Universitat Oberta de Catalunya

uoc.edu

MFA and Identity Federations

Máster de Seguridad de las Tecnologías de la Información y las Comunicaciones

Sistemas de Autenticación y Autorización

Autor: Montañés Navarro, Edgar

Consultor: Guijarro Olivares, Jordi

PRA: García Font, Víctor

Junio 2019

Ficha de trabajo final

Título de trabajo:	*MFA and Identity Federations*
Nombre del autor:	*Edgar Montañés Navarro*
Nombre del consultor/a:	*Jordi Guijarro*
Nombre del PRA:	*Víctor García Font*
Fecha de entrega (mm/aaaa):	*06/2019*
Titulación:	*Máster de Seguridad de las Tecnologías de la Información y las Comunicaciones*
Área del Trabajo Final:	*Sistemas de Autenticación y Autorización*
Idioma del trabajo:	*Español (Alf. Int.)*

Resumen

Cada vez es más habitual que las aplicaciones necesarias para el desarrollo de negocio de la Organización sean accesibles desde Internet mediante un portal web, es por eso que, es necesario poner un control de acceso. El hecho de tener que autenticarse en cada una de las aplicaciones es incómodo para el usuario, por lo que, entra en juego la Identidad Federada.

Por otro lado, se ha visto que un usuario y una contraseña ya no es suficiente como sistema de autenticación y autorización, dado que se están produciendo incidentes de seguridad relacionados con robo de credenciales. Así pues, es necesario implantar un segundo factor de autenticación o sistema multifactor para protegerse de dichos ataques.

El objetivo de este estudio es realizar un estado del arte exhaustivo de los diferentes entornos de federación de identidad, así como de las regulaciones existentes en la misma si queremos utilizar un servicio de federación, y las posibles herramientas que se pueden encontrar en Internet para integrarla. Además, se realizará un estudio de los diferentes factores de autenticación existentes, así como de las diferentes tecnologías y servicios que pueden ayudar a implementarlo. Tras el estudio, se construirá un entorno que disponga de identidad federada y se detallará el proceso de autenticación y autorización.

Palabras clave: Autenticación multifactor (MFA), Identidad federada, Segundo factor de autenticación (2FA).

Abstract

It is becoming more common that the necessary applications for the development of the business of the Organization are accessible from the Internet through a web portal, that is why, it is necessary to put an access control. It can result annoying having to authenticate in each of the applications, therefore, the Federated Identity comes into play.

On the other hand, it has been seen that a user and a password is no longer enough as an authentication and authorization system, given that security incidents related to theft of credentials are occurring. For that reason, it is necessary to implement a second factor authentication or multifactor system to protect against such attacks.

The aim of this review is to perform an exhaustive study of the different identity federation environments, as well as its existing regulations if we want to use a federation service, and the possible tools that can be found on the Internet to integrate it. . In addition, a study of the different existing authentication factors will be conducted, as well as the different technologies and services that can help to implement it. After the study, an environment that has a federated identity will be built and the authentication and authorization process will be detailed.

Keywords: Multifactor Authentication (MFA), Identity federation, Second factor autentication (2FA).

Tabla de contenidos

Índice de figuras

Índice de comandos

Índice de tablas

1 Introducción

1.1. Contexto y justificación

Cada vez es más habitual que las diferentes aplicaciones necesarias para el desarrollo de negocio de la Organización sean accesibles desde Internet mediante un portal web, es por eso que, es necesario poner un control de acceso. El hecho de tener que autenticarse en cada una de las aplicaciones es incómodo para el usuario, por lo que, entra en juego la Identidad federada.

Por otro lado, se ha visto que un usuario y una contraseña ya no es suficiente como sistema de autenticación y autorización [1], dado que en los últimos años estamos viendo incidentes de seguridad relacionados con *leaks* de contraseñas, así pues, es necesario implantar un segundo factor de autenticación (2FA, del inglés *Second Factor Autentication*) o sistema multifactor para protegerse de dichos ataques. Además, se ha visto que no todos los sistemas de segundo factor de autenticación son igual de seguros, dado que ya se han visto incidentes de seguridad relacionados con el Servicio de Mensajes Cortos (SMS, del inglés *Short Message Service*) como segundo factor de autenticación.

La necesidad a cubrir en este estudio es realizar un estado del arte de los diferentes entornos de federación de identidad, así como de los diferentes factores de autenticación existentes en la actualidad. El problema se resuelve realizando el estudio del funcionamiento de la identidad federada y de los diferentes factores de autenticación. Además, se hará un estudio sobre las diferentes herramientas y soluciones de código abierto que se ofrecen en Internet para poder integrar dichas tecnologías.

De esta manera, se construirá un entorno que disponga de identidad federada. Se aprovechará dicho entorno para describir el proceso de autenticación y autorización desde un proveedor de servicios hasta que el usuario puede acceder al mismo.

1.2. Objetivos

Los objetivos a cubrir en este Trabajo son:

- Analizar los diferentes factores de autenticación.

[1] Autenticación: Determinar que los usuarios son quienes dicen ser.
Autorización: Determinar si los usuarios tienen derecho a acceder a ciertos sistemas o contenidos.

- Analizar los diferentes servicios y tecnologías que pueden ayudar a la implantación de los factores de autenticación.
- Comprender el funcionamiento de la Identidad federada.
- Ahondar en los atributos necesarios que debemos utilizar si queremos utilizar servicios de federación.
- Describir las diferentes herramienta que pueden ayudar a implantar identidad federada en una aplicación.
- Crear un entorno que nos permita comprobar el funcionamiento de las tecnologías descritas.

1.3. Enfoque y método seguido

Cómo se llegará a realizar cada uno de estos objetivos:

- Referente a los diferentes factores de autenticación se realizará una revisión de todos los sistemas existentes, como son: SMS, OTP y llaves de seguridad U2F, explicando su funcionamiento y describiendo bondades y debilidades.
- Referente a la Identidad federada se estará analizando los diferentes estándares, la regulación en los atributos y el proceso de validación.
- Por último, se pretende diseñar un caso real. Se diseñará y se expondrá el proceso y se finalizará realizando una demostración de su correcto funcionamiento.

En referencia al estudio, será principalmente un estudio cualitativo, aunque se puede entender también cuantitativo, en cuanto a que, se valoran y analizan varios métodos diferentes.

1.4. Planificación

PLANIFICACIÓN TFM

LISTA DE TAREAS	FECHA DE COMIENZO	FECHA DE VENCIMIENTO	Tiempo	% COMPLET	LISTO
ENTREGA 1	20/02/2019	05/03/2019	13 días	100%	
La explicación detallada del problema a resolver.	20/2/19	22/2/19	2 días	100%	
La enumeración de los objetivos que se quieren alcanzar con la realización del TFM.	22/2/19	24/2/19	2 días	100%	
La descripción de la metodología que se seguirá durante el desarrollo del TFM.	24/2/19	26/2/19	2 días	100%	
El listado de las tareas a realizar para alcanzar los objetivos descritos.	26/2/19	28/02/2019	2 días	100%	
La planificación temporal detallada de estas tareas y sus dependencias.	28/02/2019	02/03/2019	2 días	100%	
ENTREGA 2	06/03/2019	02/04/2019	27 días	100%	
Estado del arte sobe los tipos de autenticación multifactor	06/03/2019	10/03/2019	4 días	100%	
Estado del arte sobre servicios y tecnologías MFA	10/03/2019	16/03/2019	6 días	100%	
Estado del arte sobre identidad federada	16/03/2019	20/03/2019	4 días	100%	
Investigación sobre la regulación de atributos en Identidad federada	20/03/2019	26/03/2019	6 días	100%	
Estudio de las diferentes herramientas que pueden ayudar a implantar identidad federada en una aplicación.	26/03/2019	02/04/2019	7 días	100%	
ENTREGA 3	03/04/2019	30/04/2019	27 días	100%	
Describir un ejemplo de aplicación de Identidad federada	03/04/2019	07/04/2019	4 días	100%	
Realizar anexo para preparar entorno para la aplicación	07/04/2019	11/04/2019	4 días	100%	
Desarrollar y documentar aplicación	11/04/2019	30/04/2019	19 días	100%	
ENTREGA 4	01/05/2019	04/06/2019	34 días	100%	
Recopilación de referencias	01/05/2019	08/05/2019	7 días	100%	
Enumerar conclusiones del trabajo y objetivos alcanzados	08/05/2019	15/05/2019	7 días	100%	
Realización de la memora	15/05/2019	04/06/2019	20 días	100%	

1.5. Breve sumario de productos obtenidos

Los productos que se pretenden obtener tres la finalización de este Trabajo son:

- Análisis de los diferentes factores de autenticación.
- Análisis de la identidad federada.
- Producto funcional de una aplicación con identidad federada.

