Cross-Site Scripting

Katherine Ramos Pereira
Universidad Mayor de San Andrés
Facultad De Ciencias Puras Y Naturales
Carrera Informática
Análisis Y Diseño De Sistemas De Información
Kt_99@hotmail.es

RESUMEN

El Cross-Site-Scripting es una vulnerabilidad que aprovecha la falta de mecanismos de filtrado en los campos de entrada y permiten el ingreso y envió de datos sin validación alguna, que permite a una tercera parte inyectar en páginas web vistas por el usuario código Java Script o en otro lenguaje script similar ejemplo. VBScriptpudiendo generar secuencias de comandos maliciosas que impacten directamente en el sitio o en el equipo de un usuario.

Este tipo de vulnerabilidad se conoce en español con el nombre de **Secuencias de comandos en sitios cruzados.**

Causa un impacto tanto a una aplicación web como a usuarios que de manera inconsciente activen dicha secuencia de comandos.

Dicho código malicioso se compone de cadenas de datos cuyo contenido son scripts completos contenidos en enlaces o ejecutados desde formularios.

En caso de que sea ejecutado el mismo se ejecutara en el equipo del usuario con todos los privilegios permitidos por las políticas de seguridad configuradas en el navegador del usuario o del sitio visitado, pudiendo realizar acciones diversas como la captura de cookies de usuario o la activación de servicios y componentes del sistema operativo del usuario víctima.

Palabras Clave
Ataque, web, malicioso, usuario, seguridad,
XSS, vulnerabilidad, aplicaciones, internet.

1. INTRODUCCION

El Cross Site Scripting (XSS) es una vulnerabilidad muy común hoy en día, se puede encontrar en la mayoría de las aplicaciones web en Internet.

Este fallo compromete más que nada, la seguridad del usuario y no la del servidor. Consiste en inyectar código HTML o Java script en una aplicación web, con el fin de que el navegador de un usuario ejecute el código inyectado al momento de ver la página alterada.

Comúnmente el XSS se utiliza para causar una acción indebida en el navegador de un usuario, pero dependiendo de la vulnerabilidad, se puede explotar el fallo para causar una acción indebida en un servidor o en una aplicación.

Esta limitación se debe a que el código HTML se interpreta en el navegador de un usuario y no en el servidor. Así que si alguien inyecta código HTML en alguna aplicación web no podría hacer daño alguno al servidor, ya que éste nunca interpreta el código HTML, solo los navegadores. Por eso este ataque se determina: “ataque del lado del cliente”.

El XSS se puede utilizar para hacer phishing, (robo de credenciales), troyanizar navegadores. Todo depende de la página.

2. ATAQUES DE CROSS-SITE SCRIPTING (XSS)

Los datos se incluyen en el contenido dinámico que se envía a un usuario de la web sin ser validado por código malicioso.

El contenido malicioso enviado al navegador web a menudo toma la forma de un segmento de Java Script, pero también puede incluir HTML, Flash o cualquier otro tipo de código que el navegador puede ejecutar. La variedad de los ataques basados en XSS es casi ilimitada, pero normalmente incluyen la transmisión de datos privados, como las galletas o cualquier otra información de sesiones a la atacante, la reorientación de la víctima a la página web controlado por el atacante, o la realización de otras operaciones maliciosos en la máquina del usuario en la apariencia del sitio vulnerable.

Los ataques XSS generalmente se pueden clasificar en dos categorías:

- Almacenados
- Reflejados.

Hay una tercera, mucho menos conocido tipo de ataque XSS llamada DOMXSS.

Los ataques XSS almacenados

Ataques almacenados son aquellos en los que el código inyectado se almacena de forma permanente en los servidores de destino, como en una base de datos, en un foro de mensajes, registro de visitantes, campo de comentarios, etc. La víctima recupera el script malicioso en el servidor cuando se solicita al almacenado información.

Ejemplo de un ataque XSS almacenado.

- El siguiente segmento de código JSP consulta una base de datos para un empleado con un ID dado y se imprime el nombre del empleado correspondiente.

```
<%...
 Statement stmt = conn.createStatement ();
ResultSetrs = stmt.executeQuery ("select * from emp where id="+eid);
if (rs != null) {
rs.next ();
  String name = rs.getString ("name");
%>

EmployeeName: <%= name %>
```

Este código funciona correctamente cuando los valores de nombre son bien educados, pero no hace nada para evitar abusos, si no lo son. Una vez más, este código puede parecer menos peligroso porque el valor del nombre se lee de una base de datos, cuyo contenido aparentemente son gestionados por la aplicación. Sin embargo, si el valor del nombre se origina a partir de datos suministrados por el usuario, a continuación, la base de datos puede ser un conducto para el contenido malicioso. Sin la validación de entrada correcta en todos los datos almacenados en la base de datos, un atacante puede ejecutar comandos maliciosos en el navegador web del usuario. Este tipo de explotación, conocido como XSS almacenados, es particularmente insidioso porque la indirección causada por el almacén de datos hace que sea más difícil identificarla amenaza y aumenta la posibilidad de que el ataque afectará a múltiples usuarios. XSS tiene su inicio en el formulario con los sitios web que ofrecen un "libro de visitas" a los visitantes. Los atacantes podrían incluir Java Script en su entrada en el libro, y todos los visitantes posteriores a la página de libro de visitas que se ejecute el código malicioso.

