

SSO (Single Sign On)



José Miguel Aguado Coca



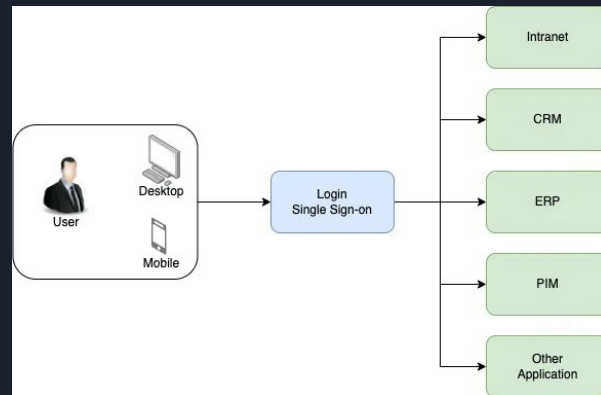
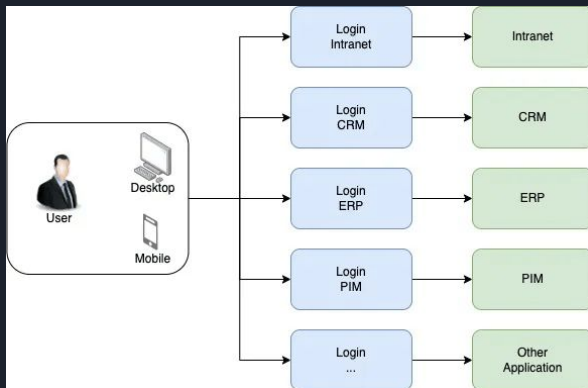
¿De qué voy a hablar?

- 1) Motivación e introducción a SSO.
- 2) Clasificación y Desarrollo de las distintas Arquitecturas SSO.
- 3) Protocolos más usados en el ámbito SSO.
- 4) Beneficios, Problemas y Desafíos de SSO.

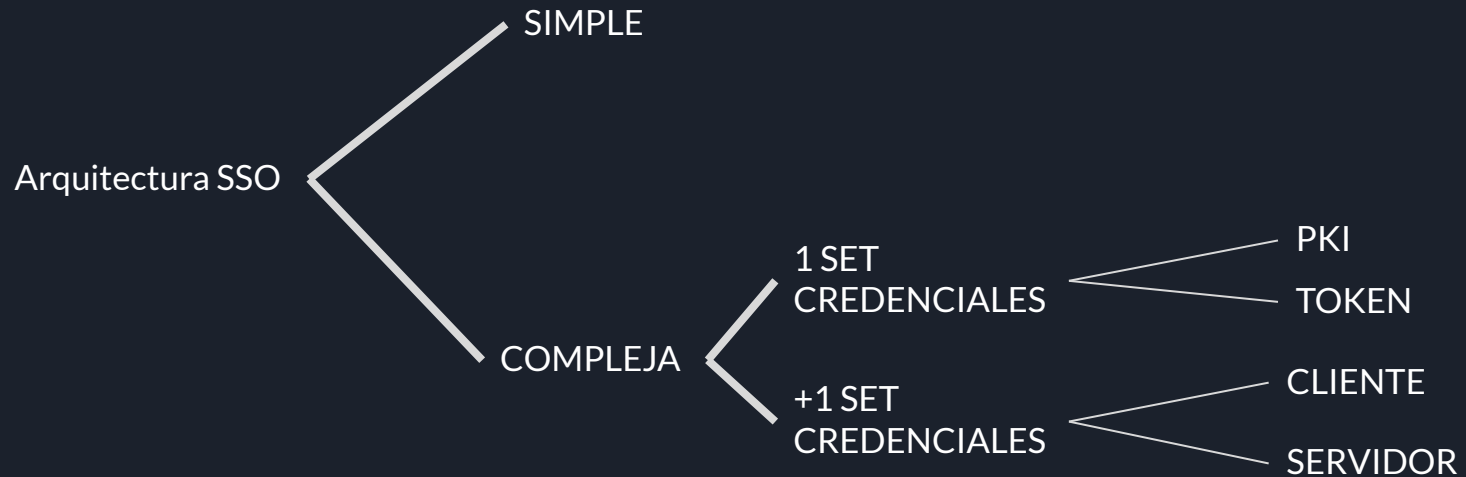
Motivación e Introducción

Muy común tener distintas cuentas con distinto nombre de usuario pero misma contraseña en muchos sitios.

