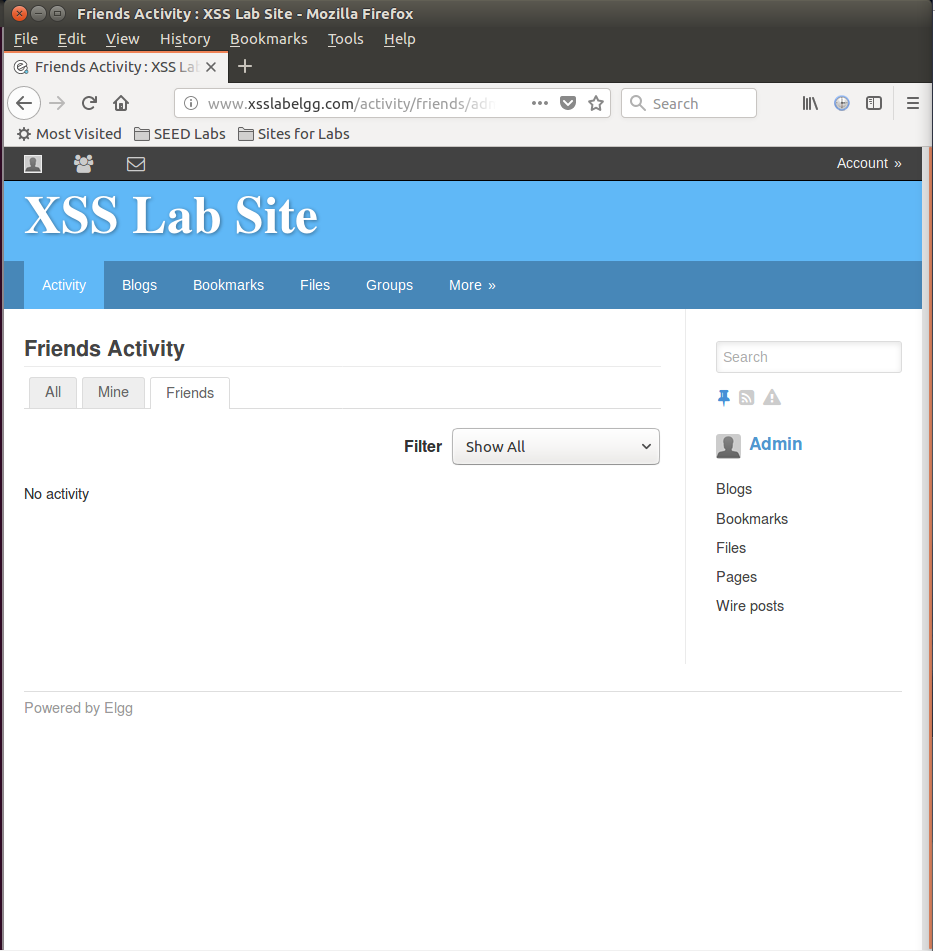
Shane Hagan

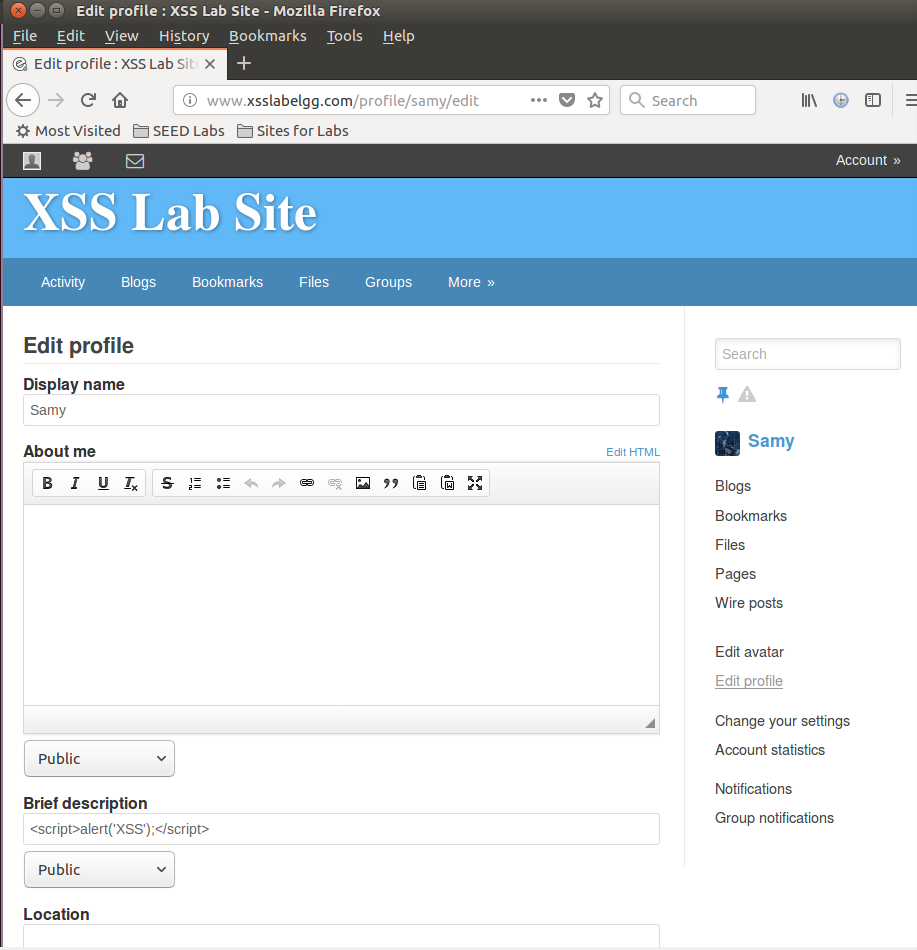
CMPSC 443

Lab 6

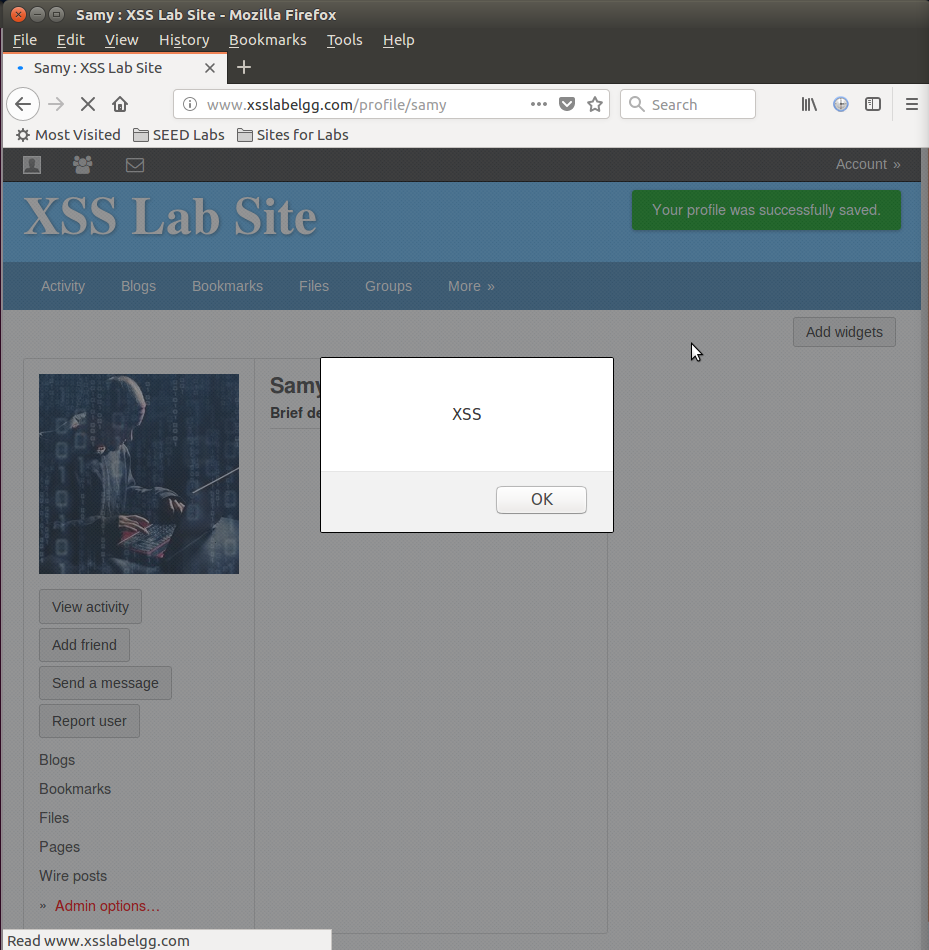
**Task 1**



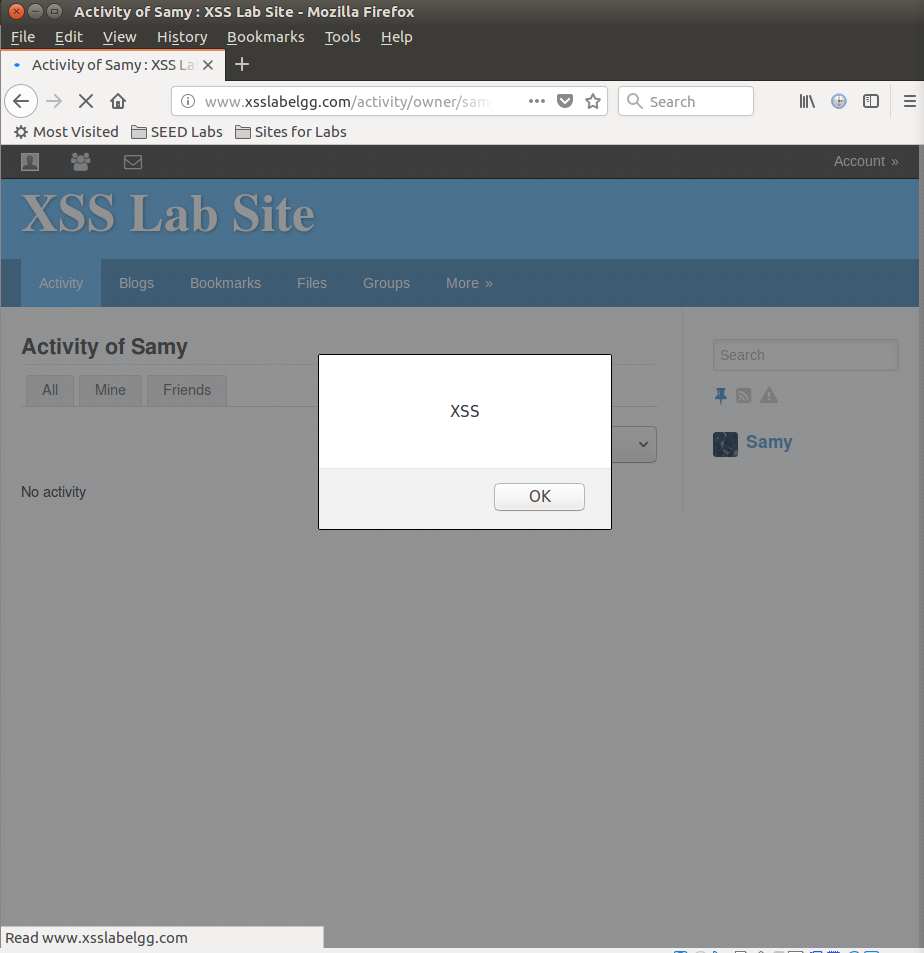
Here, I am just showing that the environment is set up