Federación Simplificada (v2)

Patrick Lunney, *Product Owner* - Inicio de sesión único y autenticación multifactor en Capital One

Por comentarios sobre este artículo, contacte nuestro Repositorio GitHub o reporte un problema

Tabla de contenidos

Resumen

Este artículo describe los fundamentos de la identidad federada de empresa y se enfoca en SAML y *OpenID Connect* (un protocolo basado en OAuth2.0). También ofrece algunos escenarios comunes donde se utilizan las federaciones, así como terminología de alto nivel. No se abarca la identidad federada académica, pero se la menciona brevemente con fines comparativos.

Introducción

Este artículo describe la federación de identidad en el contexto del inicio de sesión único en empresas y describe algunos casos de uso de integraciones de federación de empresa. Las empresas tienen varias formas de gestionar conexiones federadas: las conexiones pueden realizarse totalmente dentro de la empresa, de forma autogestionada en función de los controles de gobernanza existentes o mediante integraciones manuales. Cada modelo de integración tiene fortalezas y debilidades que se discutirán más adelante.

Terminología

Término	Definición
Federación de identidades	Una Federación de identidades es un grupo de proveedores de informática o de red que accede a operar utilizando los protocolos estándar y los acuerdos de confianza. En una situación de **Inicio de Sesión Único** (**SSO**, por sus siglas en inglés), la federación de identidad ocurre cuando un **Proveedor de Identidad** (**IdP**, por sus siglas en inglés) y un **Proveedor de Servicio** (**SP**, por sus siglas en inglés) acuerdan comunicarse mediante un protocolo estándar específico. El usuario de la empresa iniciará sesión en la aplicación usando sus credenciales de la empresa en lugar de crear nuevas credenciales específicas en la aplicación. Al usar un solo conjunto de credenciales, los usuarios tienen que administrar solamente un conjunto de credenciales. Los problemas relacionados con las credenciales, como el reseteo de contraseña, pueden ser gestionados en una ubicación y las aplicaciones pueden confiar en que los sistemas apropiados de la empresa (como los sistemas de recursos humanos) son una fuente confiable en lo que refiere al estatus y afiliación de un usuario. La federación de identidades puede tomar diversas formas. En el ámbito académico, las **federaciones multilaterales** en las que un tercero es confiado para gestionar los metadatos de varios IdPs y

	SPs, son muy comunes.[1] Sin embargo, este artículo se enfoca en los casos de uso para la empresa, donde los acuerdos de federación son bilaterales, es decir los acuerdos se dan uno-a-uno entre un IdP y un SP. Los acuerdos de Federación Bilateral son los más comunes hoy en día.
Federación Bilateral	Una federación bilateral es aquella que consiste solamente en dos entidades: un **Proveedor de Identidades (IdP)** y un **Proveedor de Servicios (SP)**. Este es el modelo más común para una federación de identidad empresarial.
Proveedor de Identidad (IdP, por sus siglas en inglés)	Un Proveedor de Identidad (IdP) envía información sobre un usuario a una aplicación. Generalmente, esta información está guardada en un repositorio de usuarios, para que un proveedor de identidades tome esa información, la transforme para pasarla al proveedor de servicio, es decir la aplicación. La organización OASIS, que es la responsable de las especificaciones SAML, define que un IdP es un "tipo de SP que crea, mantiene y gestiona la información de identidad de entidades principales y provee autenticación a otros SP dentro de la federación, como es el caso de los navegadores de Internet."[2]
Federación multilateral	Es una federación que consiste en múltiples entidades que han acordado un marco de confianza específico. Existen muchas formas de federación multilateral incluyendo modelos de distribución *hub-and-spoke* y *mesh*. Las federaciones multilaterales son el modelo de federación de identidad más usado en el ámbito académico.
OAuth 2.0	OAuth 2.0 es un protocolo de código abierto que permite a los propietarios de recursos, como aplicaciones, compartir datos con sus clientes facilitando la comunicación con un Servidor de Autorización (AS, por sus siglas en inglés). Los datos toman la forma de credenciales otorgadas a las aplicaciones para obtener información/datos de otras aplicaciones. El Servidor de Autorización es por lo general el Proveedor de Identidad (IdP). El Servidor de Autorización puede otorgar autorizaciones directa o indirectamente. Por ejemplo, el AS puede proveer atributos o datos del perfil del propietario de recursos u otorgar acceso a datos que puedan ser usados con propósitos de autorización, como permisos de un IDM o IGA.

