow for SSO. To enable SSO, we need to use an identity layer on top - Open ID Connect.

The OAuth/OpenID flow involves the participant (person to whom the Google Drive belongs), the relying party(your website), and the resource server (Google Drive). OAuth/OpenID Connect layer handles the communication between Google drive of your client and your website using ID Tokens. ID Tokens are a form of a digital passport, which is exchanged in the process of authentication.

The pros of this standard are that the authorization is done using Secure Sockets Layer (SSL), ensuring that the up-to-date cryptography standards are kept. On the other hand, the downsides of OAuth are that it does not support channel binding and client verification. Channel binding is the process of creating a unique fingerprint for the communication by binding together the transport layer and application layer together.

Vulnerabilities of SSO

As every system, SSO is not perfect and prone to attacks. The security researcher Kelby Ludwig and his team conducted research on SAML vulnerabilities. Their study explored how one could weaponize unit tests of open source SAML implementations to reveal weaknesses. The research team concluded that by inserting comments to the original XML documents, one could bypass the authentication. An attacker's inserted comments would truncate the credentials of the user, allowing him to become another user. The open-source libraries that were identified as vulnerable in the research were: python-saml, ruby-saml, saml2-js and OmniAuth-SAML.[3]

Sources

[1] https://www.varonis.com/blog/what-is-saml/

[2] https://www.softwaresecured.com/federated-identities-openid-vs-saml-vs-oauth/

[3] https://www.youtube.com/watch?v=Zjrty05REoc

[4] http://www.socialtechnologyreview.com/articles/oauth-pros-and-cons-oauth

Image:

This article was published on the Search Guard blog: https://search-guard.com/blog/elasticsearch-single-sign-on/