Los ataques XSS reflejados

Ataques reflejados son aquellos en los que el código inyectado se refleja en el servidor web, por ejemplo, en un mensaje de error, resultado de la búsqueda, o cualquier otra respuesta que incluye algunos o todos de la entrada se envía al servidor como parte de la solicitud. Ataques reflejados se entregan a las víctimas a través de otra ruta, tal como en un mensaje de correo electrónico, o en algún otro servidor web. Cuando un usuario es engañado para hacer clic en un enlace malicioso o enviar un formulario especialmente diseñado, el código inyectado viaja al servidor web vulnerable, lo que refleja el ataque al navegador del usuario. El navegador ejecuta el código, ya que provenía de un servidor de "confianza".

El ejemplo más común se puede encontrar en los sitios web de anuncios de a bordo que proporcionan funcionalidad list-style de correo basado en web.

- El siguiente segmento de código JSP lee un ID de empleado, eid, de una solicitud HTTP y lo muestra al usuario.

```
<% String eid = request.getParameter ("eid"); %>
        ...
        Employee ID: <%= eid %>
```

El código de este ejemplo funciona correctamente si eid contiene texto alfanumérico único estándar. Si eid tiene un valor que incluye meta-caracteres o código fuente, el código será ejecutado por el navegador web, ya que muestra la respuesta HTTP.
Inicialmente esto podría no parecer gran parte de una vulnerabilidad. Después de todo, ¿por qué alguien escriba un URL que causa el código malicioso se ejecute en su propio ordenador? El verdadero peligro es que un atacante crear el URL malicioso, a continuación, utilizar el correo electrónico o trucos de ingeniería social para engañar a las víctimas para que visite un enlace a la URL. Cuando las víctimas haga clic en el vínculo, sin saberlo, reflejan el contenido malicioso mediante la aplicación web vulnerable a sus propios equipos. Este mecanismo de explotación de las aplicaciones web vulnerables es conocido como XSS reflejados.

3. CONSECUENCIAS DEL ATAQUE XSS

La consecuencia de un ataque XSS es el mismo, independientemente de si se almacena o refleja (o basado en DOM). La diferencia está en la forma en la carga útil que llega al servidor. No se deje engañar en pensar que es "sólo lectura" o "brochureware", el sitio no es vulnerable a graves ataques XSS reflejados. XSS pueden causar una variedad de problemas para el usuario final, que varían en severidad desde una molestia a un compromiso completo de la cuenta. Los ataques XSS más graves implican la divulgación de cookie de sesión del usuario, lo que permite a un atacante secuestrar la sesión del usuario y hacerse cargo de la cuenta. Otros ataques dañinos incluyen la divulgación de los archivos de los usuarios finales, la instalación de programas de caballo de Troya, redirigir al usuario a otra página o sitio web, o modificar la presentación de los contenidos. Una vulnerabilidad XSS que permite a un atacante modificar un comunicado de prensa o noticia podría afectar la cotización de la empresa o disminuir la confianza del consumidor. Una vulnerabilidad XSS en un sitio farmacéutica podría permitir a un atacante modificar información de dosificación que resulta en una sobredosis. Cómo determinar si es vulnerable

Las fallas de XSS pueden ser difíciles de identificar y retirar de una aplicación web. La mejor manera de encontrar defectos es realizar una revisión de seguridad del código y la búsqueda de todos los lugares en los que la entrada de una petición HTTP podría hacer su camino en la salida HTML.

4. ¿CÓMO PROTEGERSE?

Las defensas primarias contra XSS

Se describen en el XSS Prevención OWASP (el proyecto de seguridad de aplicaciones web de código abierto). Además, es crucial que desactivar la compatibilidad con HTTP TRACE en todos los servidores web. Un atacante puede robar datos de las cookies a través de Java script. Este ataque se monta cuando un usuario publica un script malicioso en un foro así que cuando otro usuario hace clic en el enlace, una llamada de seguimiento HTTP asincrónico se activa, que recoge información de las cookies del usuario desde el servidor, y luego lo envía a otro servidor malicioso que recoge la información de las cookies para que el atacante puede montar una sesión de secuestrar ataque. Esto se mitiga fácilmente retirando el apoyo a HTTPTRACE en todos los servidores web.

5. COMPONENTES DE SEGURIDAD

El proyecto OWASP ESAPI ha producido un conjunto de componentes de seguridad reutilizables en varios idiomas, incluyendo la validación y escapar de las rutinas para evitar la manipulación de parámetros y la inyección de ataques XSS. Además, la aplicación de entrenamiento proyecto WebGoat OWASP tiene lecciones de Cross-Site Scripting y codificación de datos.

7. CONCLUSION

Las vulnerabilidades de XSS abarcaban cualquier ataque que permita ejecutar código de "scripting".

A través de un ataque XSS, se puede secuestrar cuentas, cambiar configuraciones de los usuarios, acceder a partes restringidas del sitio, modificar el contenido del sitio, etc. Las fallas de XSS pueden ser difíciles de identificar y retirar de una aplicación web. Es por eso que el proyecto OWASP usa componentes de seguridad para evitar un ataque malicioso.

8. REFERENCIAS

1. **[1]** Cross-site Scripting (XSS). Autor: fortify software Disponible en: **https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29**
2. [2] Cross-site scripting Autor: Sirdarckcat Disponible en: http://es.wikipedia.org/wiki/Cross-site_scripting
3. [3]Tutorial básico cross site scripting Autor: Adrian Disponible en: http://www.shellshocklabs.com/search/?q=cross+site+scripting
4. [4] Tutorial de cross site scripting [XSS]+como atacar ejemplos. Autor: ICEnetX Disponible en: **http://www.taringa.net/posts/ebookstutoriales/2306553/Tutorial-de-cross-site-scripting-XSS-como-atacar-ejemplos.html**

[5]XSS Capítulo 1: Introducción a Cross Site Scripting. Autor: zhynar_X Disponible en: http://wiki.elhacker.net/bugs-y-exploits/nivel-web/xss