and working correctly for this lab.



Here, I placed the javascript in the description section.

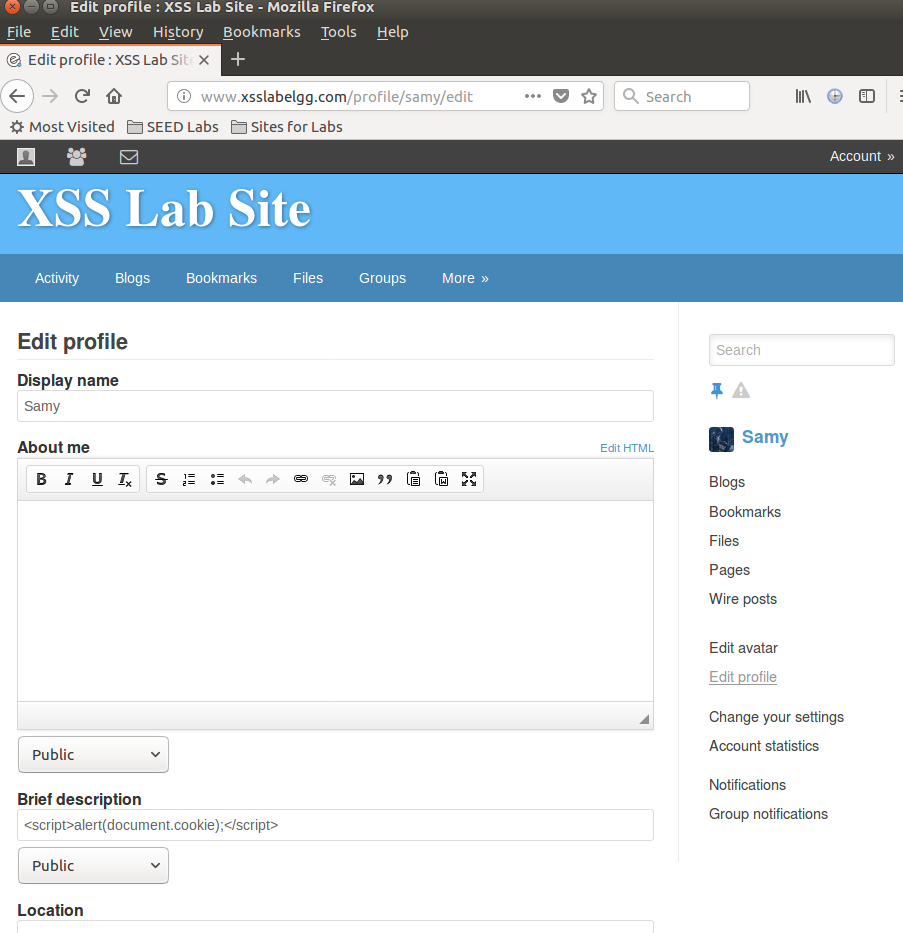


This is when we saved the profile and returned to Samy’s profile. As we can see, the alert message was successful here.