[1] "Federación multilateral," *InCommon Federation wiki*, última actualización el 17 de febrero de 2020, https://spaces.at.internet2.edu/display/federation/Multilateral+federation.

[2] Hodges, Jeff, Rob Philpott, Eve Maler, eds. "Glosario del lenguaje de marcado para confirmaciones de seguridad (SAML) de OASIS V2.0," Estándar OASIS, 15 de marzo de 2005, https://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf.

OpenID Connect	*OpenID Connect* es una capa de identidad simple sobre el protocolo OAuth 2.0. Habilita a los clientes a verificar la identidad del usuario final basándose en la autenticación llevada a cabo por un servidor de autorización y permite obtener un perfil de información básico sobre el usuario final de manera interoperable y siguiendo la transferencia de estado representacional (REST, por sus siglas en inglés).
Lenguaje de marcado para confirmaciones de seguridad (SAML, por sus siglas en inglés)	SAML es un protocolo de comunicación basado en XML entre los SP y los IdP.[3] En general, la empresa aloja el IdP mientras que las aplicaciones (incluyendo servicios en la nube) son los SP.
Proveedor de servicios (SP, por sus siglas en inglés)	Según la definición de la organización OASIS que es la responsable de las especificaciones SAML, es un "rol otorgado por una entidad del sistema donde la entidad del sistema provee servicios a entidades principales u otras entidades del sistema". En general es una aplicación que ofrece servicios a usuarios que requieren autenticación y autorización.
Inicio de Sesión Único (SSO, por sus siglas en inglés)	Un Inicio de Sesión Único es un servicio que habilita a un Proveedor de Servicios (SP) a verificar identidades de **usuarios finales**, facilitando la comunicación con los Proveedores de Identidades (IdP). SSO hace de puente para deslindar los SP y los IdP. Esto se puede hacer a través de varios protocolos como la integración basada en agente, integración directa LDAP, SAML u *OpenID Connect*, por mencionar algunos.

Explorando la identidad federada en empresa

Existen muchos escenarios comunes que los profesionales de identidad pueden encontrar federación de identidad en el contexto de una empresa. Este apartado explora los protocolos más comunes: *OpenID Connect* y SAML.

Caso de uso 1: SAML

[3] Ragouzis, Nick, John Hughes, Rob Philpott, Eve Maler, Paul Madsen, Tom Scavo, eds. "Lenguaje de marcado para confirmaciones de seguridad (SAML) V2.0 Resumen técnico" Borrador del Comité OASIS, 25 de marzo de 2008, https://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.pdf.

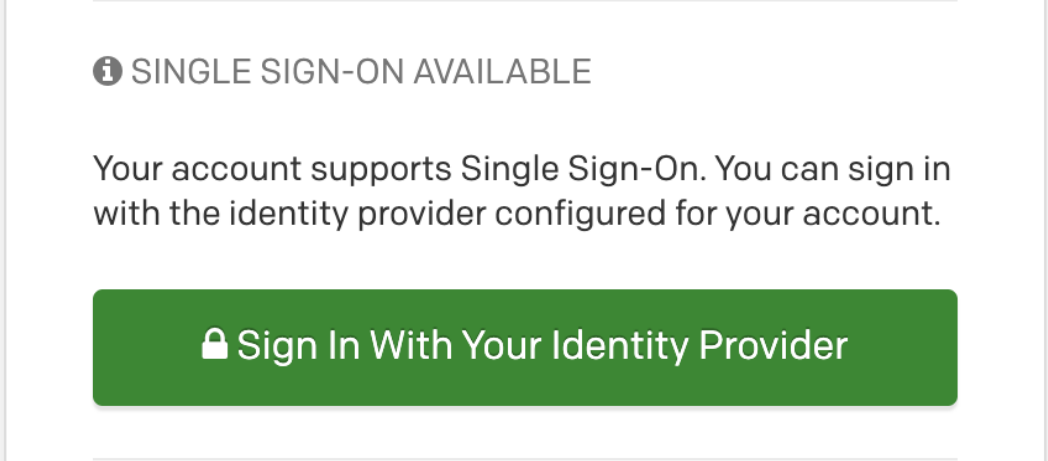

Figura 1 - Ejemplo de Interfaz de Inicio de Sesión Único

Generalmente, SAML se encuentra en aplicaciones Software como Servicio (SaaS, por sus siglas en inglés). Una aplicación es comprada o creada por una empresa para hacer "algo" y los empleados deben iniciar sesión en la aplicación. La aplicación necesitará intercambiar información con la empresa para hacer esta federación. En general, un IdP (la empresa) y un SP (la aplicación) intercambiarán metadatos que les permitirán establecer las conexiones en el sistema de Inicio de Sesión Único (SSO, por sus siglas en inglés). El intercambio de metadatos puede realizarse manualmente, pero esto lleva tiempo y puede ser un dolor de cabeza para los IdP y SP.

