

XSIO - Cross Site Image Overlaying

Sven Vetsch / Disenchant
sven.vetsch@disenchant.ch

<http://disenchant.ch>

v1.0

August 7, 2007

<i>CONTENTS</i>	2
-----------------	---

Contents

1 About this Paper	3
1.1 What is it about?	3
2 XSIO - Cross Site Image Overlaying	3
2.1 A new Attack Type	3
2.2 The name	3
2.3 How it works?	3
3 Use Cases	5
3.1 Defacements	5
3.2 Disinformation	5
3.3 Phishing	5
4 BBCode	5
5 XSIO through CSS	6
6 Countermeasures	7
7 Conclusion	7
8 About the Author	7

1 About this Paper

1.1 What is it about?

This paper describes a new attack type in the sector of web application security. It's an attack which is very easy to perform but can have a huge impact under some circumstances. Here you'll learn how to perform these kind of attacks and what impact that such an attack could have.

By the way, everything in this paper has just been tested in Firefox 2.0.0.6

2 XSIO - Cross Site Image Overlaying

2.1 A new Attack Type

As already written above, it's a new kind of attack but I think this is not the first time, that someone is using it because it's so easy to perform such an attack. I describe it in this paper because I found nothing about it on the Net. So what I want to say with this, don't expect too much but perhaps you'll also see something new which you can use. Also, it's up to you if you think it's really an attack or just a cool way of using CSS for something malicious.

2.2 The name

God damn! Yes, I know there are more and more names for attack types in the webappsec field which starts with an "X", which stands for "Cross" but I like exactly that and I think the attack described in this paper is something like XSS (Cross Site Scripting) but without script but instead there's an image. You think this sounds strange? So, stay curious and read the paper.

2.3 How it works?

So, let's start with how exactly it works. To show you directly the whole impact, I'll show you how to use XSIO against myspace.com with a 0-Day example. To be vulnerable to XSIO, a site just needs to let you set Hyperlinks (`xyz`) and Images (``).

The attack is based on CSS or better say, the "style" attribute of it. The following listing, will illustrate the attack.

Listing 1: A XSIO attack

```
1 <a href=http://disenchant.ch>
2 <img src=evil.gif
   style=position:absolute;left:123px;top:123px; />
3 </a>
```

Sven Vetsch / Disenchant
<http://disenchant.ch>

Looks like a normal image, which links to <http://disenchant.ch>, doesn't it? of course because it's nothing more but let's have a look at the mentioned myspace.com example and you'll see why this can be a problem.



1: Myspace Original



2: Myspace after XSIO

OK, through this two images, you can see what you can do with XSIO. This looks at the first view like a modification in the DOM tree but when you look at the code before you can see that it's nothing like that. You just have an image stored somewhere and then you load this image on a predefined position so that it looks like a normal part of the website and as an addition you can use it in form of a link to point users to for example a phishing site but we will have a look at this in the section 3.3 Phishing.

The following code was insert into the "About Me:" section in the myspace.com profile:

Listing 2: A XSIO attack

```

1 <a href="http://disenchant.ch">
2 <img src=http://disenchant.ch/powered.jpg
   style=position:absolute;right:320px;top:90px; />
3 </a>

```

Of course you've to be aware of different screen resolutions and you can also set the position through percent instead of pixels. Didn't I already wrote, that it's simple at all? Of course it is but you can make an image for every important part on the website and if someone clicks there, he/she will go to the site you've defined and not the page, the normal link will point to.

3 Use Cases

In this section, I'll show you three use cases for this kind of attack.

3.1 Defacements

The first thing you can probably think of is a defacement because you can use an image, which will fill the whole screen and so you will only have your image there instead of the normal page. Depending on the site, this can be something which really has a denial of service effect. If you think for example of forums and so on, when someones doing an XSIO in a comment, then nobody can answer anymore to the post where the attack was launched.

3.2 Disinformation

If you can place an XSIO on a website which has news on it, you can put your image over the them and you've got your own (wrong) information there, I think that this doesn't need an example.

3.3 Phishing

Another use case for this kind of attack, I see in phishing. For this, think about a site where you for example can click on a login button or for example an a "write a comment" link. When you set an overlay on this through XSIO, you can bring a user onto your own site where you've set up a phishing site. Also here, I think I don't have to explain this in deep.

4 BBCode

On more and more sites, you'll find BBCode or something similar but also there you can use XSIO under some circumstances. Let's have a look on an example.

Listing 3: XSIO attack on BBCode

```
1 [url=http://disenchant.ch]
2 [img]http://disenchant.ch/powered.jpg"
   style=position:absolute;right:320px;top:90px;"[/img]
3 [/url]
```

This will of course only works when BBCode is not properly implemented.

5 XSIO through CSS

This section of the paper is very theoretical because normally you won't have the possibility to use something like this and when it's possible, you can also directly execute Javascript or do something which's much more evil than a XSIO. Anyway, perhaps someone will run into a situation where this could be useful. Let's have a look on the following example.

Listing 4: XSIO through CSS

```
1 <html>
2 <head>
3   <style type="text/css">
4     h1#xsio {
5       overflow: hidden;
6       background-image: url(./test.gif);
7       background-repeat: no-repeat;
8       padding-top: 36px;
9       height: 0px;
10    }
11  </style>
12 </head>
13 <body>
14   <h1 id="xsio">test!</h1>
15   <h1 id="noxsio">test!</h1>
16 </body>
17 </html>
```

The example above will result in the following output in your (Firefox) web browser:



test!

3: XSIO through CSS

What happened here? The node with *id="xsio"* will become a background image but the original content *test!* will be overwritten by this image. So if you can control the CSS part of a website you can also do XSIO attacks, just like accessing the DOM tree. Get the nodes which you want to overlay with something and write the CSS part for it.

6 Countermeasures

There are not many different things you can do against XSIO attacks but to defend your web application against it, you need just better filters in place. So if you have to allow images and links which can be defined by users, first of course you have to care about CSRF but that's not the topic of this paper but after this, you should really implement this functionality in a way, that there can be just a location for the image and also for the link but never let the user set any attributes (for example "style"). On the client side, you can just be aware of such attacks and for example if it's used for phishing, you really have to check the link location carefully.

7 Conclusion

I think at least now you get the point and know how to perform XSIO attacks. This stuff is really simple and I think many people out there have already performed things like this but now the whole world can get this idea which can have a big impact in the security of a website.

Have Fun :)

8 About the Author



4: Sven Vetsch

My name is Sven Vetsch and I'm also known under my nickname Disenchant which is also nearly my domain; it's <http://disenchant.ch>. Under this URL you'll find my blog, where I'm blogging mainly about web application security and web technology stuff at all, I also release some proof of concepts from time to time on my blog so you're kindly invited to have a look.

To have also a look at my profession, I'm actually a Security Tester, Analyst and Engineer at Dreamlab Technologies Ltd. (<http://dreamlab.net>) which is based in Bern (Switzerland) and I'm there the guy who's responsible for all web technologies security stuff. In the mid of September this year (2007) I'll start to study computer science at "Bern University

of Applied Sciences" (<http://www.bfh.ch>) and in my spare time I like to go to concerts, meeting friends and of course also do web hacking stuff of any kind.

For more information about me or if you've got any questions, don't hesitate to write me an e-mail to sven.vetsch@disenchant.ch

Sven Vetsch / Disenchant
<http://disenchant.ch>