1.6. Breve descripción de los otros capítulos de la memoria

La memoria cuenta con los siguientes apartados:

- Un primer apartado, en el que se realiza una Introducción del tema y se especifica el porqué resulta interesante su estudio. Además, se señalan los objetivos del mismo y cómo se pretenden conseguir. Posteriormente, se realiza una distribución de tareas temporal.
- Un segundo apartado, en el que se realiza un estado del arte de las diferentes tecnologías que entran a formar parte de la identidad federada y de la autenticación multifactor.
- Un apartado en el que se describirá un ejemplo de una aplicación que dispone de Identidad federada.
- Un apartado en el que se expondrán las conclusiones extraídas tras la realización del trabajo.
- Además constará de otros apartados como:
    - Glosario: Apartado en el que se incluirá una definición corta y aclaratoria de diferentes palabras o conceptos que se consideren.
    - Anexos: En este apartado se describirán apartados más concretos y que quedan un poco fuera del ámbito del estudio, como pueda ser: la instalación de un sistema o máquina virtual, producto o tecnología.
    - Bibliografía: Apartado en el que se expondrá todas las fuentes consultadas para poder realizar este estudio.

2 Estado del arte

2.1. Autenticación multifactor

La autenticación multifactor (MFA) [1] es un sistema de seguridad que requiere más de una forma de autenticación para verificar la legitimidad de una transacción. Combina dos o más credenciales independientes: lo que sabe el usuario (ej: contraseña), lo que tiene el usuario (ej: token de seguridad) y lo que es el usuario (ej: verificación biométrica).

El objetivo de la MFA es crear una defensa por capas y hacer que sea más difícil para una persona no autorizada acceder a un objetivo.

Algunos ejemplos o escenarios de MFA incluyen:

- Iniciar sesión en un sitio web y que se solicite introducir una contraseña adicional de una sola vez (OTP) que el servidor de autenticación del sitio web envía al teléfono o dirección de correo electrónico del solicitante.
- Deslizar una tarjeta e introducir un PIN (o una huella).
- Descargar un cliente de VPN con un certificado digital válido e iniciar sesión en el VPN antes de que sea concedido el acceso a una red interna.
- Colocar un token de hardware USB en un equipo de escritorio que genera una OTP y utilizar la contraseña de una sola vez para iniciar sesión en un cliente VPN.

Aunque hay muchos escenarios más, se va a hacer foco en los 3 sistemas más comunes que se ofrecen como segundo factor de autenticación [2] en los sitios web. Estos sistemas son: Mensajes de texto (SMS), Aplicaciones que ofrecen códigos de un sólo uso (OTP) y tokens hardware U2F (del inglés *Universal Second Factor*).

[2] MFA es al menos 2FA dado que se utilizan 2 factores diferentes, aunque se pueden incluir más factores [8]. Así pues, no se debe confundir con la autenticación en 2 pasos (2SA). La autenticación en 2 pasos utiliza el mismo tipo de factor durante el proceso de autenticación.

2.1.1. Tipos de sistemas multifactor

2.1.1.1. Mensajes de texto (SMS)

Uno de los sistemas que más se ofrece como segundo factor de autenticación a nivel web es el mensaje de texto, aunque siendo puristas con la definición de autenticación multifactor, no debería considerarse un factor dado que un SMS no es ni algo que el usuario sabe, ni algo que tiene, ni algo que es. Es sólo información que llega a un dispositivo móvil que este posee, siempre y cuando el operador los envíe a la persona correcta.

Así pues, el hecho de enviar códigos de un sólo uso (OTP) por mensaje de texto es, de por sí, problemático [2] porque los mensajes se envían a menudo en texto plano, y además, se puede encontrar software o servicios que los intercepten. A partir de ahí, un atacante remoto sólo necesita el número de teléfono de destino.

Hay varias formas de poder acceder a los mensajes [3]:

- Haciendo un duplicado de la tarjeta SIM.

Figura 1: Figura que muestra un mismo mensaje recibido en una tarjeta SIM duplicada.

- Interceptando los mensajes. Podría ser interceptando dicho mensaje una vez se ha conseguido acceso a la red SS7 (Sistema de señalización por canal Nº 7, del inglés *Signaling System Number 7*) o mediante una aplicación software maliciosa instalada en el dispositivo de la víctima.

Hay que tener en cuenta que el protocolo SS7 tiene unos 40 años de antigüedad [4] y no ha sufrido modificaciones, por tanto, no es seguro, es un protocolo vulnerable. Sin embargo, dicho protocolo se sigue utilizando en las redes de comunicación móviles actuales.

Hay una larga historia de exploits del protocolo SS7. En 2016 unos hackers demostraron a la CBS cómo podían grabar llamadas y la ubicación de Ted Leiu, miembro de la Cámara de Representantes de Estados Unidos. En 2017, unos ladrones aprovecharon la verificación en dos pasos de un banco para autorizar extracciones de dinero, e interceptaron los SMS enviados a los usuarios [5].

Por otro lado, un ejemplo de este segundo método de interceptación de mensajes es iBanking.Android [6], una aplicación diseñada para Android, que ha añadido una nueva funcionalidad donde utiliza software de seguridad falso para conseguir que el usuario instale el software malicioso. A continuación, roba mensajes SMS utilizados en la autenticación de dos factores.

2.1.1.2. Aplicaciones de autenticación OTP

Además de los SMS para recibir los códigos de un solo uso, también hay otros métodos para generar códigos, como son las aplicación de autenticación [7].

Lo ventajoso de este tipo de aplicaciones es que suelen ser sencillas y muy fácil utilizar. Generalmente, los pasos a seguir son:

- Instalar la aplicación de autenticación en el dispositivo móvil.
- Entrar en los ajustes de seguridad del servicio con el que quieres utilizar la aplicación como 2FA.
- El servicio mostrará un código QR que se puede escanear directamente en dicha aplicación.

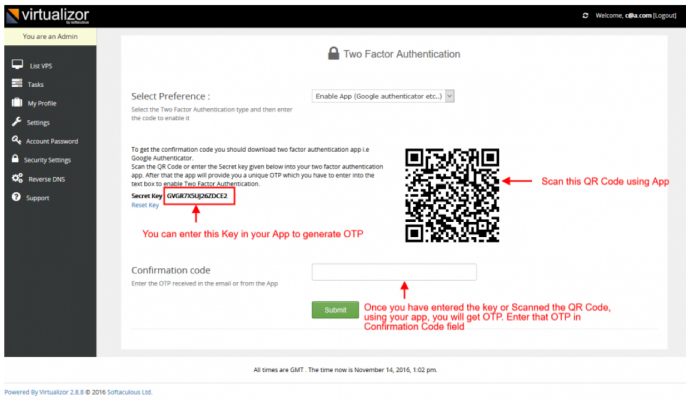

Figura 2: Figura que muestra un ejemplo de QR a escanear.

- A partir de ese momento, en la aplicación se podría visualizar el código que se debe introducir como 2FA. Dicha aplicación generará un nuevo código cada 30 segundos.

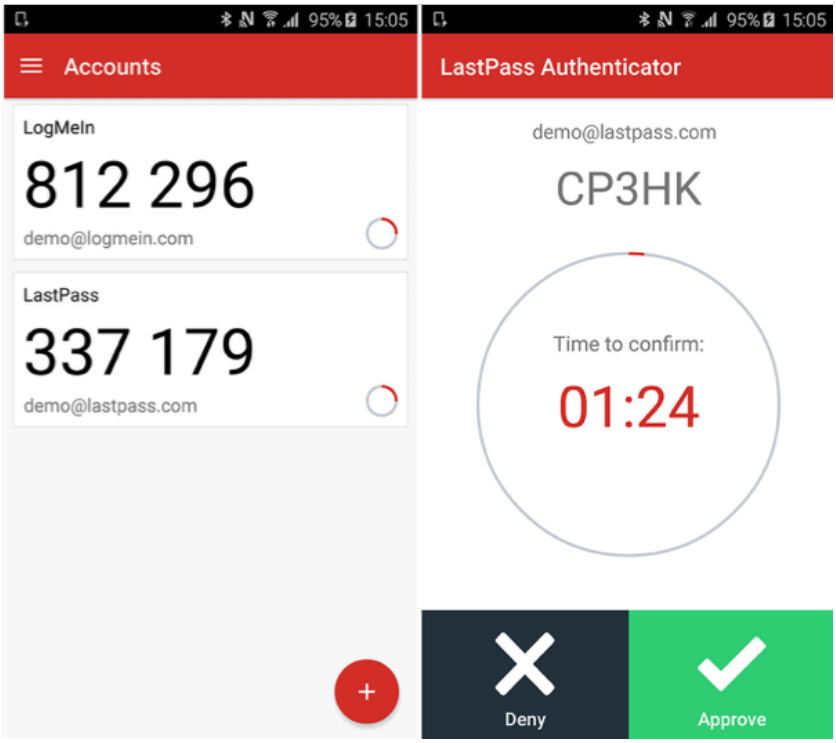

Figura 3: Figura que muestra un ejemplo del código que se muestra en la aplicación.

Los códigos se crean sobre la base de una clave (que sólo conoce el usuario y el servidor) y la hora actual, redondeada en 30 segundos. Ambos componentes son los mismos para ti y el servicio, por lo que los códigos se generan de forma sincronizada. Este algoritmo se conoce por sus siglas en inglés, OATH TOTP (contraseña de un solo uso basada en el tiempo), y es el más utilizado.