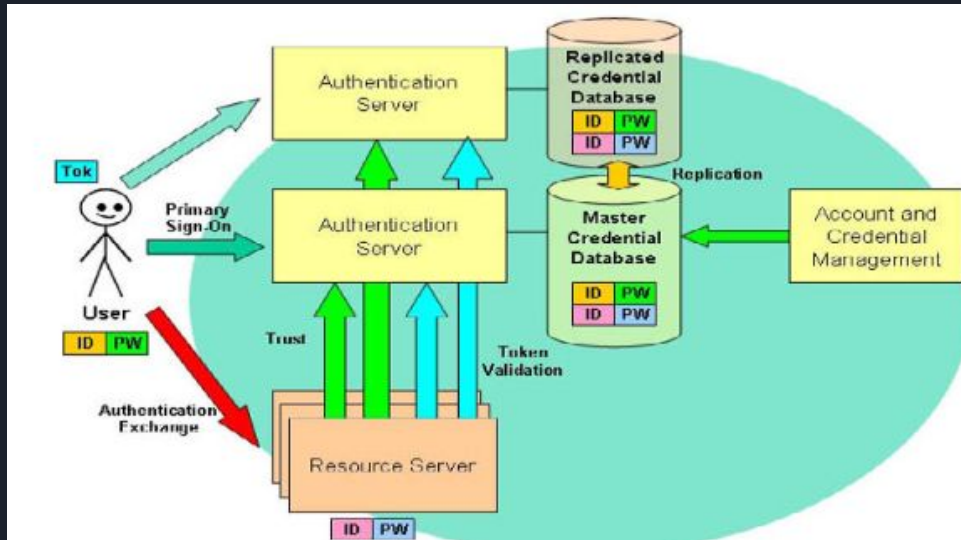
¡Esto es un peligro de seguridad! Se necesita un método para facilitarnos la vida y para mejorar nuestra seguridad.



Clasificación Arquitecturas SSO

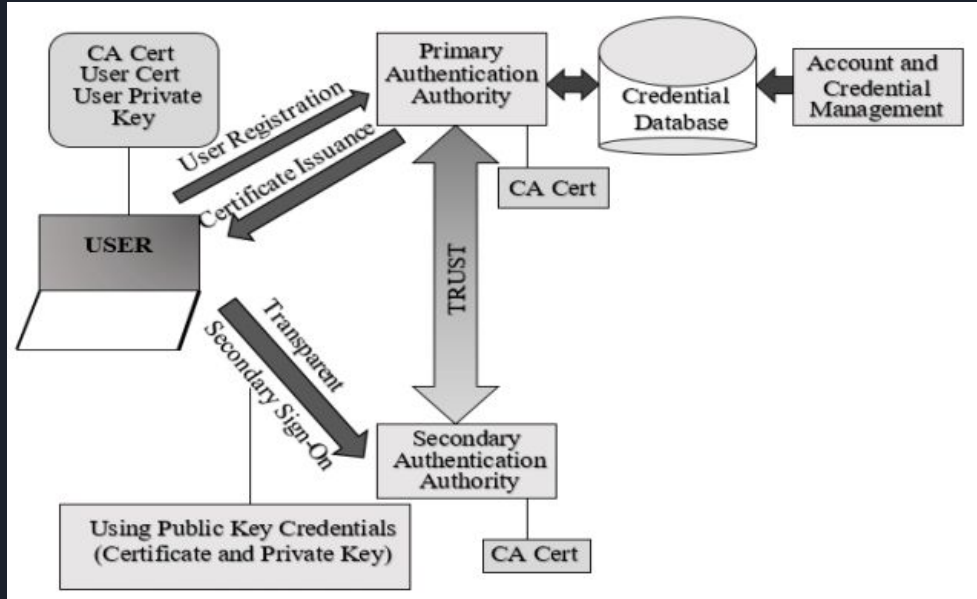


Arquitectura Simple



El usuario al querer acceder a un recurso protegido, se autentica ante un servidor de autenticación, el cual genera un token y es enviado al usuario. El usuario envía este token al servidor dónde se aloja el recurso, verifica el token con el servidor de autenticación, y si todo es correcto, le devuelve el recurso.