En el Ítem 1 del apéndice encontrará un ejemplo de un archivo de metadatos de un IdP. En este ejemplo, el operador IdP entregará dichos metadatos al operador SP. El SP puede ingresar esta información manualmente (o importarla, dependiendo de su plataforma de SSO) en su sistema SSO para permitir que los usuarios de la empresa inicien sesión en la aplicación usando sus cuentas SSO. El operador IdP tendrá que hacer lo mismo, ya sea importando el archivo de metadatos del SP o manualmente, actualizando la configuración del IdP.

Un archivo de metadatos de un IdP contiene la ID de entidad de la empresa, las URL usadas en SAML y los atributos que se pasarán en la aserción SAML (los datos que se entregan a la aplicación). Una ID de entidad es un nombre único para una entidad SAML, tanto para el IdP como para el SP. Ningún IdP o SP puede compartir la misma ID de entidad.

Piense una aserción SAML como si fuera un vale o ticket. El IdP le da al usuario un vale para ingresar al SP y el SP valida el vale usando validación de certificados. Una vez que el ticket es validado, el SP mirará los atributos para ver qué es lo puede hacer el usuario. Por ejemplo, en el Ítem 2 del apéndice, puede ver que un nombre de usuario y un correo electrónico se pasaron a ese SP.

Para más información sobre las aserciones SAML y sus componentes, vea la especificación SAML y los documentos de apoyo asociados.[4]

Un detalle más al respecto de las federaciones SAML de empresa: hay dos tipos diferentes de URLs para aplicaciones. A veces es la URL del SP, por ejemplo 'https://myhrapp.com/enterprise'. Esto se conoce como una solicitud iniciada por un SP. Otras veces, el IdP inicia la solicitud. Por ejemplo, 'https://authn.enterprise.com/idp/SAML20=myhrapp'. En ambos casos, el usuario deberá iniciar sesión en la misma aplicación inquilina de la empresa. Algunas aplicaciones soportan únicamente solicitudes de inicio de sesión iniciadas por el IdP. Otras, solo soportan solicitudes iniciadas por el SP:

A continuación, encontrará un diagrama de flujo de una autenticación SAML estándar:

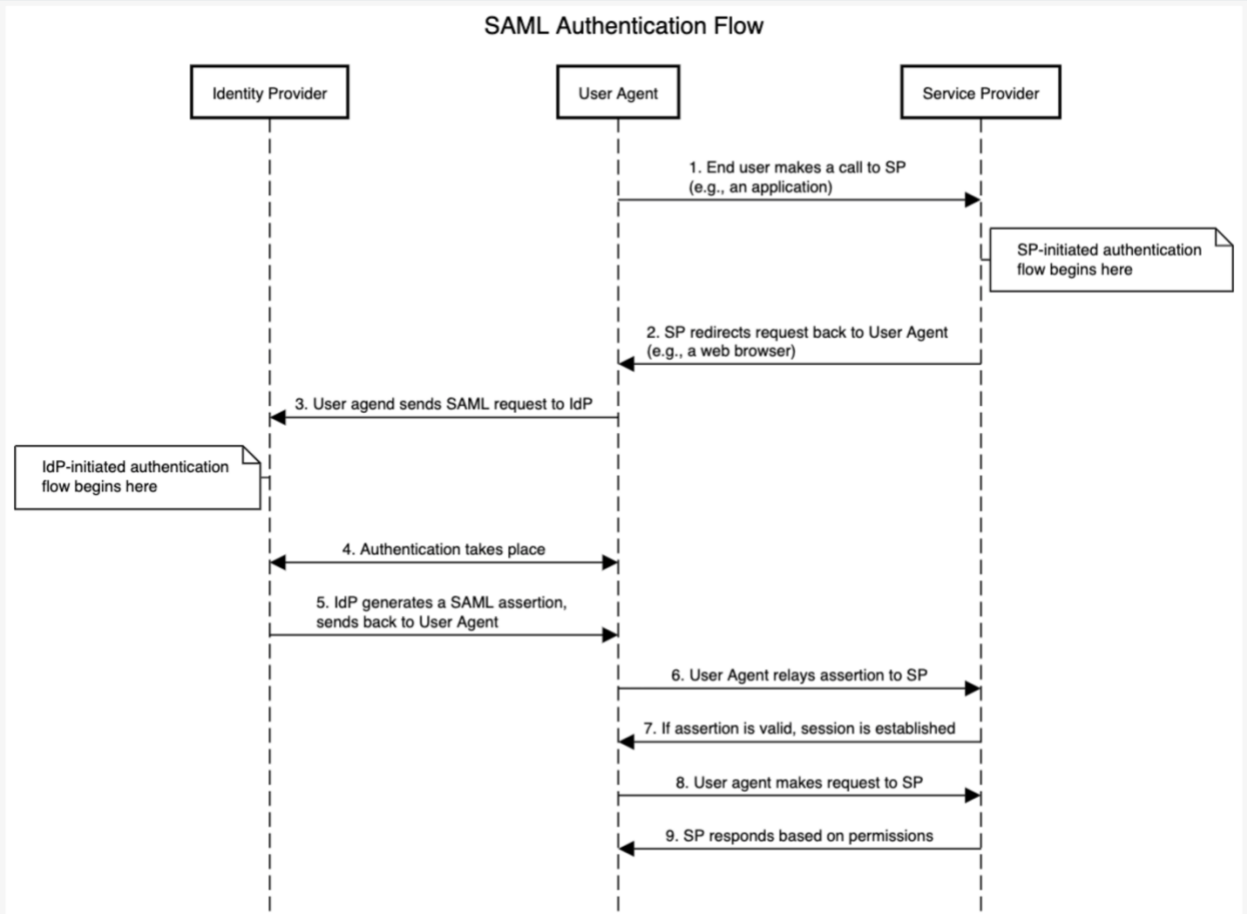

Figura 2 - Flujo de Autenticación SAML

[4] Página web de los estándares OASIS, https://www.oasis-open.org/standards/.

Es importante destacar que la *autenticación* del usuario se completa en el paso 5; el IdP ya validó las credenciales del usuario y pasa la aserción SAML nuevamente al navegador. La federación se completa en el paso 7; el navegador reenvía la aserción a la aplicación para que la misma sepa que el usuario ha sido autenticado y cree una sesión para ese usuario. En los pasos 8 y 9, se lleva a cabo la *autorización*. Basándose en la información provista por el IdP, la aplicación otorgará o denegará el acceso al usuario a determinadas partes de la aplicación.

Caso de uso 2: *OpenID Connect*

Otro tipo de federación de identidad común es interno a la empresa y cada vez más se encuentra en productos SaaS. Antes, las empresas usaban "agentes" que instalaban en los servidores web que alojaban aplicaciones. Los agentes se comunicaban con algo llamado servidor de políticas para determinar qué podía hacer o no un usuario. Esta tecnología de servidor de agente/políticas es vieja y ya casi no se utiliza en las empresas.

En su lugar, un protocolo popular que se utiliza cada vez más es *OpenID Connect* (OIDC). *OpenID Connect* es más nuevo que SAML y se basa en el protocolo OAuth2.0; la mayoría de las aplicaciones internas de empresa se basan en APIs y microservicios, motivo por el cual se prefiere OIDC.[5] Cabe destacar que algunas aplicaciones SaaS soportan *OpenID Connect*.

OpenID Connect utiliza el otorgamiento de código de autorización de OAuth2.0. Dado que *OpenID Connec*t comparte los atributos del usuario, mencionamos a OAuth2.0 en este artículo, pero será la única parte donde lo haremos. Existen muchos otros tipos de flujos (*grant_types*) en OAuth2.0 que autentican usuarios o clientes de diferentes maneras, pero estas no forman parte de la *autenticación* y *autorización* de usuario y por lo tanto están por fuera del alcance de este documento.

Flujo del código de autorización

El tipo de flujo de código de autorización está explicado en la especificación OAuth2.0.[6] *OpenID Connect* 1.0 se basa en este flujo. Un elemento importante para tener en cuenta involucra los alcances (*scopes*) de *OpenID Connect*: deben contener *openid* (y generalmente incluye *profile*). A continuación, encontrará un diagrama de flujo de ese código de autorización:[7]

[5] Hardt, D., Ed., "La infraestructura de autorización OAuth 2.0", RFC 6749, DOI 10.17487/RFC6749, octubre de 2012, <https://www.rfc-editor.org/info/rfc6749>.
[6] Ibid., vea la sección 4.1.
[7] Hardt, D., Ed., "La infraestructura de autorización OAuth 2.0", RFC 6749, Sección 4.1, DOI 10.17487/RFC6749, octubre de 2012, <https://www.rfc-editor.org/info/rfc6749>.