La alternativa es OATH HOPT (siglas en inglés de "contraseña de un solo uso basada en HMAC"). En lugar de la hora actual, este algoritmo utiliza un contador que aumenta en 1 por cada código recién creado. Pero no se utilizan mucho, ya que su uso complica la generación sincrónica de códigos por parte de la aplicación y el servicio. En pocas palabras, existe el riesgo de que el contador se averíe en el momento más inoportuna y tu contraseña de un solo uso no funcione.

Actualmente, existen múltiples aplicaciones multiplataforma que ofrecen la posibilidad de utilizarse como generador de códigos OTP, algunos ejemplos son: Google Authenticator, Microsoft Authenticator y Lastpass Authenticator.

2.1.1.3. Token hardware U2F

Otro método conocido [7], y quizás el más seguro que se utiliza como segundo factor de autenticación, es el token hardware U2F. Este método es el preferido por los especialistas de seguridad de la información, dado que para confirmar el inicio de sesión al servicio, tendrás que conectar el token U2F al dispositivo desde el cual estás iniciando sesión y pulsar en el botón token (algunos dispositivos solicitan un código o huella digital, pero esto es una función adicional).

En cuanto al funcionamiento del token (o llave), cuando se registra dicho token en un servicio, se generan un par de claves cifradas, privada y pública. La clave pública se almacena en el servidor y la privada en el token U2F. Por otro lado, la clave privada se utiliza para cifrar la confirmación de un inicio de sesión, que se pasa al servidor y puede ser descifrada utilizando la clave pública. Si un atacante intentara suplantarte y transferir una confirmación de inicio de sesión cifrada con la clave privada incorrecta, entonces, el descifrado con la clave pública sería erróneo y el servicio no concedería acceso a la cuenta.

Existen diferentes tipos de llaves U2F, según se deseen conectar al dispositivo. Algunas se conectan por USB (tipo A o tipo C), otras se conectan mediante NFC y existen incluso llaves que conectan a través de Bluetooth. Además, hay diferentes modelos de token, dependiendo de los protocolos que acepten. Los modelos básicos de token suelen admitir sólo U2F y cuestan entre 5 y 20 euros. Otros dispositivos más caros (entre 20 y 45 euros) también pueden operar como tarjetas inteligentes, generar contraseñas de un solo uso (incluidos los protocolos OATH TOTP y HOTP), generar y almacenar claves de cifrado PGP y utilizarse para acceder a los sistemas operativos.

Figura 4: Figura que muestra un ejemplo de tokens hardware de seguridad.

2.1.2. Soluciones de implantación

A la hora de querer implantar un segundo factor de autenticación debemos recordar que hay varios posibles factores, y por tanto, las herramientas y dispositivos necesarios para implantarlo variarán. Si queremos utilizar como segundo factor:

- El SMS: necesitaremos un servicio API para enviar los mensajes desde el servidor.
- La huella dactilar: necesitaremos un lector de huellas físico. Además, esto conllevará un sistema de almacenamiento que tiene que cumplir con ciertos estándares ya que se estará almacenando datos personales de nivel alto.
- Una tarjeta: necesitaremos un lector RFID y su correspondiente base de datos, en la que se asocie la tarjeta al usuarios.

2.1.2.1. Servicios de SMS

En este caso, suponiendo que deseamos implantar el SMS como segundo factor de autenticación, tal y como ya se ha comentado, será necesario disponer de un servicio API para poder enviar los mensajes de texto. Cada vez hay más servicios que ofrecen el envío de mensajes de texto mediante API. Algunos ejemplos son:

- SMS API [26]: Esta solución permite enviar SMS a través de API o través de interfaz. Permite además incluir enlaces acortados, adjuntar multimedia, programar las campañas, etc.
- Twilio [27]: En su web se puede leer: "realice el trabajo más rápido utilizando bibliotecas auxiliares, herramientas de supervisión y depuración, documentación en su idioma e incluso un entorno sin servidor para alojar su código. Con los números de teléfono disponibles en más de 30 países, el inventario de Twilio le permite elegir los números correctos con las capacidades adecuadas para su proyecto. Twilio maneja la lógica de las telecomunicaciones y las reglas específicas del operador para garantizar que su mensaje llegue a su destino".
- Esendex [28]: Tal y como se cita en su web, API SMS te permitirá enviar mensajes de manera segura y sencilla. Sólo tienes que integrar nuestro servicio en tu sistema, software o aplicación. Nuestra API SMS es muy fácil de usar y te ofrece todo lo que

necesitas para construir la mejor solución para tu negocio. Integra la API SMS de Esendex y envía automáticamente SMS desde aplicaciones, sitios web y software de forma inmediata y sencilla.

Evidentemente, elegir entre un proveedor u otro variará en función de diversos factores, como puede ser el coste del SMS, proveedor de cobertura del servicio, las posiblidades de integración que ofrezca, etc.

2.1.2.2. Servicios OTP

Si por el contrario, lo que deseamos es crear un sistema OTP, podemos utilizar, por ejemplo, FreeOTP [29]. FreeOTP permite implementar una configuración basada en contraseña única (OTP) que puede usar como un sistema 2FA.

- Esta aplicación usa sistema TOTP (OTP basado en tiempo).
- Es necesario crear un valor hexadecimal que será la clave de un usuario (como a1b2c3d4e5).
- Se debe generar una versión codificada Base32 de dicha clave.
- Una vez generado esto, se debe instalar la aplicación FreeOTP en el teléfono y especifique los dígitos totales de OTP (6 de manera predeterminada), y el tiempo que debe pasar para generar un código OTP nuevo en segundos (30 segundos).
- Se debe ingresar la clave del usuario y la versión codificada en Base32 junto con otros identificadores, como la dirección de correo electrónico para la identificación.
- Ahora, una vez ya se han añadido todos los parámetros a la aplicación, cada vez que se use FreeOTP, se recibirá un número válido durante 30 segundos, en base al tamaño y el intervalo seleccionado para la generación de códigos.
- Por último, se debe validar el número en el servidor a través de oathtool especificando los dígitos, el intervalo de generación de cógidos y el valor hexadecimal generado. Debe coincidir.

A continuación se muestra una captura de pantalla donde se ve cómo se configura FreeOTP y cómo se muestra una vez configurada:

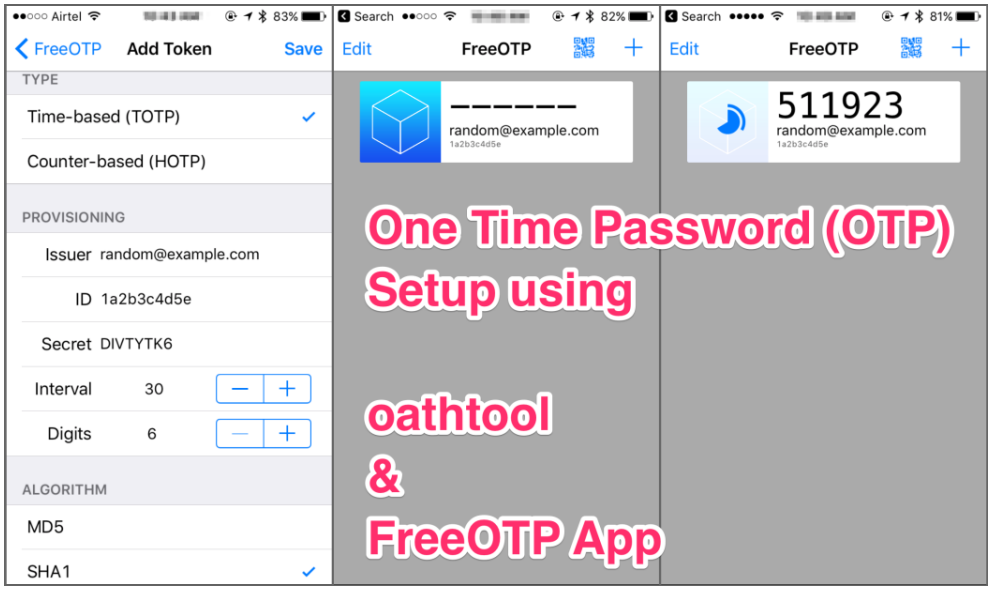

Figura 5: Figura que muestra la aplicación FreeOTP.

Para más detalles sobre los comandos que se deben ejecutar para realizar los pasos descritos anteriores, se recomienda visitar las referencias.

2.2. Identidad federada

La identidad federada [10] es una de las soluciones para abordar la gestión de identidad en los sistemas de información. El valor añadido adicional respecto a otras soluciones es la gestión de identidad interdependiente entre compañías, lo que se denomina *Federated Identity Management*.

Como cualquier solución de IdM (Gestión de Identidad, del inglés *Identity Management*), su objetivo es obtener una gestión de usuarios eficiente, la sincronización de los datos identificativos, gestión de acceso, servicios de agrupación, servicios de directorio, auditoría e informes, etc.

Mediante soluciones de Identidad Federada los individuos pueden emplear la misma identificación personal (típicamente usuario y contraseña) para identificarse en redes de diferentes departamentos o incluso empresas, para acceder a varias aplicaciones que proporcionan y hospedan diversas Organizaciones con las que mantienen una relación de negocios, etc. De este modo, las empresas comparten información sin compartir tecnologías de directorio, seguridad y autenticación, como requieren otras soluciones (metadirectorio, etc.). Para su funcionamiento es necesaria la utilización de estándares que definan meca-

nismos que permiten a las empresas compartir información entre dominios. El modelo es aplicable a un grupo de empresas o a una gran empresa con numerosas delegaciones y se basa en el "círculo de confianza" de estas, un concepto que identifica que un determinado usuario es conocido en una comunidad determinada y tiene acceso a servicios específicos.