Arquitectura compleja: PKI SSO



Esta arquitectura trabaja con certificados (Clave pública, privada, info. adicional), generados por una autoridad de certificación (AC). Cuando el usuario quiere acceder a un recurso, envía un token que contiene el certificado, la autoridad comprueba que el certificado es válido con el CA, y le da acceso al recurso.



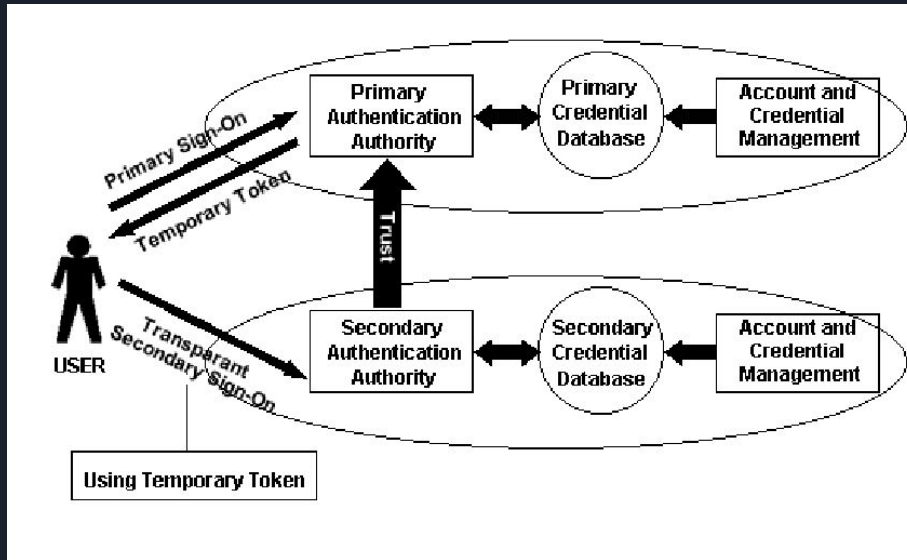
PROS:

- Vida más sencilla para el usuario, solo 1 set de credenciales.
- Cifrado asimétrico (clave pública y privada) por lo que alta seguridad.

CONTRAS:

- Infraestructura homogénea, mismo protocolo para todo.
- La validación de certificados puede ser compleja y requerir de tiempo.

Arquitectura Compleja: SSO con Token



El usuario al querer acceder a un recurso, recibe un token temporal, el cual, al acceder de forma transparente al segundo lugar, usa para verificar su identidad. La autoridad secundaria verifica que el token es válido con métodos criptográficos basados en clave secreta.



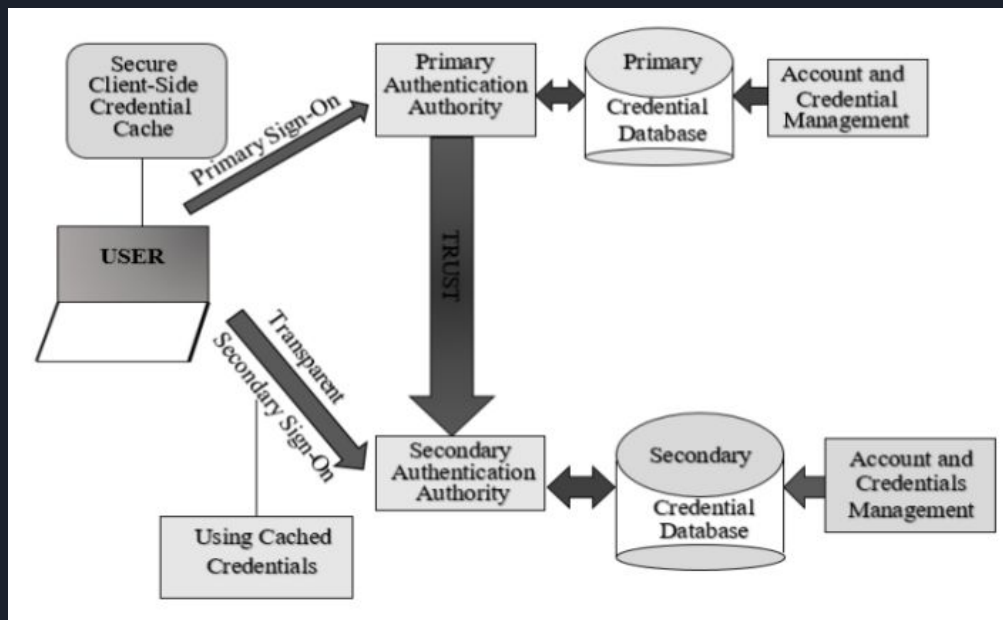
PROS:

- Al haber solo un set de credenciales, se facilita la vida del usuario.

CONTRAS:

- Infraestructura homogénea.
- Criptografía simétrica, por lo que sí se compromete la clave secreta, puede haber vulnerabilidades en el sistema.

Arquitectura Compleja: Secure Client-Side Credential Caching



El usuario inicia sesión en la autoridad primaria, y almacena las credenciales en su ordenador, las cuáles usa de forma transparente para iniciar sesión en la segunda autoridad. Cada vez que accedemos a una nueva autoridad, será necesario actualizar la caché.



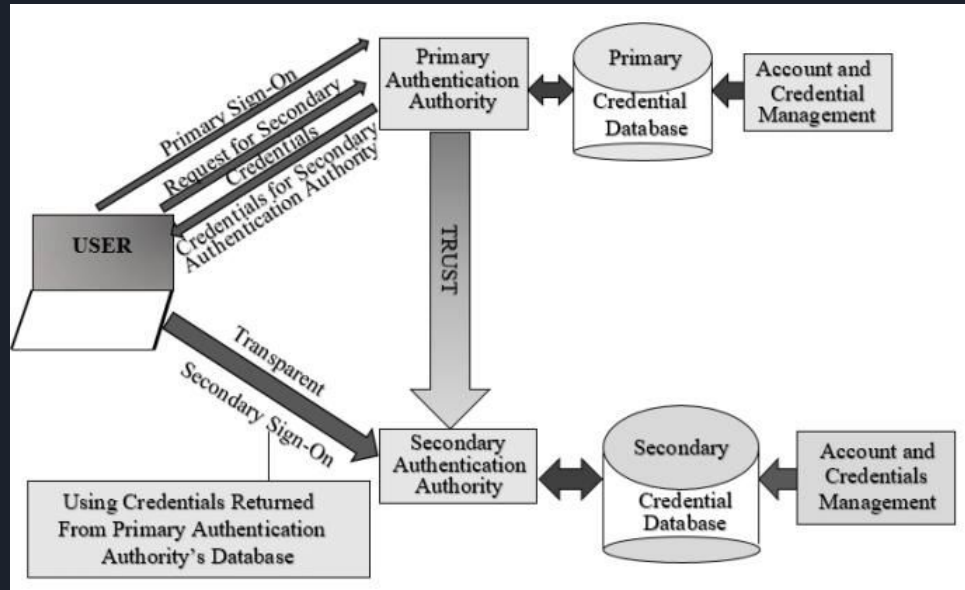
PROS:

- Arquitectura sencilla de implementar.
- Infraestructura no homogénea, se pueden implementar varios protocolos.

CONTRAS:

- Muy poco flexible.
- Al haber varios sets de credenciales, la vida del usuario y del administrador se complica.
- Se requiere mucha seguridad en el lado del cliente para evitar ataques a la caché.

Arquitectura Compleja: Secure Server-Side Credential Caching



Similar a la del cliente, pero la caché se encuentra en el servidor. El usuario inicia sesión en la autoridad primaria, y solicita las credenciales para la autoridad secundaria, las cuales usa de forma transparente para iniciar sesión en la autoridad secundaria.