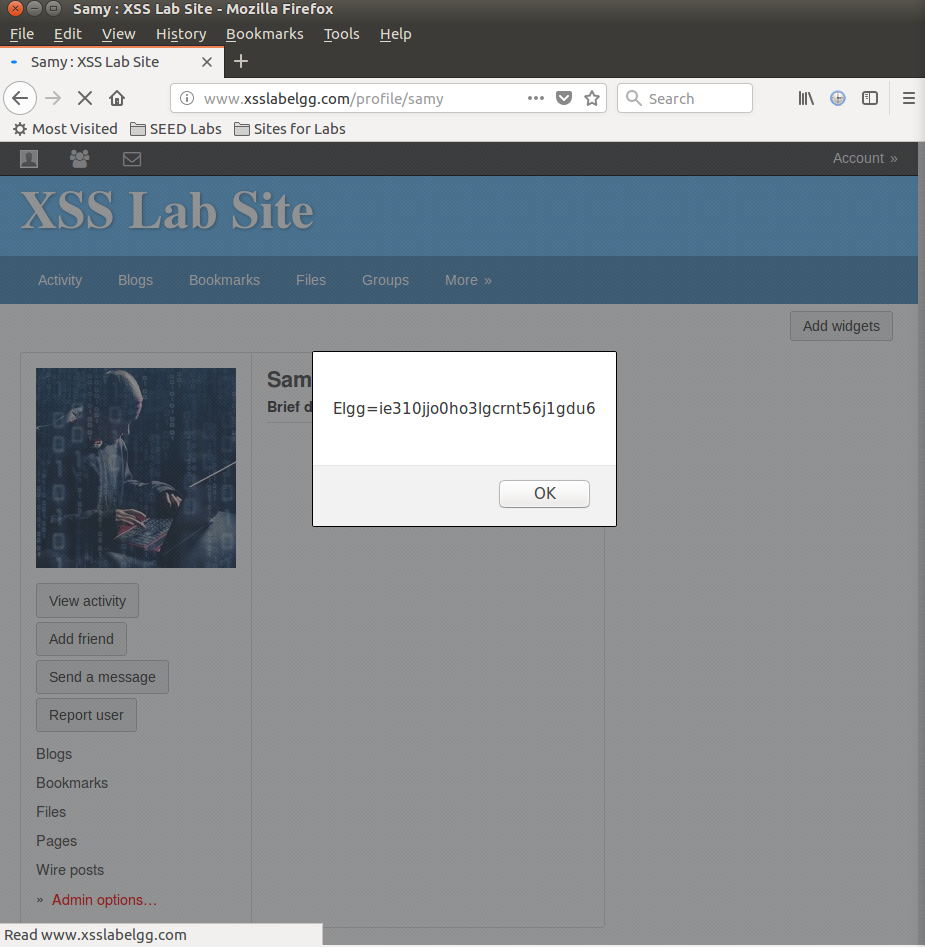


Here is just showing if we login to another account and go to the page for Samy, the alert message still pops up.

**Task 2**

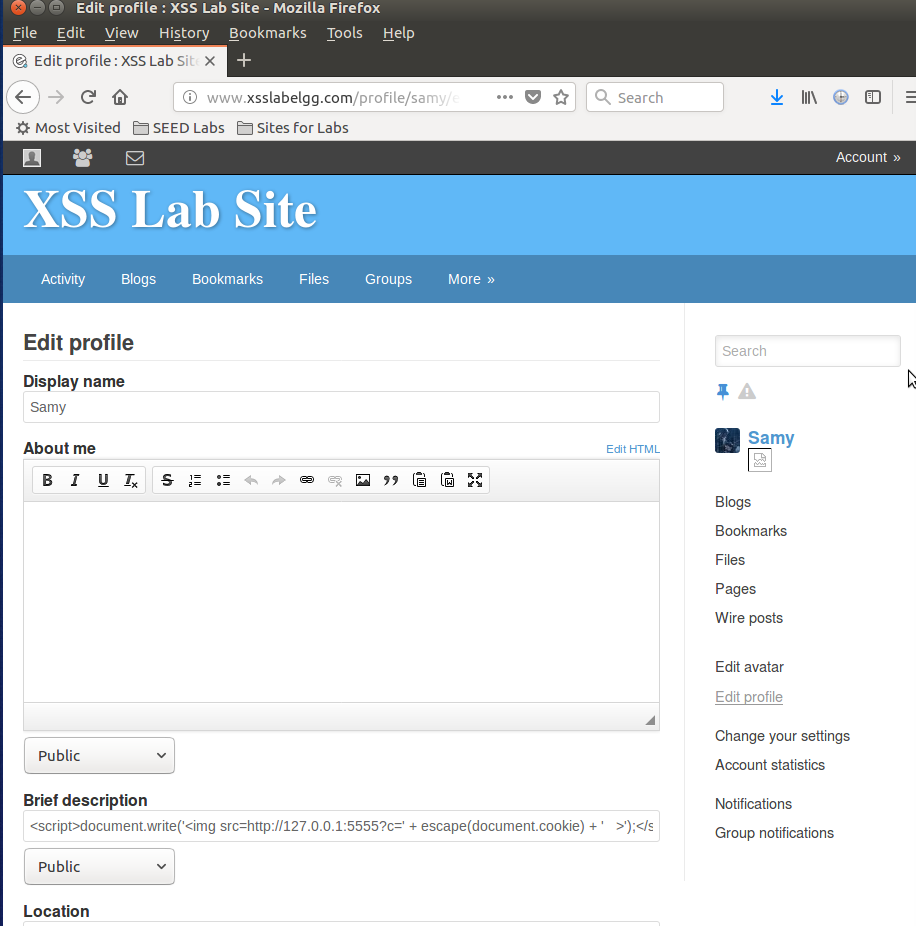


Here, we place the cookie command in the description box.

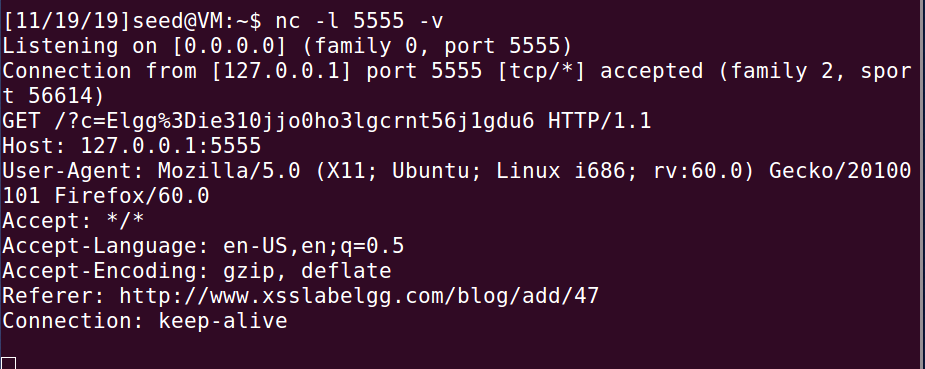


As we can see, the cookie information now appears as our alert when we get on Samy’s page.

**Task 3**

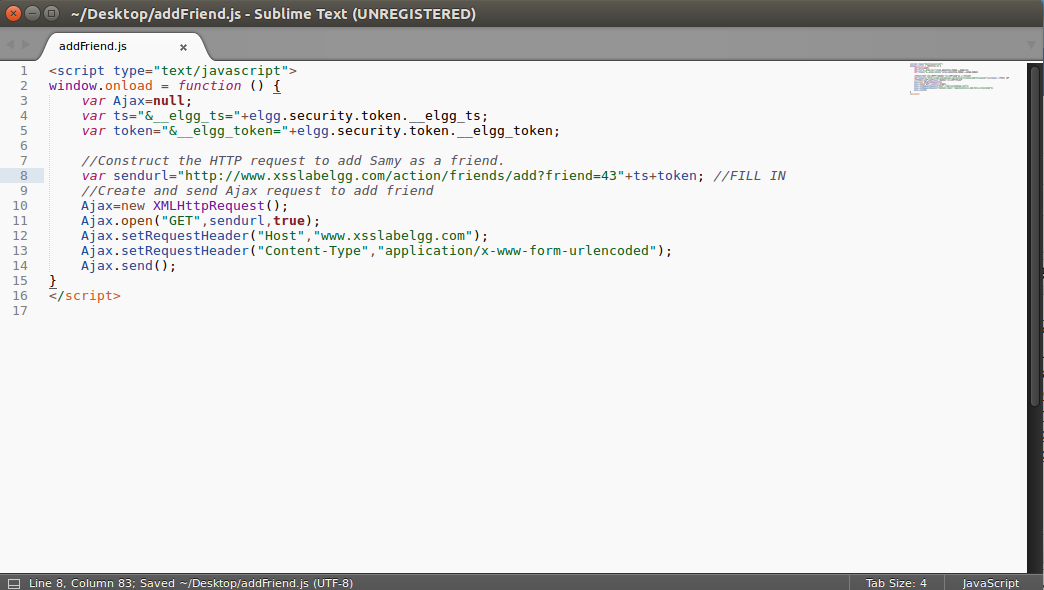


Here, we place the given javascript into the description box.