En la siguiente figura 6 se ilustra el patrón de Identidad Federada cuando una aplicación cliente necesita acceder a un servicio que requiere autenticación. La autenticación se realiza mediante un proveedor de identidad (IdP) [3] que trabaja en combinación con un servicio de token de seguridad (STS) [4]. El IdP emite tokens de seguridad que proporcionan información sobre el usuario autenticado. Esta información, conocida como notificaciones, incluye la identidad del usuario y podría incluir también otra información como la pertenencia a roles y derechos de acceso más específicos.

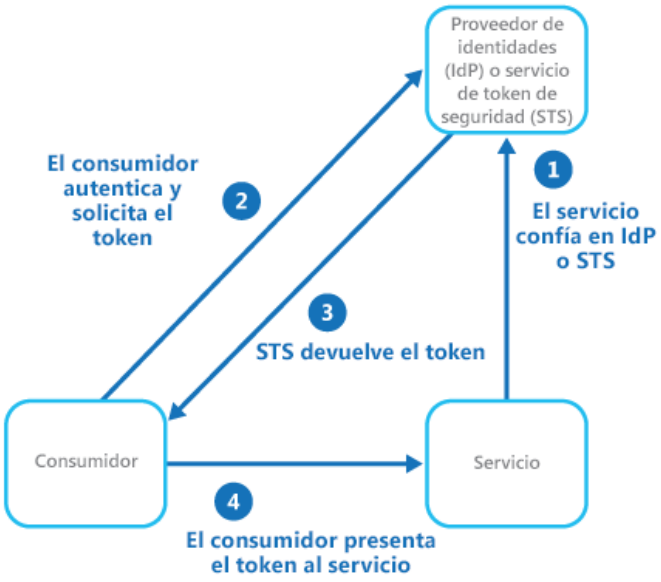

Figura 6: Figura que muestra el patrón de Identidad Federada.

Algunas de las empresas que han desarrollado software para Identidad Federada son:

- OASIS, que ofrece SAML (1.1), una especificación basada en XML que permite autenticación cruzada entre dominios.
- Microsoft e IBM, que proponen un mecanismo de seguridad de *Web Services* que incluye identidad.

[3] Un proveedor de identidad es la entidad dentro del sistema que se asegura de que el usuario realmente es quien dice ser -proporciona autenticación. También puede determinar a qué servicios, en su caso, el usuario está autorizado para acceder a través de varias entidades en el sistema. [13]

[4] Un servicio de token de seguridad es un servicio que emite, valida y renueva los tokens de seguridad.

- Por otro lado, se formó la alianza *Liberty Alliance Project* (a la que pertenece IBM desde octubre de 2004) para desarrollar estándares abiertos y neutrales para Federación de Identidad.

Como se ha comentado anteriormente, es necesario utilizar estándares para proporcionar Identidad Federada. A continuación se describirá con más detalle 3 de los estándares más conocidos y utilizados para proporcionarla, que son: SAML, 0Auth y OpenID Connect.

2.2.1. Estándares de lenguaje y protocolos

2.2.1.1. SAML

OASIS Security Assertion Markup Language (SAML) [12] es un estándar que utiliza una infraestructura basada en XML para describir e intercambiar información de seguridad entre entidades en línea. SAML 2.0 da soporte a:

- Inicio de sesión único (SSO, del inglés *Single Sign On*): Proporciona un protocolo y una gramática independiente del proveedor estándar para transferir información sobre un usuario de un servidor a otro, independientemente de los dominios DNS del servidor.

- Federación de identidad: Permite que los servicios asociados lleguen a un acuerdo y establezcan un identificador de nombre común para que el usuario comparta información sobre sí mismo más allá de los límites organizativos. Es decir [15], permite crear un círculo de confianza entre distintas organizaciones para que puedan emplear entre ellas una misma autentificación de usuario de forma nativa, sin necesidad de tener que compartir la misma tecnología de directorio, modelo de seguridad y mecanismos de autentificación o recurrir a pasarelas de conversión que les permita entenderse entre ellos.

- Gestión de identidad: Permite una gestión de identidad eficiente, puesto que permite la sincronización de datos identificativos entre las entidades federadas.

En la actualidad, aunque este estándar proporciona federación, gestión de identidad e inicio de sesión único, se ha acabado utilizando para proporcionar SSO dentro de una misma Organización. Este proceso es transparente para el usuario, quien sólo ha de hacer un

único inicio de sesión. A continuación se muestra un flujo de autenticación y autorización mediante SAML:

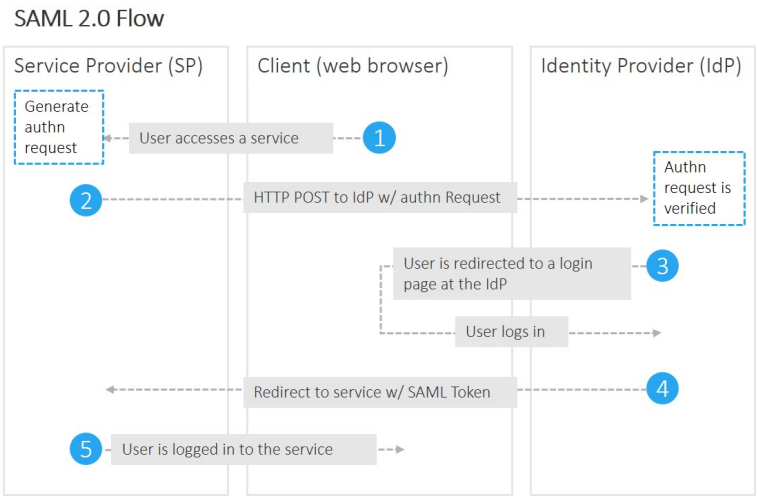

Figura 7: Figura que muestra el proceso de autenticación y autorización mediante SAML.

2.2.1.2. Oauth

OAuth (Open Authorisation) es un estándar un poco más nuevo que SAML, desarrollado conjuntamente por Google y Twitter a partir del 2006. Fue desarrollado, en parte, para compensar las deficiencias de SAML en las plataformas móviles, y está basado en JSON en lugar de XML.

A diferencia de SAML, OAuth es un protocolo de autorización, o más precisamente, de delegación de acceso; es decir, permite definir cómo un tercero va a acceder a los recursos propios. Es un protocolo de identificación abierto que utiliza un estándar Twitter, que facilita la autorización para el acceso a blogs. El sistema funciona de modo tal que en lugar de dar el nombre de usuario y la contraseña, se introduce la cuenta OAuth (oauth.net).

Es el protocolo que utiliza Twitter desde el año 2010 para ofrecer un modo rápido de identificarse en algunos servicios y otras redes sociales. Permite a los usuarios compartir sus recursos privados (fotos, vídeos, listas de contactos) almacenados en un sitio con cualquier otro sitio sin tener que introducir a mano su identificación digital (nombre de usuario y contraseña).

OAuth 2.0 es una versión revisada y simplificada de OAuth, que ya ha sido aprobada y oficialmente ha sido adoptada por grandes compañías. Respecto de la versión anterior,

presenta una mayor facilidad de implementación y una arquitectura más robusta que da soporte a mayor número de plataformas.

A continuación se muestra una imagen que describe el flujo para la autorización en 0Auth:

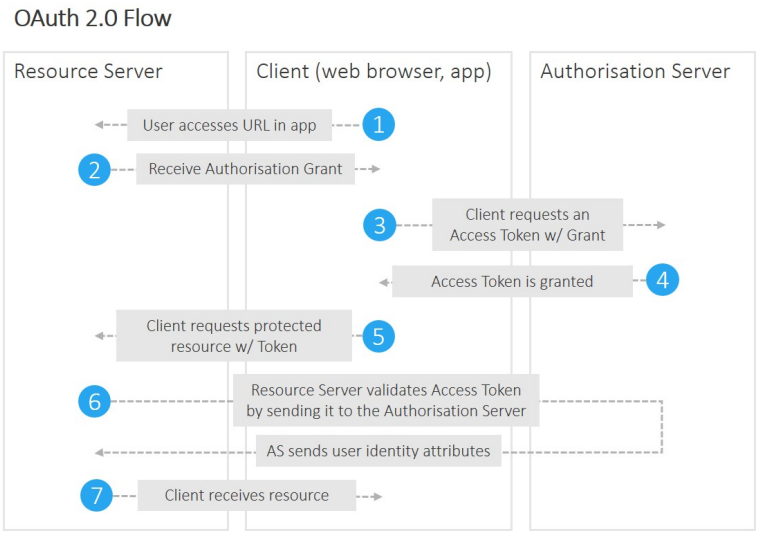

Figura 8: Figura que muestra el patrón de autorización mediante 0Auth.