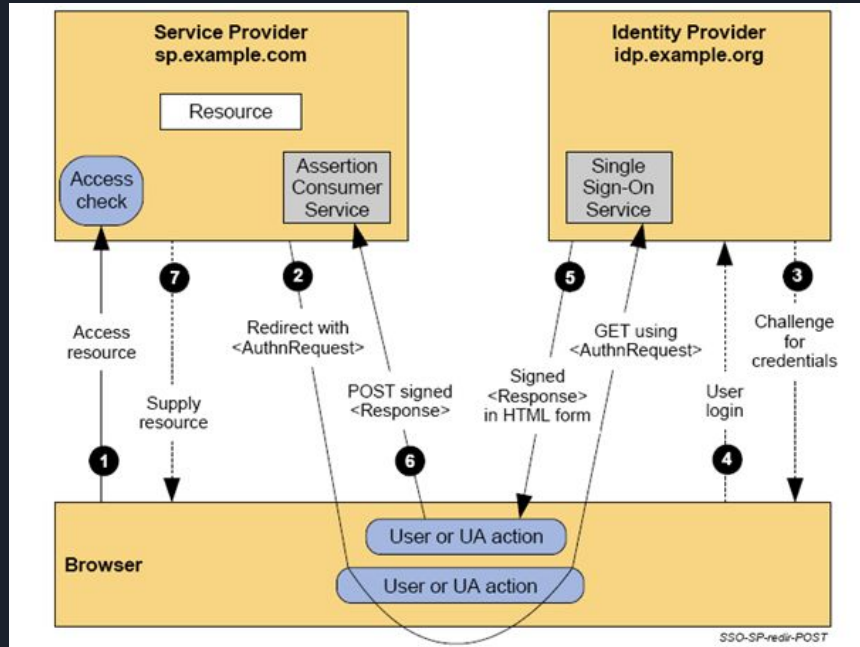
PROS:

- Infraestructura no homogénea, se pueden implementar varios protocolos.

CONTRAS:

- Requiere de mecanismo de sincronización de credenciales.
- Al haber varios sets de credenciales, la vida del usuario y del administrador se complica.
- La disponibilidad del servidor debe ser alta.

Protocolo 1: SAML



Protocolo basado en XML.

Usuario, IdP y SP.

Permite la comunicación entre dominios con distinta metodología de autenticación.

¿Cómo funciona?



Protocolo 2: OpenID

Solución de SSO descentralizada.