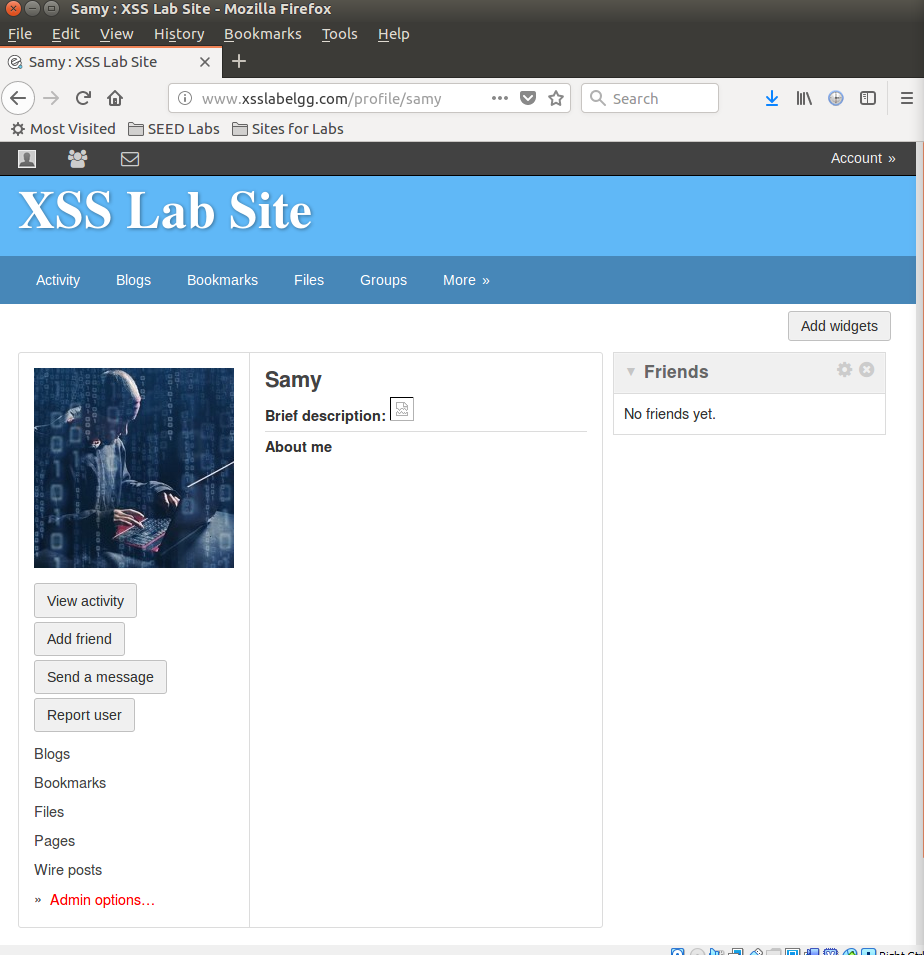


This is in the terminal when we use the nc command. We can see the tracing was successful and the attacker would have the ability to trace the victims account.

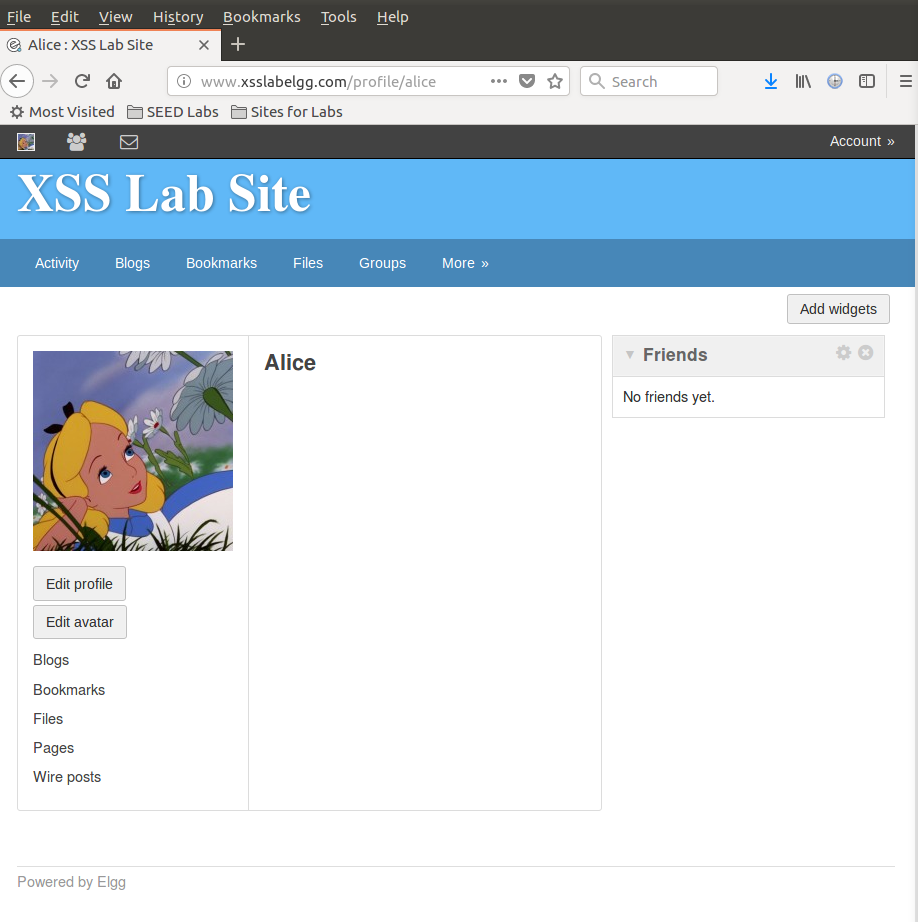
**Task 4**

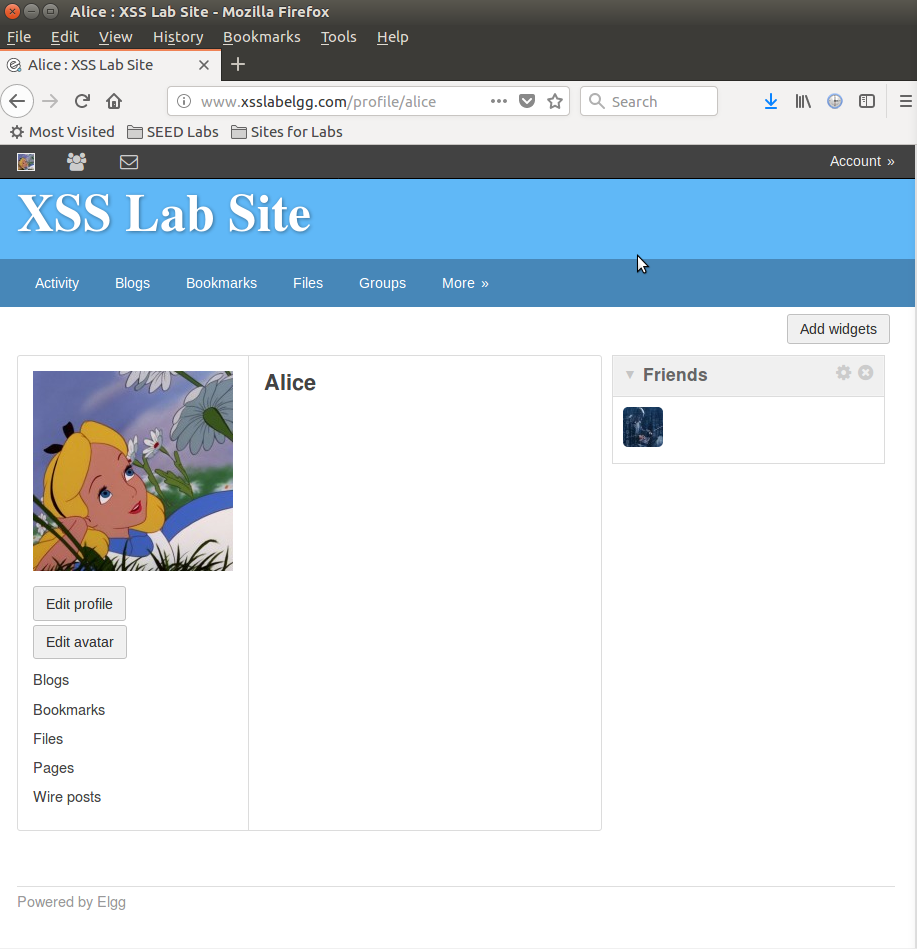


This is our javascript with the filled in information. We were able to obtain this information by using the HTML developer tool for Firefox and first adding Alice and Samy to see the exact code that is outputted for this function.