2.2.1.3. OpenID Connect

OpenID Connect [16] es un estándar aún más nuevo, desarrollado en el 2014. Es una capa de identidad sobre el protocolo OAuth 2.0, el cual permite a los clientes verificar la identidad de un usuario basado en la autenticación realizada por un servidor de autorización, así como para obtener información de perfil del usuario utilizando un esquema REST. En términos técnicos, OpenID Connect especifica un RESTFul HTTP API, utilizando JSON como formato de datos.

OpenID Connect permite un gran variedad de clientes, incluyendo clientes Web, aplicaciones móviles y clientes basados en JavaScript. Los clientes pueden pedir y recibir información sobre los usuarios y sesiones. La especificación puede extenderse, soportando características adicionales como la encriptación de los datos, el discovery de proveedores OpenID y la gestión de sesiones.

Con Gigya [17], puede actuar como un proveedor de OpenID Connect (OP), autenticar a los usuarios mediante el protocolo de OpenID Connect (OIDC), o como una parte

dependiente (RP) que solicita la autorización de un OP.

A continuación se muestra una imagen que describe el flujo para la autenticación en OpenID mediante Gigya:

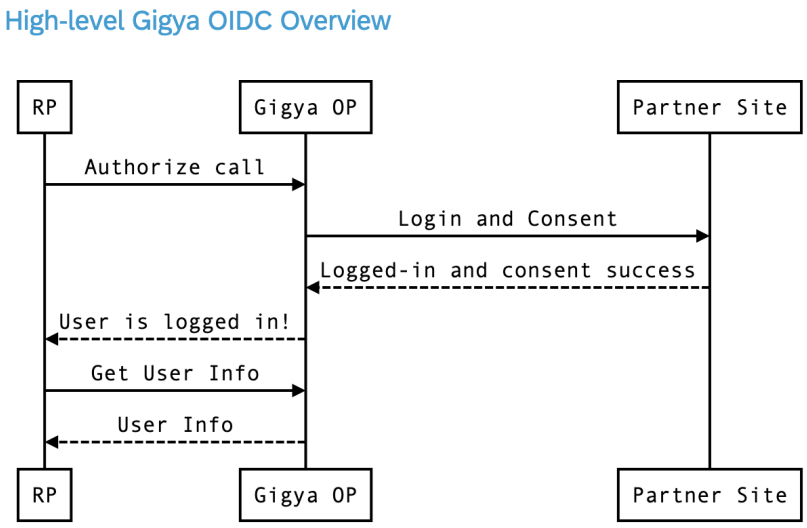

Figura 9: Figura que muestra el patrón de autenticación de OpenID mediante Gigya.

2.2.2. Regulación de atributos

Una vez descritos los estándares en cuanto a lenguajes y protocolos se refiere, no debemos olvidar que el fin de la Identidad federada es poder autenticarse en diferentes empresas o asociaciones que utilizan identidad federada, en definitiva, se trata de poder autenticarse en los diferentes proveedores de servicios que no guardan el usuario y la contraseña de usuarios que no pertenecen a su Organización. Es por ello que, también existe una regulación o política en cuanto a los atributos que debe emitir un proveedor de Identidad.

Así pues, por ejemplo, si deseamos utilizar la federación de identidad digital de segunda generación ofrecida por el Servicio de Federación de Identidades de RedIRIS (SIR2) [19], debemos tener en cuenta los siguientes atributos:

**La tabla que se muestra a continuación ha sido modificada parcialmente. Se puede encontrar la información exacta en el enlace proporcionado en referencias.*

Nombre	**Esquema**	**Descripción**	**Tipo**	**Usos**
commonName	person	Nombre y apellidos.	Recomendado	SIR2, eduGAIN
displayName	inetOrgPerson	Nombre y apellidos.	Recomendado	SIR2, eduGAIN
eduPerson Affiliation	eduPerson	Afiliación del sujeto en la organización (personal de administración, docente, investigador, estudiante,...).	Recomendado	SIR, SIR2, eduGAIN
eduPerson Entitlement	eduPerson	URI (URN o URL) indica los derechos para recursos.	Recomendado	SIR, SIR2, eduGAIN
eduPerson PrincipalName	eduPerson	Identificador único y persistente de un usuario.	Recomendado	SIR2, eduGAIN
eduPerson ScopedAffiliation	eduPerson	Afiliación de una persona con un dominio de organización.	Recomendado	SIR2, eduGAIN
eduPerson TargetedID	eduPerson	Atributo persistente, no reasignado, que preserva la privacidad y es compartido entre entidades que se coordinan entre sí.	Requerido	SIR, SIR2, eduGAIN
mail	inetOrgPerson	Dirección de correo electrónico.	Recomendado	SIR (opc), SIR2, eduGAIN
schacHome Organization	SCHAC	Organización a la que pertenece un sujeto, usando el dominio de la misma.	Recomendado	SIR, SIR2, eduGAIN
schacHome OrganizationType	SCHAC	Tipo de organización.	Recomendado	SIR2, eduGAIN
schacPersonal UniqueCode	SCHAC	URN del usuario: urn:schac: personalUniqueCode: es:rediris:sir: mbid:{¡algor¿}¡hash¿.	Recomendado	SIR, SIR2, eduGAIN
uid	inetOrgPerson	Un identificador corto de usuario, conocido en la organización a la que pertenece.	Recomendado	SIR (opc), SIR2, eduGAIN

Tabla 1: Tabla que muestra los atritubos para usar la identidad federada de Rediris.

A parte de SIR2, podemos encontrar otras políticas de liberación de atributos como son las del CSUC [20] o eduGain [21], que nos convendría tenerlas en cuenta si queremos federarnos en servicios referentes a la educación y/o la investigación. No se detallan las tablas de atributos dado que se puede encontrar toda la información en la web y con el ejemplo anterior ya es suficiente para alcanzar el objetivo del apartado, que no es otro que entender que existen políticas de liberación de atributos.

2.2.3. Soluciones de implantación

Implantar Identidad federada en una Organización es cada día más común, igual que el hecho de que un sitio web ofrezca autenticación multifactor. Prueba de ello, es la multitud de herramientas que están apareciendo para ayudar en la implatación, y muchas de estas herramientas son de carácter Open Source, es decir, que puedes obtenerlas libremente y sin coste.

Algunos de estos proyectos o herramientas que se pueden encontrar por Internet son, entre otros:

- Shibboleth [22]: El software Shibboleth es una de las soluciones de identidad federada más implementadas en el mundo, que conecta a los usuarios con aplicaciones tanto dentro como entre organizaciones. Todos los componentes de software del sistema Shibboleth son gratuitos y de código abierto. Proporciona capacidades de inicio de sesión único y permite que los sitios tomen decisiones informadas de autorización para el acceso individual de los recursos en línea protegidos de manera que preserven la privacidad.

- Simple SAML PHP [23]: SimpleSAMLphp es una aplicación escrita en PHP que se ocupa de la autenticación. El proyecto está dirigido por UNINETT y tiene una comunidad de usuarios y contribuyentes externos. El enfoque principal de SimpleSAMLphp es proporcionar soporte tanto para Proveedores de servicios como para Proveedores de Identidad. También es compatible con otros protocolos y marcos de identidad, como Shibboleth 1.3, A-Select, CAS, OpenID, WS-Federation o OAuth.

- Open Conext [24]: OpenConext es el software de código abierto desarrollado por SURFnet. La Red Nacional de Investigación y Educación (NREN), las organizaciones colaborativas u otras partes pueden usar OpenConext para crear su propia infraestructura de colaboración.

- ADAS SSO [25]: adAS es un Servidor de Autenticación Avanzado que realiza funciones de Proveedor de Identidad con una herramienta gráfica de configuración, integrada que facilita su administración, puesta en marcha y mantenimiento. Gracias a las capacidades técnicas de adAS es muy fácil ofrecer acceso a las aplicaciones que se incluyen en el Single Sign-On. Algunas de las aplicaciones más relevantes que permiten el acceso a través de adAS son: Moodle, Sakai, Drupal, WordPress, Google Apps, Microsoft Sharepoint y Liferay entre otras.

3 Ejemplo de Aplicación

En esta sección se va a describir una aplicación que se ha encontrado en Internet, y que utiliza Identidad federada. Además, de explicar los detalles más relevantes de código y de la propia aplicación, se describirá y mostrará el proceso de autenticación y autorización.

En este caso, la aplicación que se describirá proviene del manual Spring Security SAML Extension [30]. Tal y como podemos leer en dicho manual, en el Apartado 1.2: *The extension enables applications to act as a Service Provider in federations based on Web Single Sign-On and Single Logout profiles of SAML 2.0 protocol*, lo que se traduce en que, la aplicación es un Proveedor de Servicios con Single Sign On que utiliza SAML para comunicarse con el Proveedor de Identidad (IdP) y realizar la autenticación.

Por tanto, mediante este manual conseguiremos:

- Una interfaz web en el que se empezará el proceso de autenticación, es decir, un proveedor de servicios.
- Durante el proceso de autenticación y autorización se utilizará Identidad federada, aunque es transparente para el usuario. Para ello se utilizará un IDP.
- Una vez se haya trasmitido el Token de seguridad, podremos acceder a dicha aplicación.

La aplicación se debe descargar de: https://repo.spring.io/list/release/org/springframework/security/extensions/spring-security-saml/1.0.9.RELEASE/

3.1. Requisitos

Para montar dicha aplicación necesitaremos, por tanto, un entorno virtual formado por una máquina servidor. Podríamos añadir una máquina cliente, pero para este caso, no es estrictamente necesario.