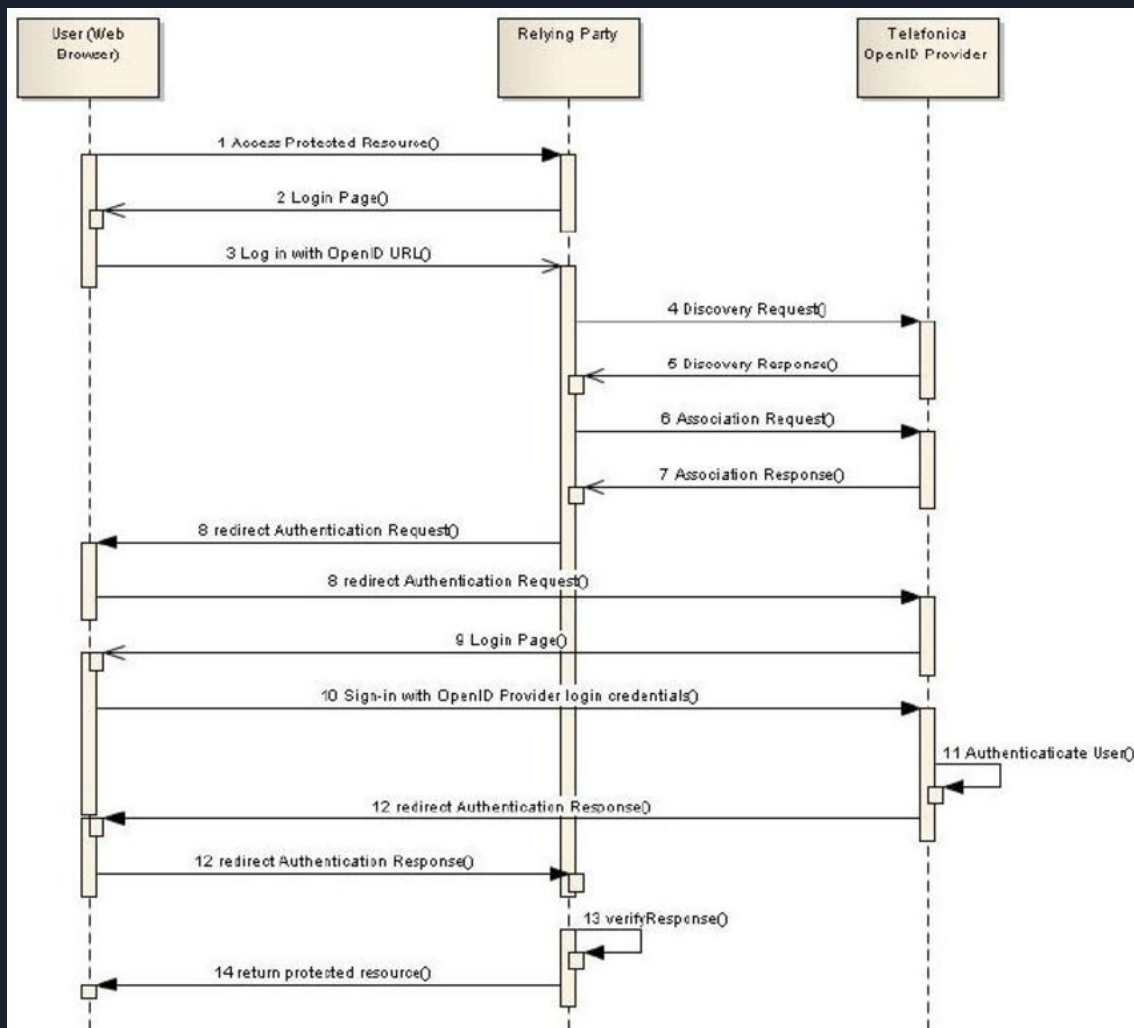
Usuario.

Relying Party: sitio al cual quiere acceder el usuario y obtener un recurso de este.

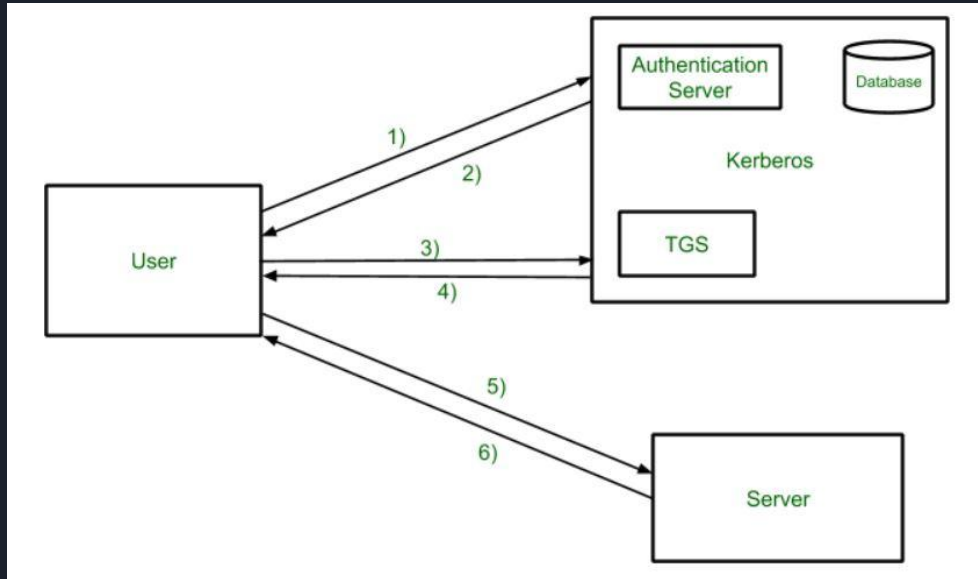
Identifier u OpenID: url elegida por el usuario para nombrar a su identidad.
<http://josemi.myopenid.com>

Identity Provider: encargado de registrar el identifier y autenticar al usuario.