This is a screenshot of Samy (above) and Alice’s (below) accounts before the we visit Samy’s account.





Here is a screenshot of Alice’s account after we visit Samy’s account. As we can see, before Alice had no friends in her right tab, and after visiting, she now has a friend, being Samy. From this, we can conclude the javascript executed successfully.

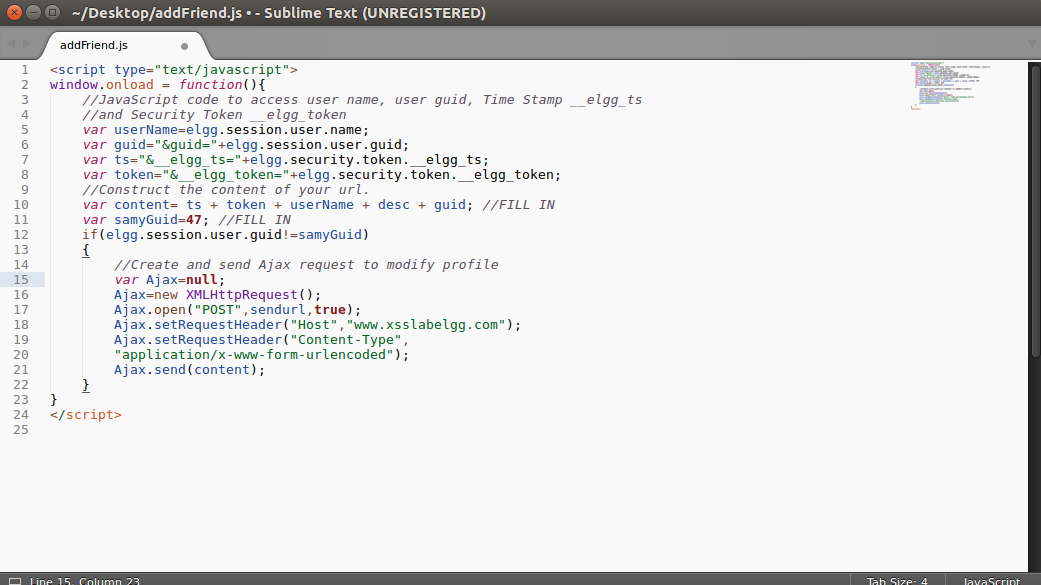
Question 1: Explain the purpose of Lines ➀ and ➁, why are they are needed?

Answer: For line 1 – this line contains the code “\_\_\_elgg\_ts” and “\_\_elgg\_\_token”, which are security measures taken by the site to prevent any type of threats. So, this line is somewhat of a security measure for the user’s of the site, even though the attacker would be able to figure out these variables by using Firefox’s developer tool. For line 2 – this is so the attacker will be able to alter the victim’s account, so the attack can be successful.

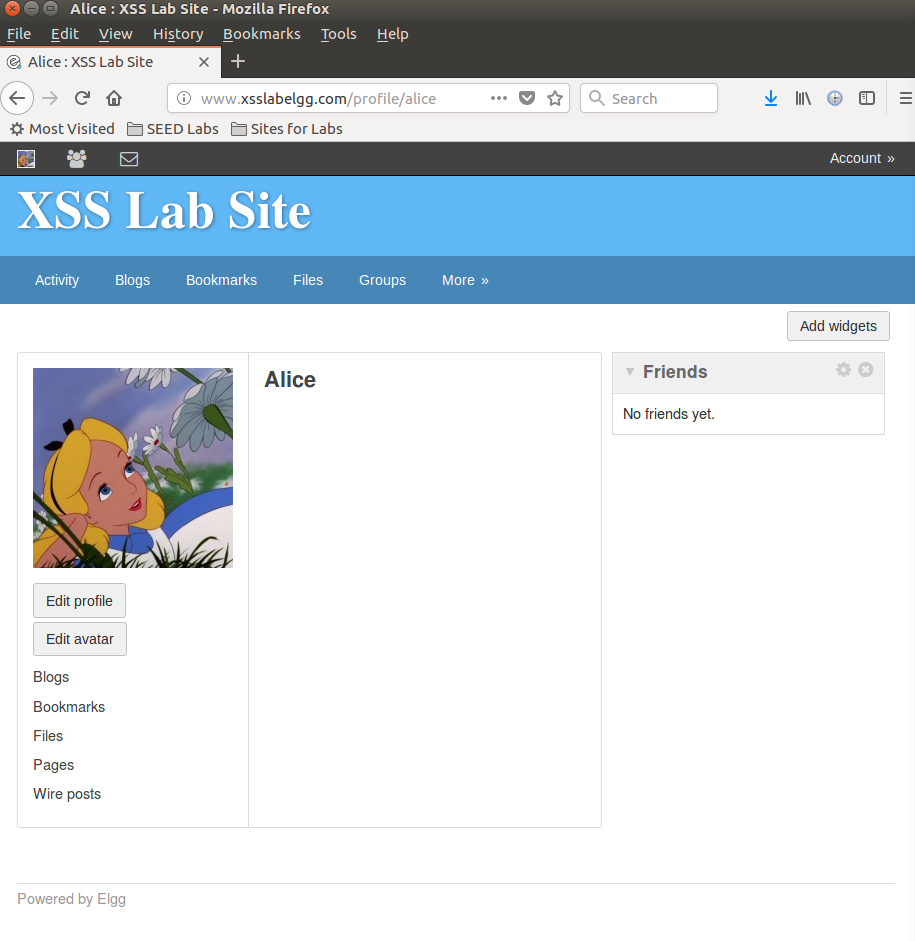
Question 2: : If the Elgg application only provide the Editor mode for the "About Me" field, i.e., you cannot switch to the Text mode, can you still launch a successful attack?

No, it will not work. It was said before that the text mode adds characters. As long as we place the code in the editor mode field, we will be fine.