El servidor debe contener:

- Máquina virtual con SO Debian 9 Strech
- Software Java JDK
- Apache Maven

Además de esto, necesitaremos un navegador web para poder visualizar el proceso de autencación.

La instalación de los requisitos del servidor, se describe con detalle en el Anexo A.1 Instalación de los requisitos de la aplicación.

3.2. Diseño

Lo que se pretende describir en este apartado es el proceso de un usuario que quiere acceder a una aplicación, proveedor de servicios en esta caso, y cómo éste se autentica mediante un IDP, cómo se comunica el SP y cómo viaja la solicitud al IDP y se genera la autorización para acceder a dicha aplicación.

3.2.1. Proveedor de servicios (SP)

El proveedor de servicio, tal y como ya se ha descrito, no deja de ser la interfaz web donde el usuario empieza el proceso de login para acceder a una aplicación u Organización. Una definición más técnica podría ser que un Proveedor de Servicios es una entidad del sistema que recibe y acepta (consume) aserciones de autenticación emitida por un proveedor de identidad (IdP).

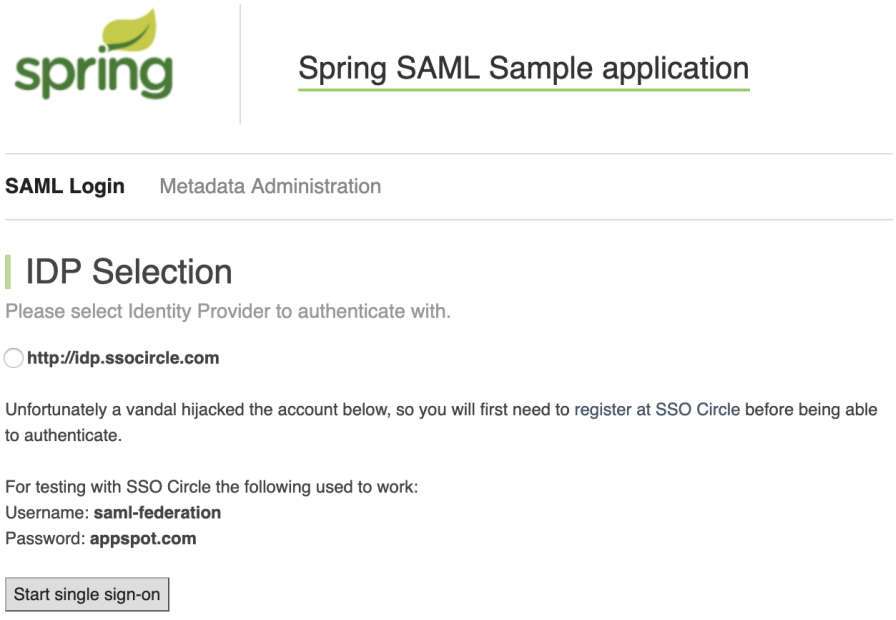

Figura 10: Figura que muestra la interfaz de la aplicación para login, el proveedor de servicios.

En la mayoría de casos, el proveedor de servicios ofrece directamente el panel de acceso o login donde se introducen las credenciales (principalmente usuario y contraseña) y se empieza el proceso de autenticación, aunque posteriormente se deba seleccionar el IdP a utilizar. En este caso, la aplicación nos obliga a elegir primero el IdP que queremos utilizar para autenticarnos. De esta manera, queda más evidente que se está utilizando un IdP externo.

Tras seleccionar el IdP, debemos pulsar sobre el botón *Start Single Sign-On* para inciar el proceso.

3.2.2. Proveedor de Identidad (IdP)

Lo siguiente que veremos, será una interfaz, donde ahora sí, hemos de introducir nuestro usuario y contraseña:

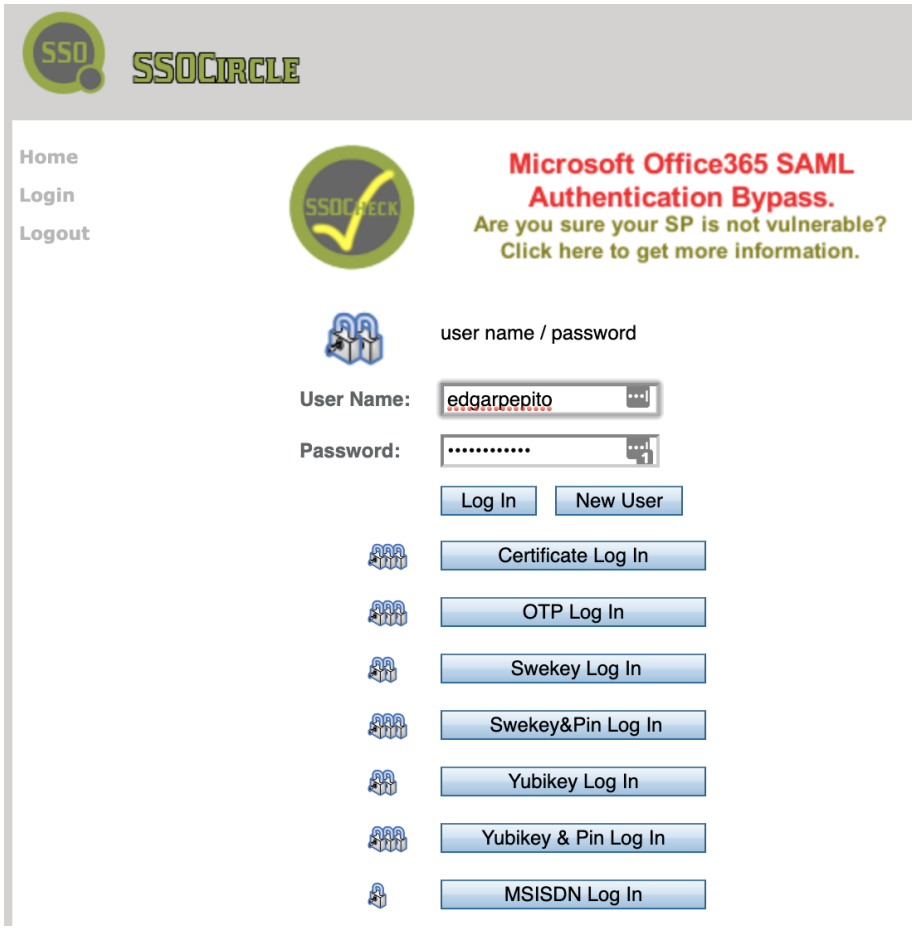

Figura 11: Figura que muestra la interfaz para realizar SSO.

Tal y como se ha indicado anteriormente, el IdP utilizado es externo, es decir, está publicado en Internet, por lo que, necesitaremos que el servidor tenga comunicación hacia el exterior, al menos hacia el recurso `https://idp.ssocircle.com/sso/` y puerto 443/TCP.

En este caso, no debemos olvidar que el proveedor de identidad (IdP) es una entidad del sistema que emite afirmaciones de autenticación. Esto quiere decir que, el IdP:

- Recibe una solicitud de autenticación de un usuario de confianza a través de un navegador web.
- Autentica al usuario.
- Responde al usuario de confianza con una aserción de autenticación SAML.

En el caso de esta aplicación, una vez autenticado en el IdP con usuario y contraseña, vemos como se hace una redirección hacia la web de la aplicación y se nos muestra lo siguiente:

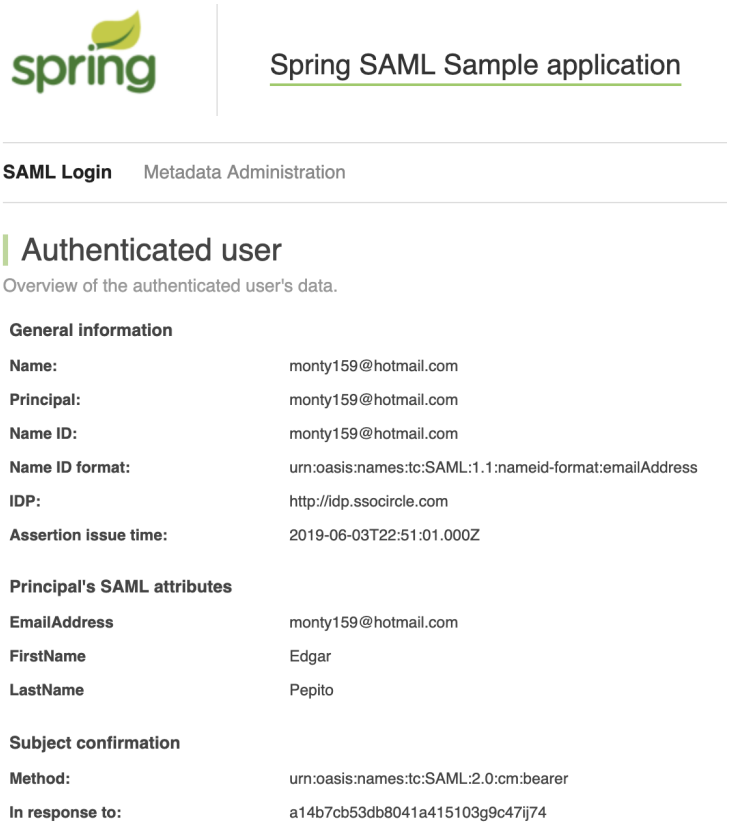

Figura 12: Figura que muestra la aplicación una vez se nos ha garantizado el acceso.