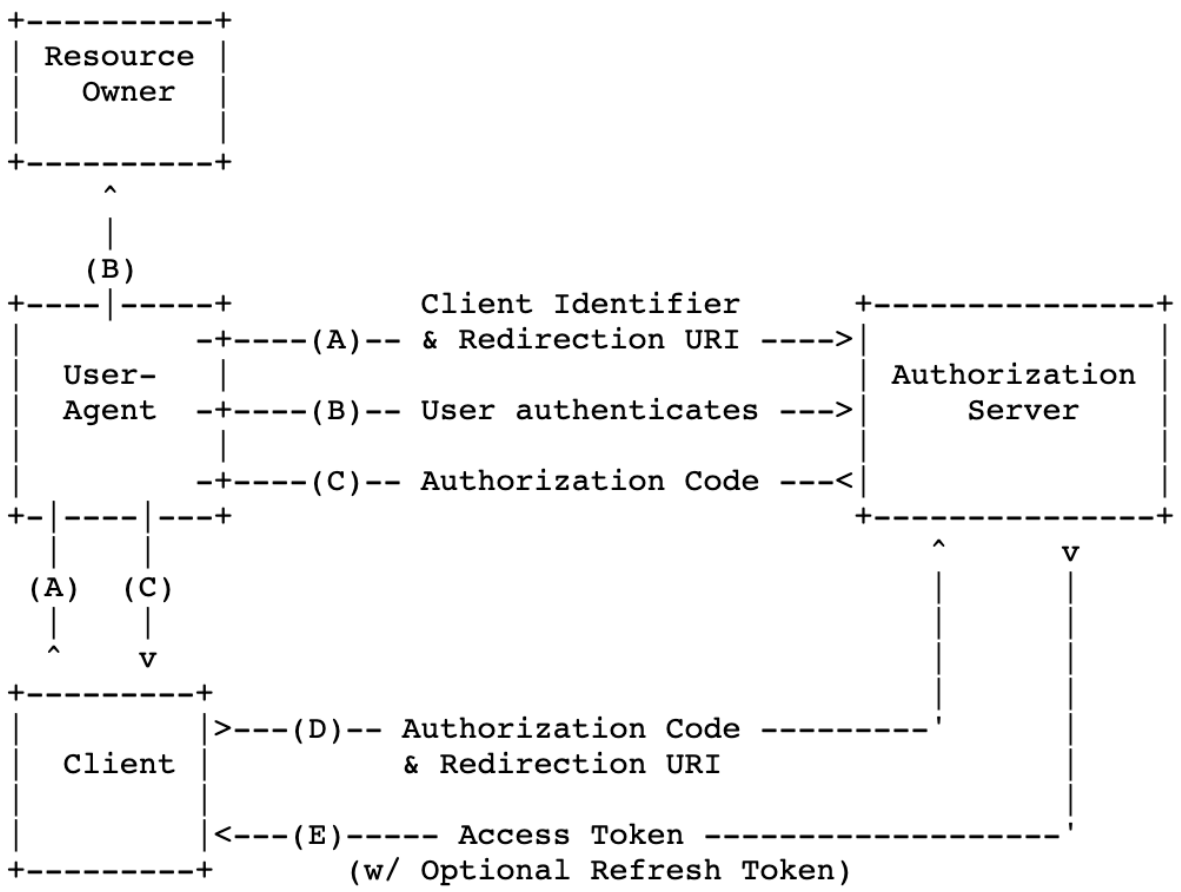

Figura 3 - Flujo de otorgamiento de autorización OAuth 2.0

En este diagrama vemos que el usuario irá primero a un navegador (agente de usuario) e iniciará una solicitud ante el servidor de autorización. El servidor de autorización requerirá que el usuario ingrese sus credenciales (B). Luego de recoger las credenciales, el navegador enviará esa información al servidor de autorización que a su vez responderá enviando un código al navegador (C). El *backend* de la aplicación (Cliente, C) tomará ese código y lo cambiará por un token de acceso (D, E). En *OpenID Connect* existe un paso opcional F en el cual el cliente puede solicitar información adicional sobre el usuario (atributos) realizando una solicitud API al *endpoint 'userinfo'*. El Servidor de Autorización responderá a esta solicitud API entregando la información del usuario y permitiendo así que el cliente *autorice* el usuario.

Para ver las solicitudes API, refiérase al ítem 3 del apéndice.

Los desafíos de las federaciones de empresa

¿Cuándo usar SAML versus *OpenID Connect*?