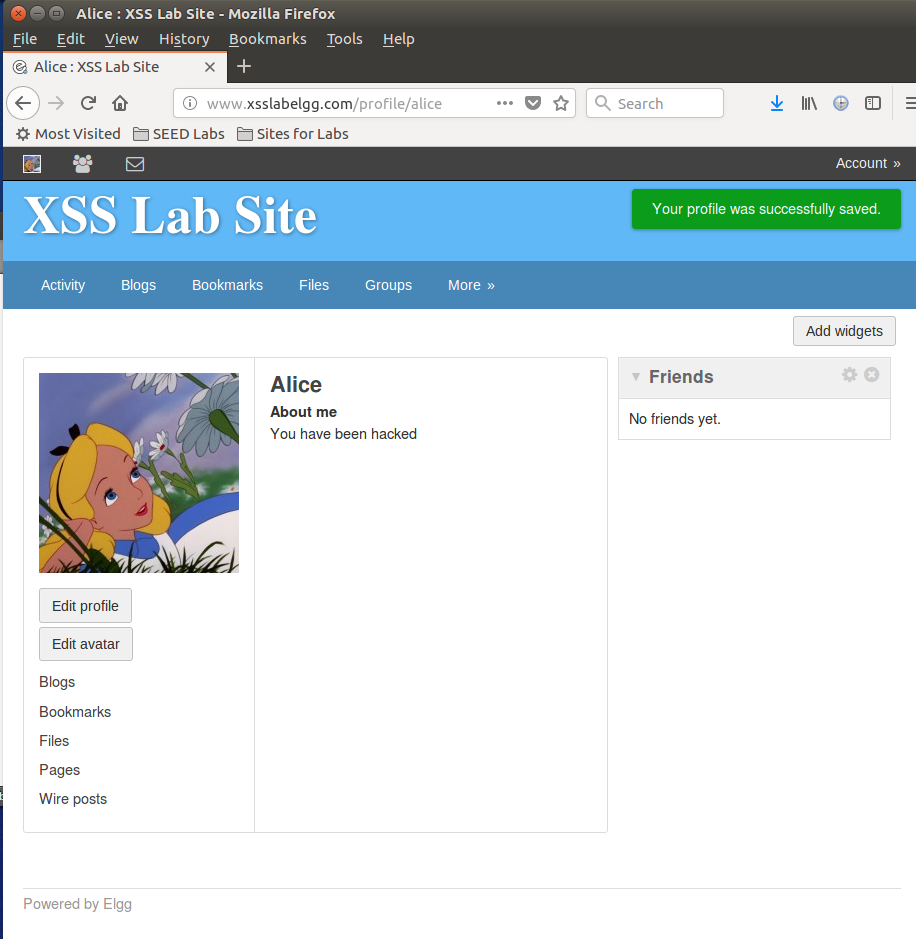
**Task 5**



Here is a screenshot of the code we use to edit the “about me” section for victim’s profiles. Once again, we used the Firefox inspect tool to see exactly what occurred with the site when the about me was changed.



Alice’s account profile before visiting Samy’s page.

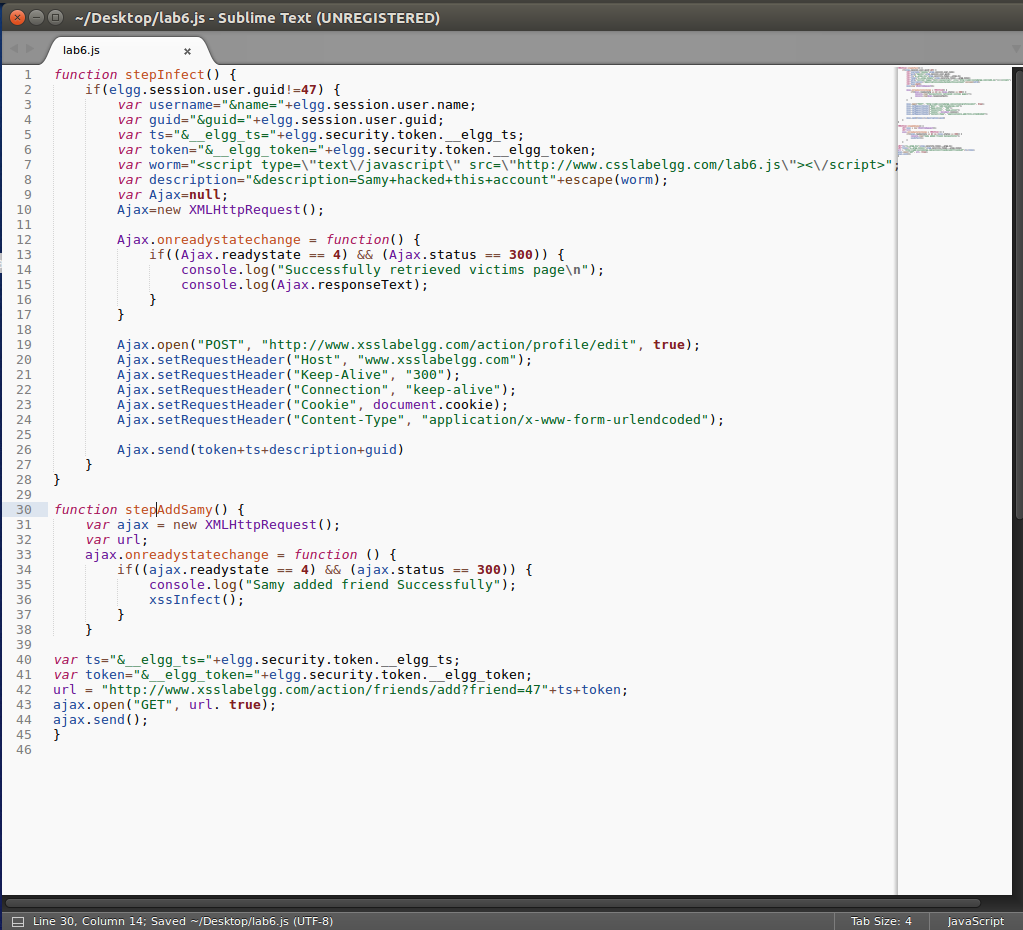


After adjusting the code, a couple screenshots before this, it was determined we’d need to put in the about me so it would actually fill out Alice’s about me. Here, I chose “you have been hacked” and added this to the code so it would populate. This is the screenshot after visiting Samy’s profile as Alice. As we can see, it was successful because Alice’s about me has been populated with the message.

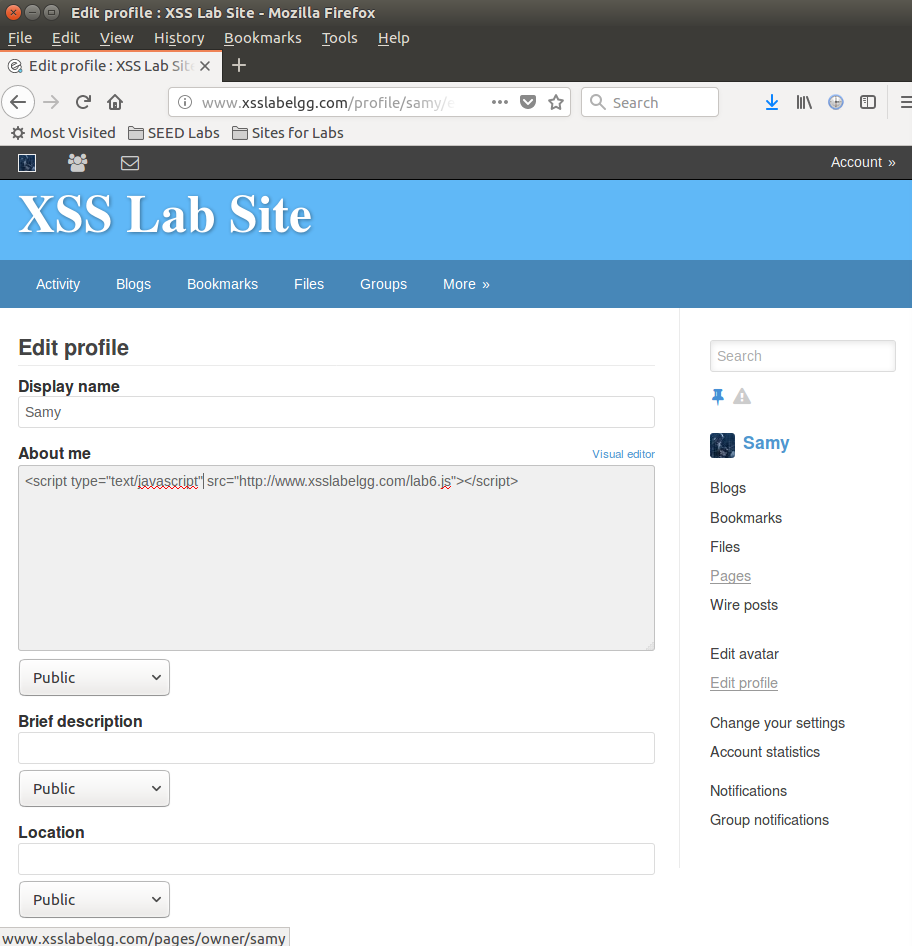
Question 3: Why do we need Line ➀? Remove this line, and repeat your attack. Report and explain your observation.