El objetivo de acceder a la aplicación ya se ha conseguido, pero ¿Cómo es posible que un Proveedor de servicios se fíe de un Proveedor de Identidad? ¿Qué es lo que sucede paralelamente mientras el usuario está pulsando en el botón para realizar el login? Aquí es donde entra en juego el estándar de metadatos de SAML. Es mediante dicha implementación como ambos proveedores establecen una línea de confianza e interoperabilidad.

Los metadatos de SAML no deja de ser un conjunto de datos establecidos en un documento XML que contiene la información necesaria para la interacción con proveedores de servicios o proveedores de identidad. El documento contiene, por ejemplo, URL de puntos finales, información sobre enlaces admitidos, identificadores y claves públicas.

Lo normal es generar un documento de metadatos en el propio proveedor de servicios, y posteriormente, enviarlo o subirlo a los proveedores de identidad con los que se desee habilitar el inicio de sesión.

En el caso de la aplicación sucede exactamente lo mismo que se ha descrito. Se generará el XML metadata en el SP, y posteriormente se subirá al IdP. Un ejemplo de código es el siguiente:

```
<bean id="metadataGeneratorFilter" class="org.springframework.security.saml.
    metadata.MetadataGeneratorFilter">
  <constructor-arg>
    <bean class="org.springframework.security.saml.metadata.MetadataGenerator">
      <property name="entityId" value="urn:test:Edgar:MAD"/>
      <property name="extendedMetadata">
        <bean class="org.springframework.security.saml.metadata.ExtendedMetadata">
          <property name="signMetadata" value="false"/>
          <property name="idpDiscoveryEnabled" value="true"/>
        </bean>
      </property>
    </bean>
  </constructor-arg>
</bean>
```

Cuadro 1: Código que muestra el SP Metadata.

Una vez compilamos la aplicación, recogemos el XML generado desde el enlace http://localhost:8080/spring-security-saml2-sample/saml/metadata y lo subimos al IdP. De tal manera que podremos observar en dicho IdP:

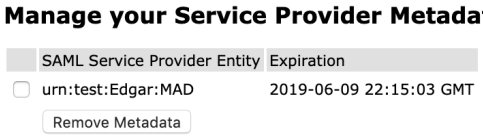

Figura 13: Figura que muestra el SP Metadata.

Tal y como se ha comentado, esta información no deja de ser un fichero XML. A continuación se muestra un ejemplo del fichero XML generado, que se muestra directamente en la propia aplicación:

Figura 14: Figura que muestra la aserción XML.

De manera similar, cada proveedor de identidad pondrá a su disposición sus propios metadatos para que pueda importarlos a su aplicación de proveedor de servicios. Es decir, debemos indicarle al SP a qué IdP debe dirigirse. Un ejemplo de código creado y extraído de la aplicación:

```
<bean id="metadata" class="org.springframework.security.saml.metadata.
    CachingMetadataManager">
  <constructor-arg> <list>
      <bean class="org.opensaml.saml2.metadata.provider.HTTPMetadataProvider">
        <constructor-arg>
          <value type="java.lang.String">https://idp.ssocircle.com/idp-meta.xml</
    value>
        </constructor-arg>
        <constructor-arg> <value type="int">5000</value> </constructor-arg>
        <property name="parserPool" ref="parserPool"/>
      </bean> </list> </constructor-arg> </bean>
```

Cuadro 2: Código que muestra el IdP Metadata.

De esta manera, podremos observar en el IdP los múltiples SP desde donde se está realizando login y existe federación:

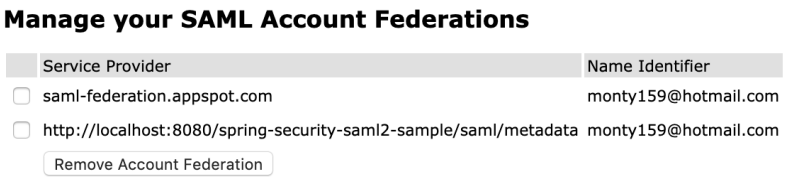

Figura 15: Figura que muestra los diferentes SP.

Adicionalmente, queda señalar el proceso de logout. Tal y como se puede ver en la figura 14, se puede realizar un Local Logout o Global logout. En este caso, es conveniente destacarlo, dado que sucederá una cosa diferente dependiendo del tipo de logout que seleccionemos:

- En caso de que seleccionemos el Logout local, veremos que cuando volvamos a realizar login en la aplicación, no será necesario volver a introducir las credenciales en el IdP.
- Por el contrario, si seleccionemos Global Logout, la próxima vez que queramos hacer login en la aplicación, volveremos a tener que introducir las credenciales en el IdP, como si del primer inicio de la aplicación se tratara. Sería idéntico al proceso descrito durante este apartado.

Una vez realizado el logout, la aplicación nos mostrará la siguiente notificación:

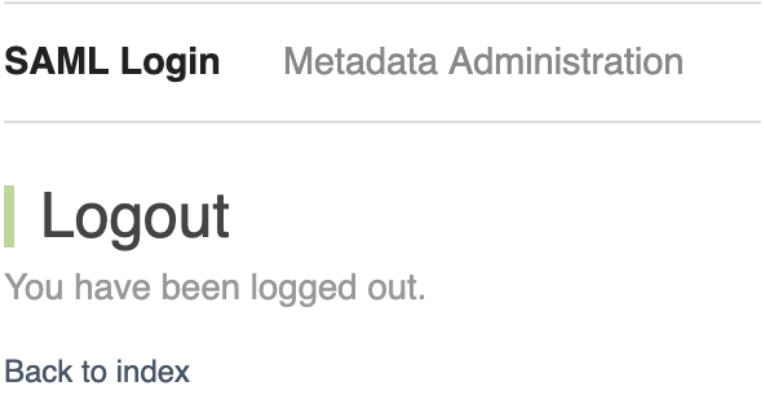

Figura 16: Figura que muestra que se ha realizado logout en la aplicación.

Por último, vemos que la aplicación también permite autenticación multifactor, tal y como se muestra en la figura 11. En este punto, sólo se quiere destacar un aspecto, y es que, el encargado de iniciar el proceso multifactor no es otro que el proveedor de Identidad, no entra dentro de las funciones del proveedor de servicios.

4 Conclusiones

El objetivo de este trabajo de fin de máster era implantar identidad federada y doble factor de autenticación en un entorno local para comprender sus múltiples beneficios. Aún así, también hay varios puntos no tan positivos que se han extraído durante el proceso, y son necesarios comentar.

Referente al doble factor de autenticación, a nivel de usuario, se debe remarcar 2 puntos importantes:

- A día de hoy, existen multitud de aplicaciones y sitios web que todavía no ofrecen la posibilidad de utilizar doble factor de autenticación, dado que no lo han implantado, por lo que, este hecho supone un riego. Además, aunque se entiende que es recomendable ofrecer dicha posibilidad, no es obligatorio, así pues, los sitios web podrán no implantarlo.

- Por otro lado, la seguridad siempre pone en entredicho la usabilidad. El hecho de utilizar un segundo factor de autenticación implica que dependemos de disponer de éste en todo momento. En el caso de utilizar una llave de seguridad, por ejemplo, implica que si olvidamos la llave o la perdemos no podremos acceder a los servicios. De igual manera, en el caso de utilizar una aplicación OTP o el SMS como segundos factores, implica que debemos disponer del dispositivo móvil. Así pues, si nos olvidamos el dispositivo o lo perdemos significará que tampoco podremos acceder a los servicios.

En definitiva, en cuanto al multifactor se refiere, es muy recomendable habilitarlo en todos los servicios que esté disponible, pero es importante disponer de diversos factores de autenticación, por si no disponemos de un factor concreto en un momento determinado, no perder el acceso a los servicios.

Por otro lado, en cuanto a la identidad federada, se ha visto que su implantación es cada vez más recomendable, dado que permite reducir costes significativos:

- El hecho de que un proveedor de servicios pueda confiar en un IdP permite que las diferentes Organizaciones dispongan de Bases de Datos mucho más pequeñas, la cantidad de datos a almacenar es inmensamente más pequeña.

- El proceso de autenticación es estándar, por lo que, si una Organización se federa,

permitirá que usuarios de otra Organización puedan acceder ha dichos servicios sin necesidad de darse de alta.

- Es un proceso de autenticación y autorización seguro. Una Organización no debe centrarse en cómo almacenar información de otras Organizaciones, o si debe cumplir con cierta normativa, ya que se fiará del estándar y, por tanto, del IdP. Además, en el caso de que sufriera una brecha de seguridad, la Identidad de los usuarios externos (entendiendo usuarios de otras Organizaciones) no se vería comprometida.

5 Glosario

Autenticación: Determinar que los usuarios son quienes dicen ser.

Autorización: Determinar si los usuarios tienen derecho a acceder a ciertos sistemas o contenidos.