La respuesta corta a esta pregunta es: depende. A veces existen límites en lo que pueden hacer los SP y los IdP. Ambas integraciones tienen sus pros y sus contras así que es realmente un tema de elección (o de límites) entre los IdP y los SP.[8]

El artículo "Introducción a la identidad - Parte 2: Administración de accesos" del Cuerpo de Conocimiento de IDPro escrito por Pamela Dingle brinda una visión interesante sobre la evolución de las herramientas de autenticación y control de accesos.[9] Concretamente la sección 'La innovación móvil y API nos dio OAuth y las estructuras de autorización delegada ' ofrece una perspectiva interesante de la evolución que llevó al desarrollo de OAuth a pesar de la existencia de SAML.

Atributos - datos y formateo

Las aplicaciones requieren diferentes nombres para los atributos. A veces un atributo debe llamarse primernombre mientras que otras aplicaciones pueden requerir primernombre o quizás nombredepila. Esto puede acarrear problemas ya que la aplicación puede no ser capaz de recoger el atributo en la aserción SAML/*endpoint userinfo* que necesita para autorizar al usuario. Aquí es donde el IdP y SP deben colaborar para determinar cómo deben enviarse los atributos. En algunas empresas el atributo nombre no cambia; la empresa fuerza la aplicación a adoptar su formato de atributo. En otras ocasiones, la aplicación fuerza al IdP a cambiar los atributos. También puede ocurrir algo llamado mapeo de atributos. La mayoría de los *plugins* SAML y *OpenID* permiten que esto ocurra en archivos de mapeo de atributos, como Shibboleth.[10] El IdP enviará los atributos y luego de recibirlos, el SP podrá transformarlos al formato correcto.

Tamaño de la aserción

Una gran cantidad de información puede ser pasada al SP y la aserción puede tornarse tan grande que puede romper el SP. Esto es bastante común cuando las aplicaciones autorizan usuarios mediante un Directorio Activo o grupos LDAP (también conocidos como inflación SID, que son básicamente grandes blobs de datos con información sobre el usuario), y el IdP envía un despliegue de todos los grupos del Directorio Activo. La aserción SAML contendrá tanta información que el SP no podrá analizarla y el usuario no podrá entrar en la aplicación. Para resolver este problema a menudo se necesitan integraciones personalizadas que requieren una configuración especial en el IdP para gestionar las

---

[8] Para más información sobre los pros y los contras de SAML y OAuth, vea https://www.okta.com/identity-101/whats-the-difference-between-OAuth-openid-connect-and-saml/ o https://auth0.com/intro-to-iam/saml-vs-openid-connect-oidc/

[9] Dingle, Pamela, "Introducción a la identidad – Parte 2: Administración de accesos," Cuerpo de Conocimiento de IDPro, 17 de junio de 2020, https://bok.idpro.org/article/id/45/.

[10] Shibboleth Consortium, https://www.shibboleth.net/.

aserciones para esa única aplicación. Asimismo, el tamaño de las aserciones puede limitarse basándose en servidores web, navegadores e incluso servidores *proxy* . El problema puede solucionarse mediante procesos de gobernanza de identidad que limitan el número de grupos de Directorio Activo y eliminan membresías innecesarias.

Intercambio de Recursos de Origen Cruzado (CORS, por sus siglas en inglés)

El Intercambio de Recursos de Origen Cruzado, comúnmente conocidos como CORS, causa problemas en muchas empresas. CORS es un estándar que permite que un servidor relaje la política del mismo origen.[11] En general, una llamada API de una aplicación no puede enviarse a otra aplicación. Por ejemplo, si hago una solicitud a application1.com/api, espero que la solicitud regrese a mí y no que se envíe a application2.com/api. Se trata de dos dominios diferentes y application1.com podría estar enviando información maliciosa a application2.com/api.

CORS se usa expresamente para permitir ciertas solicitudes de origen cruzado y rechazar otras. Por ejemplo, si un sitio ofrece un servicio integrable es posible que sea necesario relajar ciertas restricciones. Si intento cargar application1.com y la misma requiere recursos de application2.com, mi navegador hará la solicitud a través de application1.com y la enviará a application2.com, haciendo así una llamada API de dominio cruzado. CORS permite que la solicitud pase y recupere la información de modo que yo pueda visitar la aplicación.

Implementar una configuración CORS es un desafío. Es también potencialmente inseguro. Lo que pueden hacer la mayoría de los IdP es relajar sus políticas para permitir intercambios entre dominios de alto nivel, por ejemplo, *.enterprise.com o *.partner.com. De esta forma, no habrá restricciones en el origen de las solicitudes.[12]

[11] "Política del mismo origen," MDM documentos web, https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy.