Answer: after removing the line, we see it was not successful. This is because we need to ensure that the victim’s user guid is not the same as Samy’s.

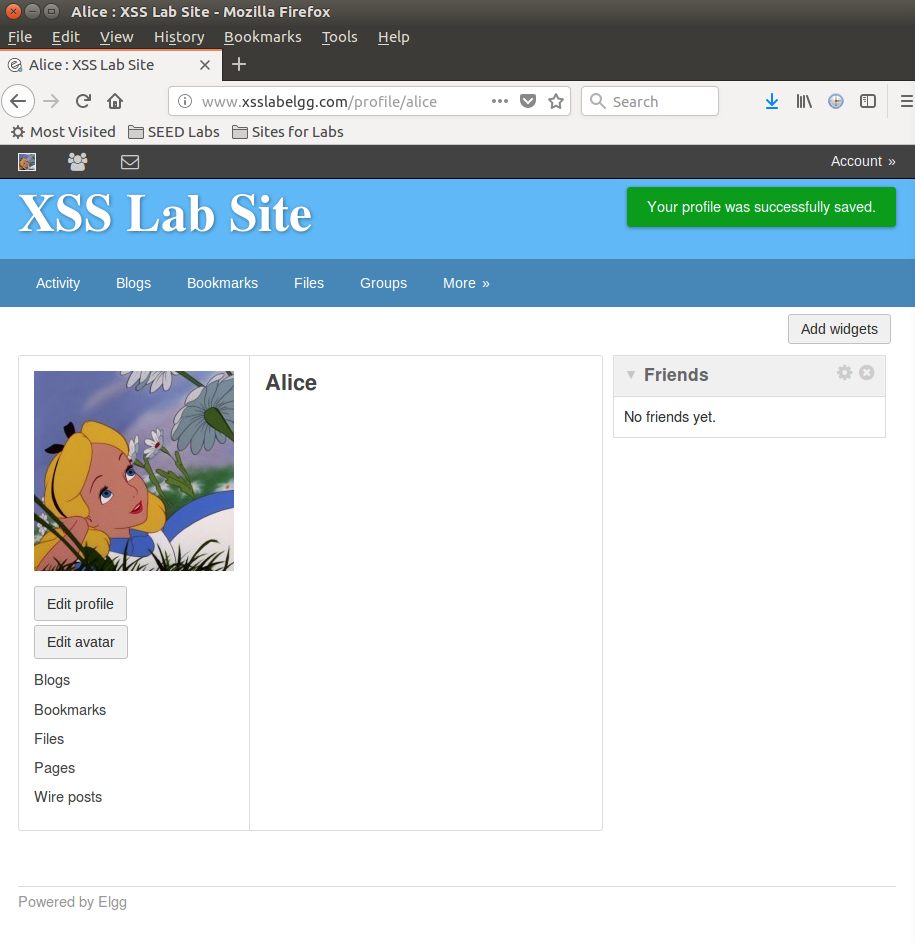
**Task 6**



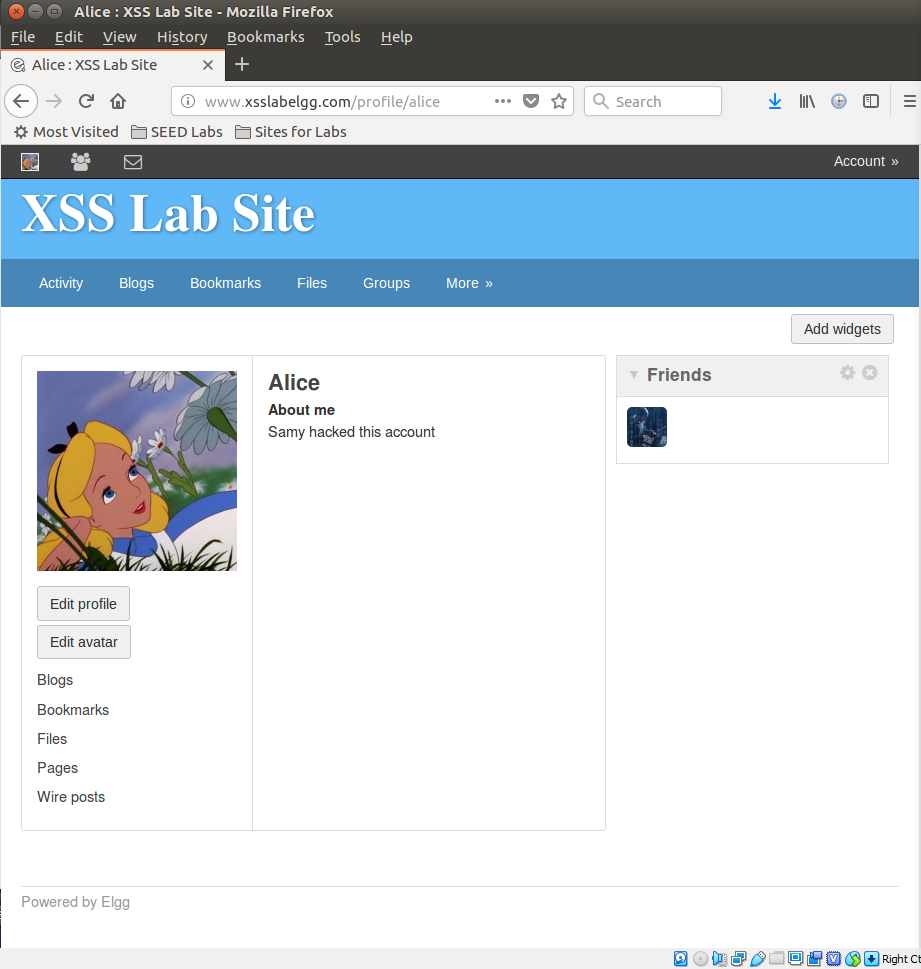
Screenshot of our code for Task 6, combining our code from the previous two steps so it would just self-propagate and become a worm.



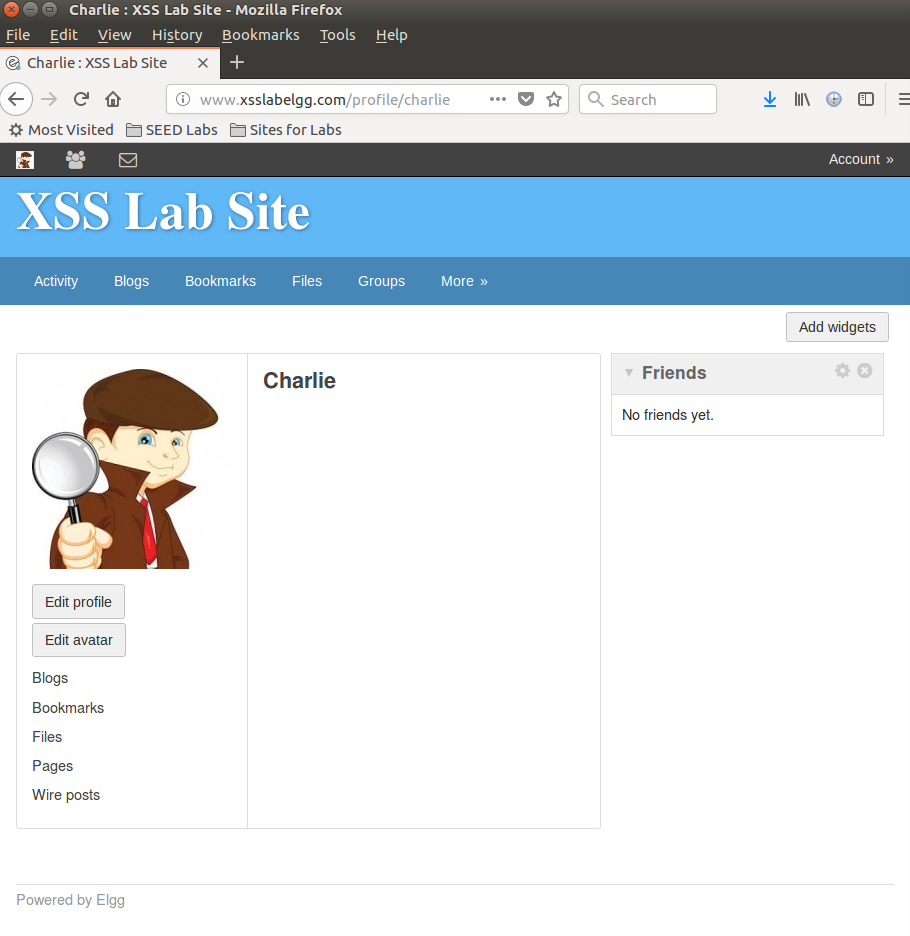
Here, we do the command in task 1, which is using the command to just link to our javascript file, which is called lab6.js.



