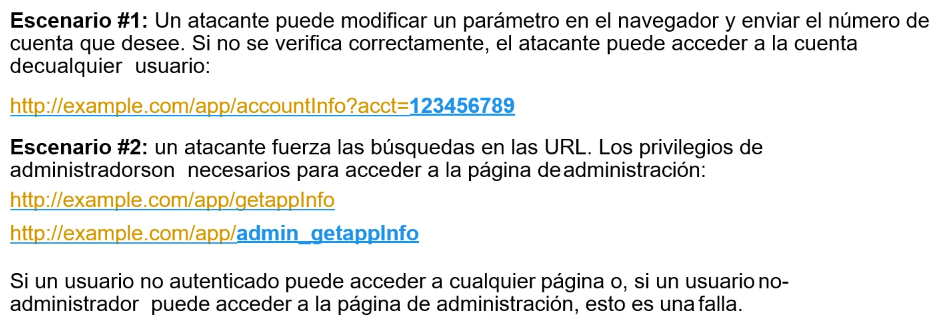
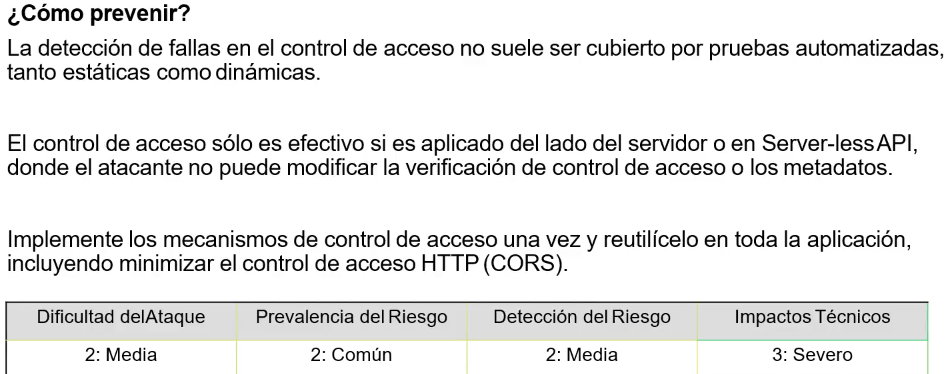
**Pérdida de Control de Acceso**



Esto hace referencia cuando nuestra aplicación web tiene referencias internas a módulos, a carpetas, a páginas, a enlaces o incluso a registros que no están bien segurizadas. El hacker, conociendo nuestra URL, podría entrar en todos estos recursos.

Inicialmente, un hacker no tendría por qué conocer las URL que le dan acceso a las diferentes aplicaciones o módulos de nuestra aplicación web. Sin embargo, si es un usuario anterior El cual ha hecho algo fraudulento y se le ha dado de baja y se ha guardado ciertas URLs o conoce algunos parámetros de envío porque es un usuario ya registrado, podría acceder a la cuenta, por ejemplo, de otro usuario.





Si hacemos pruebas unitarias de nuestra aplicación web, comprobando todos los privilegios de seguridad con el que entra cada usuario, tenemos bastante factibilidad de eliminar estos riesgos porque podemos hacer una comprobación antes que una aplicación pase a producción.

El control de acceso solo es efectivo si es aplicado del lado de servidor. El atacante en el lado servidor, evidentemente no puede modificar nada, ni la verificación ni el control de acceso, ni los metadatos.

En cambio, si lo hiciéramos desde el lado cliente, el hacker puede ver la petición que se está realizando hacia el servidor y de ese modo modificar el envío desde su propio navegador cliente.

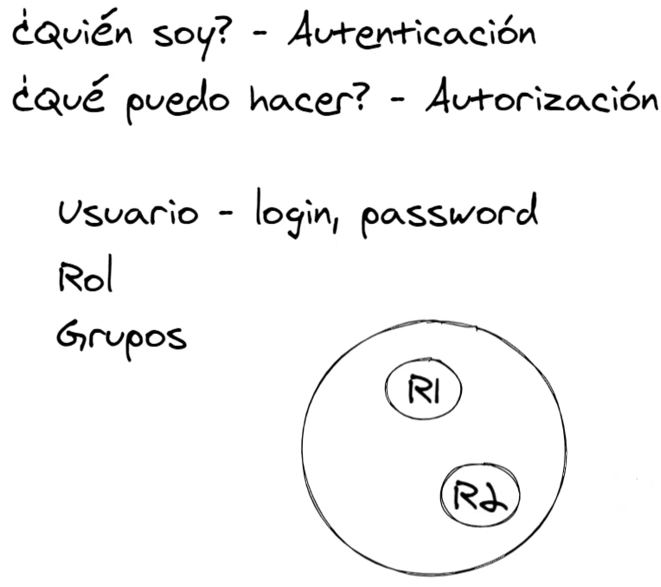
La dificultad de este ataque es media, ya que se necesita que nuestro hacker conozca bien las URL y cómo cambiar los parámetros de acceso.

La prevalencia de riesgo es común ya que son muchos los que intentan cambiando la URL, atacar a diferentes fuentes que tengamos en el lado del servidor.

La detección de riesgo es media ya que una aplicación web tiene muchas URLs distintas o puntos de entrada por las que nos pueden atacar, por lo tanto, debemos detectar cada una de ellas. Y esto con las pruebas unitarias se hace razonablemente bien.

El impacto técnico es severo porque estamos permitiendo a alguien entrar en algún lugar en el que no debería tener acceso y posiblemente ganaría privilegios extra.

**Autenticación y Autorización**

****

*Hay dos conceptos claves en seguridad y dos preguntas que deben ser contestadas.*

Lo primero de todo que tenemos que tener en cuenta es la pregunta de quién soy? Muchas veces cuando entramos en una aplicación web. Esta es la primera pregunta que deben contestar nuestros clientes, y esta pregunta es contestada por el concepto de autenticación.

Sin embargo, hay otra pregunta que también tiene su relevancia y esta es qué puede hacer? Esta pregunta está seleccionada y respondida por el concepto de autorización.

Son muchos los frameworks web que estudian estas dos preguntas y que intentan contestarlas, de manera que nosotros, cuando nos lo creemos en una aplicación, hagamos login en una aplicación. Intentar contestarla.

La autenticación va relacionada con el concepto de usuario que habitualmente tiene un login o un nombre de usuario y un password, para identificar ese usuario de forma inequívoca.

El que pueda hacer va habitualmente seleccionado con el concepto de rol.

Un usuario habitualmente puede tener varios roles. Así, por ejemplo, una aplicación web podría tener rol cliente y rol administrador.

Los grupos son colecciones de usuarios que comparten características similares:

* Agrupan usuarios con necesidades de acceso similares
* Simplifican la administración de permisos
* Permiten aplicar políticas de seguridad de manera colectiva
* Facilitan la gestión de recursos compartidos

Comparativa Roles y Grupos

* Los roles definen permisos y responsabilidades
* Los grupos facilitan la asignación de roles a múltiples usuarios
* Un usuario puede pertenecer a varios grupos y tener múltiples roles

Ejemplo: En una empresa de informática, tenemos el grupo de desarrollo de software y los roles dentro de dicho grupo de Team Líder, desarrollador, analista de sistemas, analista funcional, administrador de base de datos, etc.