2FA: del inglés *Second Factor Autentication*. En español a este término hace referencia a: Segundo factor de autenticaión.

MFA: del inglés *Multi Factor Autentication*. En español a este término hace referencia a: Autenticación multifactor. MFA es al menos 2FA dado que se utilizan 2 factores diferentes, aunque se pueden incluir más factores.

2SA: Autenticación en 2 pasos. La autenticación en 2 pasos utiliza el mismo tipo de factor durante el proceso de autenticación.

SMS: del inglés *Short Message Service*. En español a este término hace referencia a: Mensaje de texto.

OTP: del inglés *One Time Password*. En español a este término hace referencia a: Contraseña de un solo uso.

U2F: del inglés *Universal Second factor*. En español a este término hace referencia al estándar de la verificación en dos pasos por hardware.

SSO: del inglés *Single Sign On*. En español a este término hace referencia a: Inicio de sesión único.

IdP: Un proveedor de identidad es la entidad dentro del sistema que se asegura de que el usuario realmente es quien dice ser -proporciona autenticación. También puede determinar a qué servicios, en su caso, el usuario está autorizado para acceder a través de varias entidades en el sistema. [13].

STS: Un servicio de token de seguridad es un servicio que emite, valida y renueva los tokens de seguridad.

A Anexo

A.1. Instalación de los requisitos de la aplicación

El sistema operativo utilizado para las pruebas ha sido una máquina virtual con Sistema Operativo Debian 9 Strech. A continuación se especificarán los comandos necesarios para preparar el entorno y poder ejecutar las aplicaciones

Como se ha visto en los requisitos, es necesario instalar Java, para ello en Debian, lo hacemos mediante el siguiente comando en terminal:

```
1 user@mv:/# aptitude install openjdk-8-jdk
```

Cuadro 3: Comando utilizado en Debian para instalar java desde terminal.

Si queremos comprobar la versión de Java JDK instalada en el activo:

```
1 user@mv:/# mvn --version
2 Java version: 1.8.0_212, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-8-
    openjdk-amd64/jre
3 user@mv:/# find / -name java -type f -executable
4 /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
```

Cuadro 4: Comandos utilizados para visualizar la versión de java instalada en el activo desde terminal.

Otro requisito imprescindible es Apache Maven. A continuación se listan los comandos necesarios para instalarlo. Lo primero es descargarlo y descomprimirlo:

```
1 user@mv:/# cd /usr/local
2 user@mv:/# wget http://www-eu.apache.org/dist/maven/maven-3/3.6.0/binaries/apache-
    maven-3.6.0-bin.tar.gz
3 user@mv:/# tar xzf apache-maven-3.6.0-bin.tar.gz
4 user@mv:/# ln -s apache-maven-3.6.0 apache-maven
```

Cuadro 5: Comandos de terminal utilizados para descargar e instalar Apache Maven.

Una vez descargado Apache Maven, se deben especificar varias variables de entorno en la configuración del mismo para su correcto funcionamiento. En concreto, se debe modificar el archivo */etc/profile.d/apache-maven.sh*:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
export M2_HOME=/usr/local/apache-maven
export MAVEN_HOME=/usr/local/apache-maven
export PATH=${M2_HOME}/bin:${PATH}
```

Cuadro 6: Contendido del fichero apache-maven.sh.

Seguidamente se deben refrescar las variables de entorno:

```
user@mv:/# source /etc/profile.d/apache-maven.sh
```

Cuadro 7: Comando para cargar las nuevas variables de entorno para Apache Maven.

Por último queda confirmar que Apache Maven se ha instalado correctamente. Para ello, debemos ejecutar el siguiente comando y ver el resultado:

```
user@mv:/# mvn --version
Apache Maven 3.6.0 (97c98ec64a1fdfee7767ce5ffb20918da4f719f3; 2018-10-24T20
    :41:47+02:00)
Maven home: /usr/local/apache-maven
Java version: 1.8.0_212, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-8-
    openjdk-amd64/jre
Default locale: es_ES, platform encoding: UTF-8
OS name: "linux", version: "4.9.0-9-amd64", arch: "amd64", family: "unix"
```

Cuadro 8: Comando para comprobar que Apache Maven ha sido correctamente instalado.

Puede obtener más información sobre la instalación de Apache Maven en la siguiente referencia: [31].

Referencias

[1] Autenticación multifactor (MFA):
https://searchdatacenter.techtarget.com/es/definicion/Autenticacion-multifactor-MFA
Acceso: 12 febrero 2019.

[2] La autenticación multifactor no es una panacea de seguridad:
https://searchdatacenter.techtarget.com/es/consejo/La-autenticacion-multifactor-no-es-una-panacea-de-seguridad
Acceso: 12 marzo 2019.

[3] It's Time to Stop Using SMS and 2FA Apps for Two-Factor Authentication:
https://www.makeuseof.com/tag/two-factor-authentication-sms-apps/
Acceso: 12 marzo 2019.

[4] Roban datos de clientes de un operador por culpa del protocolo de red SS7:
https://www.adslzone.net/2018/05/31/roban-datos-operador-ss7/
Acceso: 12 marzo 2019.

[5] Por qué la autentificación de doble factor no basta:
https://www.kaspersky.es/blog/ss7-attack-intercepts-sms/12962/
Acceso: 12 marzo 2019.

[6] Cómo detectar y mitigar técnicas avanzadas de evasión de malware:
https://searchdatacenter.techtarget.com/es/consejo/Como-detectar-y-mitigar-tecnicas-de-evasion-avanzadas-de-malware
Acceso: 12 marzo 2019.

[7] La autentificación de doble factor en SMS es insegura, ¿qué alternativas hay?:
https://www.kaspersky.es/blog/2fa-practical-guide/17187/
Acceso: 12 marzo 2019.

[8] Autenticación Multifactor:
https://blog.isecauditors.com/2016/07/autenticacion-multi-factor-la-nueva-apuesta-segur
2.html
Acceso: 12 marzo 2019.

[9] Por qué no debes confiar en la autenticación en dos pasos a través de SMS:
https://www.genbeta.com/seguridad/por-que-no-debes-confiar-en-la-autenticacion-en-dos-p
Acceso: 12 marzo 2019.

[10] Identidad federada:
https://es.wikipedia.org/wiki/Identidad_federada
Acceso: 12 marzo 2019.

[11] Patrón de Identidad Federada:
https://docs.microsoft.com/es-es/azure/architecture/patterns/federated-identity
Acceso: 12 marzo 2019.

[12] Qué es SAML 2.0:
https://www.ibm.com/support/knowledgecenter/es/SSQL82_9.5.0/com.ibm.bigfix.doc/Platform/Config/c_what_is_saml_2_0.html
Acceso: 12 marzo 2019.

[13] SAML: Qué es, para qué se usa, cómo funciona:
https://cioperu.pe/articulo/24726/saml-que-es-para-que-se-usa-como-funciona/?p=2
Acceso: 12 marzo 2019.

[14] The Difference Between SAML 2.0 and OAuth 2.0:
https://www.ubisecure.com/uncategorized/difference-between-saml-and-oauth/
Acceso: 12 marzo 2019.

[15] Gestión de Identidad en la Nube: Un caso usando SAML:
http://conaiisi.unsl.edu.ar/2013/228-465-2-DR.pdf
Acceso: 12 marzo 2019.

[16] OpenID Connect:
https://es.wikipedia.org/wiki/OpenID_Connect
Acceso: 12 marzo 2019.

[17] OpenID Connect:
https://developers.gigya.com/display/GD/OpenID+Connect
Acceso: 12 marzo 2019.

[18] OpenID Connect:
https://openid.net/connect/
Acceso: 12 marzo 2019.

[19] Política de liberació de atributos SIR2:
http://www.rediris.es/sir2/federacion/atributos/
Acceso: 23 marzo 2019.

[20] Política de liberació de atributos CSUC:
https://www.csuc.cat/es/investigacion/federacion-de-identidades-unificat/politica-de-liberacion-de-atributos
Acceso: 23 marzo 2019.

[21] Política de liberació de atributos eduGain:
https://technical.edugain.org/entities
Acceso: 23 marzo 2019.

[22] Shibboleth:
https://www.shibboleth.net/
Acceso: 20 abril 2019.

[23] Simple SAML:
https://simplesamlphp.org/
Acceso: 20 abril 2019.

[24] Open Conext:
https://openconext.org/
Acceso: 20 abril 2019.

[25] ADAS SSO:
http://www.adas-sso.com/es/
Acceso: 20 abril 2019.

[26] SMS API:
https://www.smsapi.com/es
Acceso: 20 abril 2019.

[27] Twilio:

https://www.twilio.com/sms

Acceso: 20 abril 2019.

[28] Esendex:

https://www.esendex.es/api-sms

Acceso: 20 abril 2019.

[29] FreeOTP:

https://exain.wordpress.com/2017/08/17/create-one-time-password-otp-for-your-applicatio

Acceso: 20 abril 2019.

[30] Get started con SAML:

https://docs.spring.io/spring-security-saml/docs/1.0.x/reference/html/chapter-introduction.html

Acceso: 10 mayo 2019.

[31] Install Apache Maven:

https://tecadmin.net/install-apache-maven-on-debian/

Acceso: 10 mayo 2019.