[12] Para más información, vea https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS y https://web.dev/cross-origin-resource-sharing/.

Conclusión

Este documento es una reseña de alto nivel sobre las federaciones de aplicaciones en empresas. Los protocolos más comúnmente usados son SAML y *OpenID Connect*. Ambos son ampliamente utilizados en el mundo empresarial, así como en el de los consumidores. Cuando ves esta pantalla:

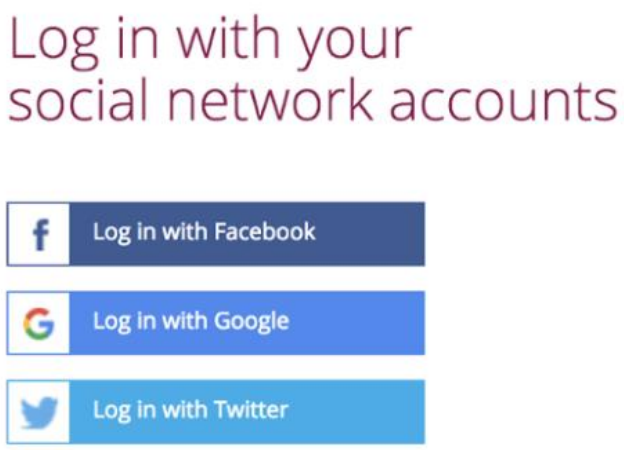

Figura 4 - Ejemplo de una pantalla de inicio de sesión con una red social

estás de hecho seleccionando el IdP con el que quieres iniciar sesión en el SP. En la mayoría de los casos, también puedes iniciar sesión directamente en la aplicación. Cabe destacar que cuando inicias sesión en una aplicación utilizando un Proveedor de Identidad como una red social, estás pasando información sobre tu persona, de la misma forma que una empresa pasa información sobre ti a un SP en la empresa. En cuanto a las redes sociales, es importante entender los términos y condiciones en cuanto a lo que pueden hacer con tus datos. En aplicaciones de empresa, esto se hace con equipos legales de por medio para garantizar que no habrá filtraciones de datos.

Dado que cada vez más aplicaciones son aplicaciones SaaS, las empresas están formando cada vez más federaciones. Por ello, habrá innovaciones continuas en la comunidad del inicio de sesión único a fin de que sean más seguras, incorporando la autenticación multifactor en el flujo, por ejemplo.

Biografía del Autor

Me llamo Patrick Lunney. En los últimos ocho años, he gestionado y he sido dueño de proveedores de identidad en dos empresas *Fortune 50*. Durante ese período, he trabajado con cientos de aplicaciones SaaS así como con aplicaciones locales para garantizar que las federaciones se realicen de forma correcta y segura. Actualmente soy el propietario de los productos Inicio de Sesión Único y Autenticación Multi Factor para Capital One. Todas las aplicaciones en nuestra empresa deben usar *OpenID Connect* o SAML para el Inicio de Sesión Único, salvo contadas excepciones. Desempeño esta función desde el año 2019.

Registro de cambios

Fecha	Cambio
03-06-2022	V2 publicada; Título cambiado, actualización de la definición OIDC, detalle agregado: flujos iniciados en el SP
19-04-2021	V1 publicada

Apéndice:

Ítem 1: Solicitud SAML

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" ID="mzWO1kVu-
dAmFIdmN.08s9bOaCH" cacheDuration="PT1440M" entityID="IdProvider">
   <md:IdPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
   WantAuthnRequestsSigned="false">
      <md:KeyDescriptor use="signing">
         <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <ds:X509Data>
               <ds:X509Certificate>
               </ds:X509Certificate>
            </ds:X509Data>
         </ds:KeyInfo>
      </md:KeyDescriptor>
      <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</md:NameIDFormat>
      <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="https://authn.enterprise.com/idp/SSO.saml2"/>
      <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="https://authn.enterprise.com/idp/SSO.saml2"/>
      <saml:Attribute xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Name="firstname"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified"/>
      <saml:Attribute xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Name="groups"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified"/>
      <saml:Attribute xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Name="lastname"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified"/>
      <saml:Attribute xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Name="userid"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified"/>
      <saml:Attribute xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Name="email"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified"/>
```

```
    </md:IdPSSODescriptor>
    <md:ContactPerson contactType="administrative"/>
</md:EntityDescriptor>
```

Ítem 2: Respuesta SAML

