Resumen

Cada vez está tomando más importancia la web, y actualmente están apareciendo multitud de aplicaciones, ya sea por parte de empresas o de instituciones, como Universidades, etc. Estas aplicaciones normalmente requieren autenticación, y la mayoría de ellas basan dicha autenticación en bases de datos locales.

Con el crecimiento de las aplicaciones, crece el número de usuarios de estas, y por parte del usuario, crece el número de cuentas creadas para diferentes aplicaciones. Para paliar este problema, nace el Single Sing On(SSO), que proporciona una única cuenta, y un único punto de autenticación, para diferentes aplicaciones web de una misma entidad, por ejemplo la Universidad de Sevilla.

El siguiente paso natural, es el uso de aplicaciones de otras entidades, y aquí es donde radica la importancia de la **federación de identidad**.

La federación de identidad consiste en que una serie de entidades **confían** en otras para la autenticación de los usuarios. Es decir, que una aplicación de una entidad, acepta usuarios de otra entidad. Además estos usuarios se autenticarán en su entidad, por lo que la gestión de usuarios, contraseñas, y atributos, queda delegada a cada entidad. Por tanto el usuario final tiene acceso a todas las aplicaciones federadas, de todas las entidades que conforman la federación de identidad.

Esto facilita enormemente la gestión de usuarios, por parte de las entidades, puesto que tan solo tienen que gestionar sus propios usuarios, y prestan servicio a usuarios de otras entidades, gracias a que la autenticación es delegada.

La idea principal de este proyecto es llevar las facilidades que proporciona la federación de identidad fuera del ámbito de la web, más concretamente al ámbito del **SSH** (Secure SHell).

Para el caso del SSH, si un usuario tiene acceso a diferentes máquinas, tendrá diferentes cuentas, y diferentes contraseñas que recordar, almacenar y gestionar, con la problemática que eso conlleva. Además sus contraseñas estarán en máquinas que no tiene por qué controlar, por lo que si se compromete alguna de estas máquinas, estará comprometida su contraseña.

Utilizando SSH sobre la federación de identidad, se pueden eliminar estos problemas e incrementar la comodidad, tanto por parte de los usuarios, como por parte de los administradores. Para poder acceder por SSH, un usuario tendría que autenticarse en la federación, y una vez autenticado, podrá entrar en todas las máquinas que ofrezcan el servicio de SSH federado, sin necesidad de poner contraseña, basándose en el mecanismo de clave pública y clave privada, y estando el servidor en cualquier entidad de la federación.

Así pues, por parte del usuario, se tiene acceso a diferentes máquinas necesitando recordar y gestionar una sola contraseña. Además esta contraseña nunca se entrega a un servidor extraño, sólo a la entidad de la cual procedes (a través de una páginas web segura), y en la cual confías puesto que es la encargada de gestionar tu identidad.

Y por parte del administrador, se puede delegar la gestión de usuarios, confiando en la federación. Automatizando la creación y destrucción de cuentas, y sin necesidad de proporcionar ninguna contraseña.

Índice general

Capítulo 1

Introducción

1.1. Objetivos

Las facilidades que ofrece la federación de identidad son más que evidentes, y podría ser interesante en muchos casos, tener estas facilidades para otros servicios que requieran autenticación.

El principal objetivo de este proyecto es llevar las facilidades de la gestión de identidad del ámbito de la web a otros servicios, como por ejemplo el SSH. En este proyecto nos hemos centrado en integrar la federación de identidad con el acceso por SSH, y puede servir como prueba de concepto a la hora de llevar la autenticación por federación de identidad a servicios diferentes de la web.

Las características más importantes de la federación de identidad, que nos serán útiles en el SSH son:

1. **Acceso a recursos de otras entidades**: La base de la federación de identidad es poder acceder a recursos de otra entidad con la misma cuenta con la que accedes a los recursos o servicios de tu propia entidad.
2. **Gestión de identidad distribuida**: Al encargarse cada entidad de la federación de sus propios usuarios, y basándonos en las relaciones de confianza de la federación, se puede dar servicio a un mayor número

de usuarios gestionando tan solo una pequeña cantidad de ellos. Esto puede crear algún tipo de duda, puesto que se pierde el control sobre los usuarios, pero no hay que olvidar que la federación es una red de confianza, donde cada entidad debe confiar en las demás, y para ello hay mecanismos seguros, como por ejemplo los certificados.

3. **Unicidad de contraseña**: La federación de identidad nos brinda la posibilidad de acceder a diferentes servicios, que requieren autenticación, con la misma cuenta y la misma contraseña, y sin necesidad de replicar esta en los diferentes servicios, sino estando en tu propia entidad, incrementando así la seguridad de la misma, y la comodidad a la hora de cambiar de contraseña, o de nombre de usuario.