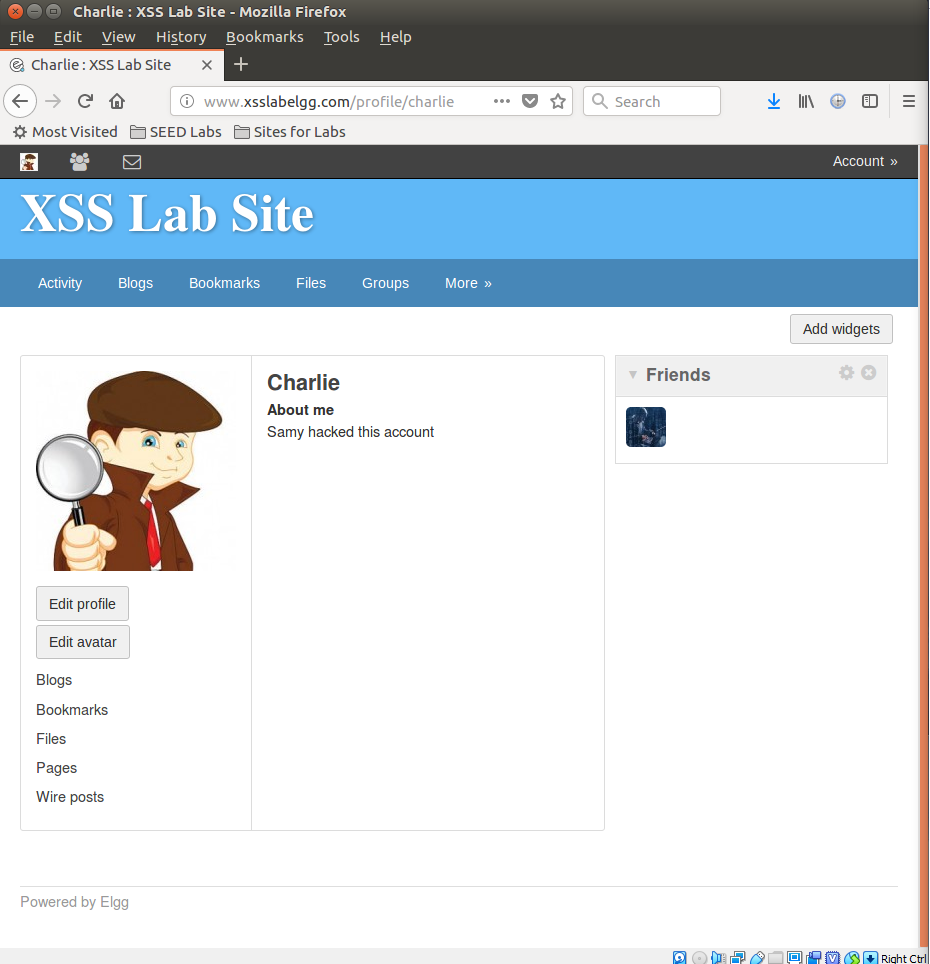
Screenshot of Alice’s account before we visit Samy’s page.



Went to Samy’s account and then came back to Alice’s. We can see that the “about me” has been populated with our message, which was in the screenshot of the code in this step, and, we see that Samy has been added as a friend on the right once again. This means that both attacks were successful.

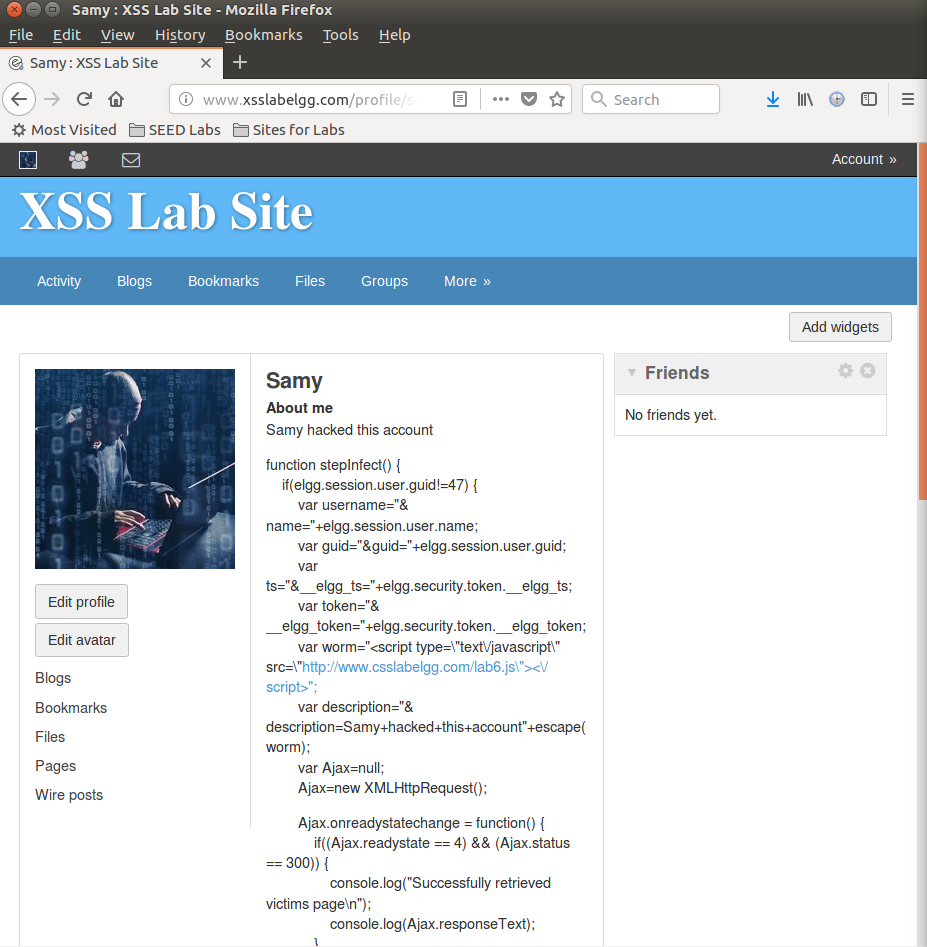


Here, we will use Charlie’s account as another account to make sure the worm can reach another profile. Here is a screenshot of his profile before we visit any other profiles.



Here is a screenshot of Charlie’s profile after we visited the other profiles. As we can see, this was successful as well because the “about me” and friends tabs have both been populated, like Alice’s account. This is because the worm is self-propagating, so first Alice is infected because she visits Samy, then Charlie visits Alice, so he gets infected as well.

**Task 7**



Once the HTMLawed countermeasure is turned on, we see all the script tags become disables and the code we implemented is shown on Samy’s page. The attack also doesn’t work because the tags are disabled as a security measure.

Now, after we uncomment the files, the attack still isn’t successful because HTML encodes our files with the special script characters, which is still blocked by the first countermeasure. In conclusion, turning HTMLawed on will cause it to not work, as will both countermeasures.