```
<samlp:Response Destination="https://serviceprovider.com/acs"
   ID="HpiyLr_zVMK.jxdUHXxRvjJ8Fwy" IssueInstant="2020-11-24T01:53:06.809Z" Version="2.0"
   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
   <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">IDprovider</saml:Issuer>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
         <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
         <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
         <ds:Reference URI="#HpiyLr_zVMK.jxdUHXxRvjJ8Fwy">
            <ds:Transforms>
               <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <ds:DigestValue>PwJICHFA1QIlML2p5MyJaRib5TDY4TWj5J7IEAjn1Yo=</ds:DigestValue>
         </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue> Signature
      </ds:SignatureValue>
      <ds:KeyInfo>
         <ds:X509Data>
            <ds:X509Certificate>
</ds:X509Certificate>
</ds:X509Data>
         <ds:KeyValue>
            <ds:RSAKeyValue>
                        <ds:Modulus>
                        </ds:Modulus>
               <ds:Exponent>AQAB</ds:Exponent>
            </ds:RSAKeyValue>
         </ds:KeyValue>
      </ds:KeyInfo>
</ds:Signature>
<samlp:Status><samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</samlp:Status>
<saml:Assertion ID="bJUFiJZEXV0rDgdTh9HnF2CbrIq" IssueInstant="2020-11-24T01:53:07.104Z"
      Version="2.0" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
      <saml:Issuer>IDprovider</saml:Issuer>
      <saml:Subject>
         <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">ztl593</saml:NameID>
         <saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"><saml:SubjectConfirmationData NotOnOrAfter="2020-11-
24T01:58:07.104Z"
            Recipient="https://serviceprovider.com/acs"/></saml:SubjectConfirmation>
      </saml:Subject>
      <saml:Conditions NotBefore="2020-11-24T01:48:07.104Z" NotOnOrAfter="2020-11-24T01:58:07.104Z">
         <saml:AudienceRestriction>
```

```
            <saml:Audience>http://www.serviceprovider.com/</saml:Audience>
         </saml:AudienceRestriction>
      </saml:Conditions>
      <saml:AuthnStatement AuthnInstant="2020-11-24T01:53:07.103Z"
         SessionIndex="bJUFiJZEXV0rDgdTh9HnF2CbrIq">
         <saml:AuthnContext>

<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Telephony</saml:AuthnContextClassRef>
         </saml:AuthnContext>
      </saml:AuthnStatement>
      <saml:AttributeStatement>
         <saml:Attribute Name="mail" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
            <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">Patrick.Lunney@idprovider.com</saml:AttributeValue>
         </saml:Attribute>
      </saml:AttributeStatement>
   </saml:Assertion>
</samlp:Response>
```

Ítem 3: *OpenID Connect*

Para comenzar el proceso, el agente de usuario primero realizará una solicitud GET al servidor de autorización, transmitiendo información sobre la aplicación a la que el usuario desea llegar.

```
curl --request GET \
--header ‘content-type: application/x-www-form-urlencoded' \
 --url
“${sso_prefix}/authorization?response_type=code&redirect_uri=${redirect_uri}&scope="op
enid profile”&client_id=${client_id}
```

El resultado de esta solicitud es la página de inicio de sesión (suponiendo que no haya sesión), y un usuario ingresará sus credenciales para que el servidor de autorización pueda autenticar al usuario. Posteriormente, se envía un código_autorización a la aplicación en el navegador. El backend de la aplicación debe tomar ese código_autorización e intercambiarlo por un token de acceso.

Para intercambiar el código_autorización por el token de acceso:

```
curl --request POST \
      --url “https://${sso_prefix}/token” \
      --header 'content-type: application/x-www-form-urlencoded' \
      --header 'Authorization: Basic base64(urlencode("${client_id}:${client_secret}))' \
      --data “code=${code}” \
      --data “grant_type=authorization_code” \
      --data “redirect_uri=${redirect_uri}” \
```

```
    --data 'scope=openid profile'
```

Después de este intercambio, la aplicación puede realizar una llamada API backend al servidor de autorización para obtener información adicional sobre el usuario para una autorización adicional.

```
curl --request GET \
--header ‘content-type: application/x-www-form-urlencoded' \
--header 'Authorization: Bearer ${token}
 --url “${sso_prefix}/userinfo
```

Esto les dará a las aplicaciones información como esta:

```
{
  "sub"        : "83692",
  "name"       : "Alice Adams",
  "email"      : "alice@example.com",
  "department" : "Engineering",
  "birthdate"  : "1975-12-31"
}
```