4. **Login único**: También se busca implementar el Single Sing On(SSO) para el SSH sobre federación, de tal forma que un usuario sólo tenga que autenticarse una vez, y a partir de ahí, tener acceso, sin necesidad de introducir ningún tipo de contraseña, a todos los servidores SSH disponibles.

Por otra parte, hemos elegido llevar la federación al servicio SSH porque es ampliamente utilizado, además de que ofrece una gran potencia y versatilidad, abriendo así la puerta a la utilización de otros servicios de forma fácil.

Por ejemplo, en el ambiente académico, puede ser interesante dar acceso a un servidor SSH a todos los alumnos de Informática, bien sea para que utilicen un supercomputador, o para que tengan una cuenta dónde hacer las practicas. Dentro del objetivo de este proyecto entraría delegar la gestión de estos usuarios a la federación, facilitar así el proceso, así como por parte del alumno, como por parte del administrador de las máquinas.

1.2. Caso de uso

En el siguiente caso de uso se muestra el funcionamiento básico del sistema, así como una serie de detalles que serán explicados detalladamente en la sección “Implementación y despliegue” [4].

Caso de uso del proyecto SSH sobre federación de identidad:

- **Descripción**:

  La federación está pensada para aplicaciones webs, pero sería interesante poder utilizar estos mecanismos para aplicaciones que autentican de otra manera diferente.

  En el caso del ssh federado intentamos llevar el concepto de hacer login una sola vez, y en tu entidad, al acceso por ssh. Buscando poder acceder por ssh a diferentes máquinas sin tener que escribir usuario y contraseña, una vez nos hayamos autenticado. A través de ssh se pueden hacer muchas más cosas, como por ejemplo túneles ssh, port-forwarding, etc.

- **Proceso de Autenticación**:

  1. Se accede a una página especifica, protegida tras un SP.
  2. El usuario se autentica en la federación, y puede ver la página.
  3. Esta aplicación web intentará conseguir la clave RSA publica del usuario a través de los datos que manda la federación.
  4. Una vez autenticado en esa aplicación web, el usuario puede acceder a las cuentas ssh federadas de las que disponga sin tener que introducir password.

  Se puede ver un esquema del funcionamiento de este proceso en la figura 1.1

- **Proceso de Autenticación alternativo**: La clave publica que recibe la aplicación a través de la federación será la de la máquina habitual del usuario. En caso de estar utilizando otra máquina es posible utilizar otra clave temporalmente.

  1. Se accede a una página especifica, protegida tras un SP.
  2. El usuario se autentica en la federación, y puede ver la página.
  3. En la aplicación web se introduce la clave publica RSA temporal, para esta sesión.

4. Una vez autenticado en esa aplicación web, el usuario puede acceder a las cuentas ssh federadas de las que disponga sin tener que introducir password.

- **Implementación**: Para la implementación se ha optado por utilizar el mecanismo de acceso por clave publica-privada que nos ofrece el mismo protocolo ssh. Este mecanismo es el siguiente: El usuario crea un par de claves, para la máquina en la que se encuentra (ssh-keygen).

  Para dar acceso remoto sólo necesitamos conocer la clave publica ($HOME/.ssh/id_rsa.pub).

  Para poder acceder es necesario que el usuario disponga de su par privado.

  El servidor openssh, mira en el directorio personal del usuario, y busca en el archivo authorized_keys ($HOME/.ssh/authorized_keys), antes de pedir password. Si encuentra alguna clave, intenta la autenticación por RSA, que es automática, sin petición de password. Por lo tanto nuestro objetivo es utilizar este servicio, pero en lugar de mirar en un archivo local, preguntaremos a un servidor remoto.

- **Requisitos**:

  Para poder acceder a cualquier máquina remota por ssh, en el servidor se debe poner el sshd parcheado. Además se debe crear una cuenta de usuario. Es recomendable el deshabilitar la posibilidad de cambiar el password, puesto que si se puede cambiar el password de la cuenta, es posible acceder a esta sin pasar por la federación. También es conveniente no permitir la creación, o borrar, los ficheros dentro de .ssh del home del usuario, por la misma razón que lo anterior.

  También será necesario definir un schema para la federación, que añada el campo ssh_rsa_public_key, si queremos que el acceso sea lo más automatizado posible.

- **Temas a discutir**:

  1. Timeout para hacer login.
  2. Schema a usar para guardar la clave publica en LDAP.
  3. Diferentes roles como otros atributos, para dar permiso a unos y a otros nos.

