CSRF PROTECTION

Jose Luis Alvarez Rios

5°F

Programacion

Indice

Introduccion

En esta presentación daremos explicación sobre un tema muy usado en la mayoría de los usuarios de Laravel el cual es "CSRF" el cual hace mas fácil la protección de la misma aplicación, protege los ataques de falsificación de solicitudes entre sitios.

Ya que la misma falsificación de solicitudes entre sitios es un exploit malicioso mediante el cual se ejecuta con comandos no autorizados en nombre de un usuario autenticado.

Explicaremos como hace la protección , cuando lo hace y que debes hacer para que funcione al igual que que otras aplicaciones o programas necesita para su aplicación.

También explicaremos para que funcionan los token, como lo hace que se necesita para que funicione y si es muy necesario para este método CSRF.

Al final del documento, después de la conclusión final al igual que un glosario por si algunas palabras no fueron claras y una bibliografía donde tu puedes verificar la información e informarte mas sobre este tema.

Investigacion

¿Qué es?

Cross-site Request Forgery, o CSRF, consisten en forzar a un usuario a ejecutar peticiones no deseadas a una web en la que están autentificados sin que este se dé cuenta. Este tipo de ataque no busca el robo de datos, si no el que estas peticiones provoquen cambios, ya que el atacante no tiene forma de ver la respuesta a estas peticiones falsas.

¿Cómo funciona?

Si la víctima se trata de un usuario normal, estas peticiones falsas pueden limitarse a por ejemplo enviar fondos a otra cuenta, cambiar el correo de contacto o la contraseña de acceso, y otras acciones que según el tipo de web pueden llegar a ser muy dañinas para la víctima. Sin embargo, si la víctima cuenta con privilegios administrativos, la web en su totalidad está comprometida, pudiendo causar daños muy graves e irreparables.

En este tipo de ataques, es necesaria la utilización de ingeniería social para engañar a la víctima y provocar que ejecute la petición falsa, por ejemplo enviando un link por email:

```
<a href="http://banco.com/transferir.php?cuenta=juan&cantidad=10000">Ver fotos</a>
```

Si la víctima recibe este enlace, lo abre y está autentificada en la página pero esta no está preparada para evitar ataques CSRF, transferiría a la cuenta «juan» la cantidad indicada.

No sólo es posible realizar este ataque mediante peticiones GET, también es posible con peticiones POST, si el atacante coloca un formulario malicioso en una web y logra que la víctima acceda y lo ejecute.

¿Cómo protegernos?

La técnica más comun para asegurar nuestra web contra este tipo de ataques y que nuestros usuarios no se conviertan en víctimas, es mediante el uso de tokens o identificadores únicos, generados de forma aleatoria y asociados a la sesión del usuario, que son insertados como campos ocultos en los formularios y añadidos en los enlaces para las acciones que se puedan considerar «sensibles». Cuando el usuario realiza una de estas operaciones, el servidor comprueba que el token enviado es correcto, permitiendo que se ejecute la operación, o cancelándola en caso de que sea incorrecto, de forma que nos aseguramos de que sea el usuario legítimo quien realmente ha hecho la petición, ya que es imposible para el atacante conocer el token de su víctima.

Hoy en día cualquier CMS moderno cuenta con medidas de protección contra los ataques CSRF, pero es esencial mantener las versiones siempre a la última para obtener las correcciones de fallos de seguridad. En caso de no usar un CMS y usar programación propia, es muy recomendable implementar medidas para proteger a los usuarios contra este tipo de ataque.

MAS INFORMACIÓN

Un ataque CSRF es una técnica maliciosa que obliga al navegador web de una víctima, autenticada previamente en algún servicio como correo electrónico o banca online, a enviar una petición HTTP falsa a una aplicación web vulnerable.

Así el ciberdelincuente es capaz de realizar una acción a través de su víctima, ya que la actividad maliciosa será procesada en nombre del usuario conectado, por lo que la aplicación pensará que se trata de una petición legítima.

Las vulnerabilidades CSRF explotan la confianza del usuario,y se llevan a cabo solo cuando este ha validado sus datos en una aplicación o sitio web. Al introducir solicitudes que parecen válidas, pero que son falsificadas, el atacante podrá modificar el comportamiento de la aplicación a su favor.

El mecanismo más usado para prevenir vulnerabilidades de sitios cruzados es solicitar información adicional en cada petición HTTP para determinar si en realidad viene de una fuente confiable. Este proceso consiste en la inclusión de un token secreto o valor aleatorio; que se genera y se informa al navegador del usuario en el momento que inicie sesión.

Este código de validación debe ser difícil de adivinar y si una solicitud no incluye dicho código o este no coincide con el valor esperado, el servidor rechazará la petición.

Otra acción que puede ayudar a prevenir la falsificación de solicitudes en sitios cruzados es el envío doble de cookies. Se trata de una variante del mecanismo del token; ya que en este

caso el código debe corresponderse con el identificador de sesión en la cookie.

El servidor debe comprobar que ambas sean iguales en cada solicitud. Pero como el sitio de donde proviene el ataque no es el mismo que el de la víctima; no se podrá realizar esta verificación y por ende, se rechazará la petición.

Además de los mecanismos anteriores, las empresas también pueden aplicar otras acciones como solicitar al usuario credenciales o un CAPTCHA. Asimismo, muchos sitios suelen limitar el tiempo de vida las sesiones, aunque es importante hacerlo de una manera que no afecte la usabilidad de la web.

Hoy día se considera que el 70% de los sitios web son hackeables. Y aunque hayas implementado medidas para prevenir vulnerabilidades CSRF; debes saber que las técnicas de los atacantes evolucionan rápido, por lo que siempre debes estar atento. Recuerda además que después de una actualización o un cambio en el código de tu sitio; algunos controles podrían eliminarse y dejar el sitio vulnerable.

Glosario

Exploit: Las definiciones habituales hablan de un programa o código que se aprovecha de un agujero de seguridad (vulnerabilidad).

Token: Es una cadena de caracteres que tiene un significado coherente en cierto lenguaje de programación. Es utilizado para facilitar el proceso de autenticación de usuarios.

CSRF: Cross-site Request Forgery o Falsificador de solicitudes entre sitios

CMS: Son aplicaciones que nos permiten gestionar de una manera cómoda los contenidos publicados en los sitios web.... CMS son las siglas de Content Management System, que se traduce directamente al español como Sistema Gestor de Contenidos.

Peticiones GET: HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado.

Maliciosa: Que habla o actúa con intención encubierta para beneficiarse en algo o perjudicar a alguien.

Cookie: Una cookie es un archivo de pequeño tamaño enviado por un sitio web y almacenado en el navegador del usuario, de

manera que el sitio web puede consultar la actividad previa del navegador.

CAPTCHA: CAPTCHA son las siglas de Completely Automated Public Turing test to tell Computers and Humans Apart (prueba de Turing completamente automática y pública para diferenciar ordenadores de humanos).

Hacker: Persona con grandes conocimientos de informática que se dedica a detectar fallos de seguridad en sistemas informáticos.

Hackeable: Sitio vulnerable.

Bibliografias

https://blog.evidaliahost.com/cross-site-request-forgery-csrf/

https://laravel.com/docs/8.x/csrf

https://developer.mozilla.org/es/docs/Web/HTTP/Methods

https://www.gb-advisors.com/es/como-prevenir-vulnerabilidades-sitios-cruzados-csrf/