¿Cómo funciona OpenID?



Protocolo 3: Kerberos



Protocolo usado en la mayoría de plataformas.

Funciona con tickets validados con criptografía y transportados con Remote Call Procedure.

Existencia del Key Grating Server.

Usado en la arquitectura de token.

¿Cómo funciona?



Beneficios SSO

- Aumento de la productividad para el usuario: no es necesario memorizar tantas contraseñas.
- Aumento de la productividad para el desarrollador: los desarrolladores no tienen que preocuparse por la autenticación en los frameworks.
- Facilidad de uso: se disminuye la complejidad para los administradores al administrar los usuarios.
- Seguridad: algo lógico, que el mayor beneficio a obtener es un aumento considerable de la seguridad.



Problemas SSO

- Escalabilidad: implementar SSO se puede volver una tarea compleja.
- Seguridad: si el usuario deja abierta una sesión en su ordenador y un atacante consigue acceder a esa sesión, podrá acceder al resto de servicios disponibles



Desafíos SSO

- Ataques de phishing: conseguir diferenciar entre sitios webs reales y falsos en los que se inicia sesión SSO.
- Seguridad: las credenciales, al ser almacenadas en algún lugar, hace que los usuarios duden de que se convierta en información sensible. Es un desafío seguir mejorando este aspecto y hacer cambiar la mentalidad de los usuarios.
- Resistencia al cambio: muchos usuarios no quieren adaptar estas mecánicas SSO y prefieren seguir usando métodos convencionales de almacenamiento de contraseñas.
- Desconocimiento de SSO: los usuarios que quieren implementar SSO, al no conocer bien los conceptos, y el proceso, pueden frustrarse y abandonar sus planes.



Referencias Bibliográficas

Bazaz, T., & Khalique, A. (2016). A Review on Single Sign on Enabling Technologies and Protocols. *International Journal Of Computer Applications*, 151(11), 18-25. <https://doi.org/10.5120/ijca2016911938>

Patil, Anita & Pandit, Rakesh. (2013). Analysis of Single Sign on for Multiple Web Applications. https://www.researchgate.net/publication/369269685_Analysis_of_Single_Sign_on_for_Multiple_Web_Applications


PPT - Single Sign-On PowerPoint Presentation, free download - ID:507407. (2012, junio 29). SlideServe. <https://www.slideserve.com/jersey/single-sign-on>

Facchini, C. (2023, julio 31). *Architecture for Single Sign-on (SSO) and identity provider*. Medium. <https://medium.com/@corrado.facchini/single-sign-on-sso-46ea458aec30>

De Clercq, J. (2002). Single Sign-On Architectures. En *Infrastructure Security* (pp. 40–58). Springer Berlin Heidelberg. https://link.springer.com/chapter/10.1007/3-540-45831-X_4

(S/f). Amazon.com. Recuperado el 15 de mayo de 2024, de <https://aws.amazon.com/es/what-is/sso/>

SakshiBhakhra Follow, S. (2019, julio 11). *Kerberos*. GeeksforGeeks. <https://www.geeksforgeeks.org/kerberos/>



Gomez Marmol, Felix & Kuhnen, M.Q. & Martinez Perez, Gregorio. (2011). Enhancing OpenID through a Reputation Framework. 1-18. 10.1007/978-3-642-23496-5_1.

https://www.researchgate.net/publication/221108184_Enhancing_OpenID_through_a_Reputation_Framework

SAML2 IdP overview 1.1. (s/f). Eclipse.org. Recuperado el 15 de mayo de 2024, de

https://wiki.eclipse.org/SAML2_IdP_Overview_1.1

N. Shaikh, K. Kasat and S. Jadhav, "Secured Authentication by Single Sign On (SSO): A Big Picture," *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, 2022, pp. 951-955, doi: 10.1109/ICCCIS56430.2022.10037708.

<https://ieeexplore.ieee.org/document/10037708>