4. Posibilidad de cambiar password, y permisos en la cuenta previamente creada.

1.3. Antecedentes

(feide, parche opensshldap)

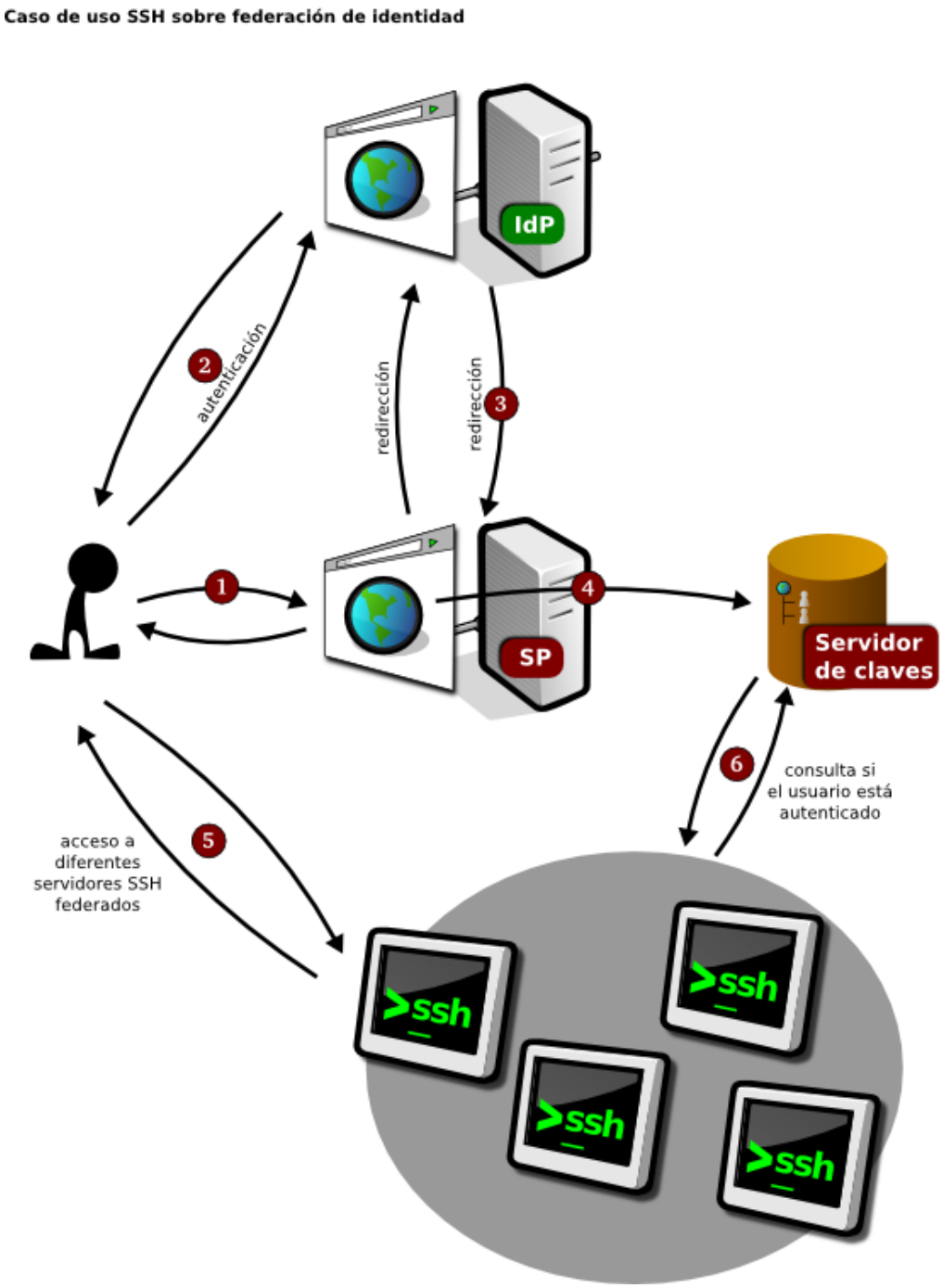

Figura 1.1: Caso de Uso

Capítulo 2

Análisis del problema

2.1. Necesidad de federación

2.2. Uso de la federación fuera de la web

2.3. La importancia del ssh

(túneles, importación X ...)

Capítulo 3

Solución propuesta

3.1. Parche para openssh (proceso de autenticación sshd)

3.2. Servidor de claves, flexibilidad, LDAP, python...

3.3. Aplicación de login.

3.4. Ejemplo aplicación creación de cuentas.

Capítulo 4

Implementación y despliegue

4.1. Código del openssh, (mínimas variaciones)

4.2. Código de las aplicaciones federadas, (ssh, useradd)

4.3. Necesidades para montar la plataforma

Capítulo 5

Conclusiones

Capítulo 6

Conclusiones

6.1. Por qué es importante, y por qué soy el mejor

Capítulo 7

Anexos

7.1. Aplicación del parche sobre openssh

7.2. Despliegue del servidor ssh con el parche

7.3. Seguridad mínima (permisos, cambiar passwd, etc)

7.4. Despliegue de la aplicación federada

7.5. Schemas ldap utilizados, dónde encontrarlos, cómo instalarlos en openldap