**Cross-Site Scripting Lab 2.0**

Table of Contents

[Environment Setup 3](#_Toc119738437)

[Task 1 6](#_Toc119738438)

[Task 2 9](#_Toc119738439)

[Task 3 11](#_Toc119738440)

[Task 4 13](#_Toc119738441)

[Question 1 14](#_Toc119738442)

[Question 2 14](#_Toc119738443)

[Task 5 15](#_Toc119738444)

[Question 3 18](#_Toc119738445)

[Task 6 19](#_Toc119738446)

[Task 7 31](#_Toc119738447)

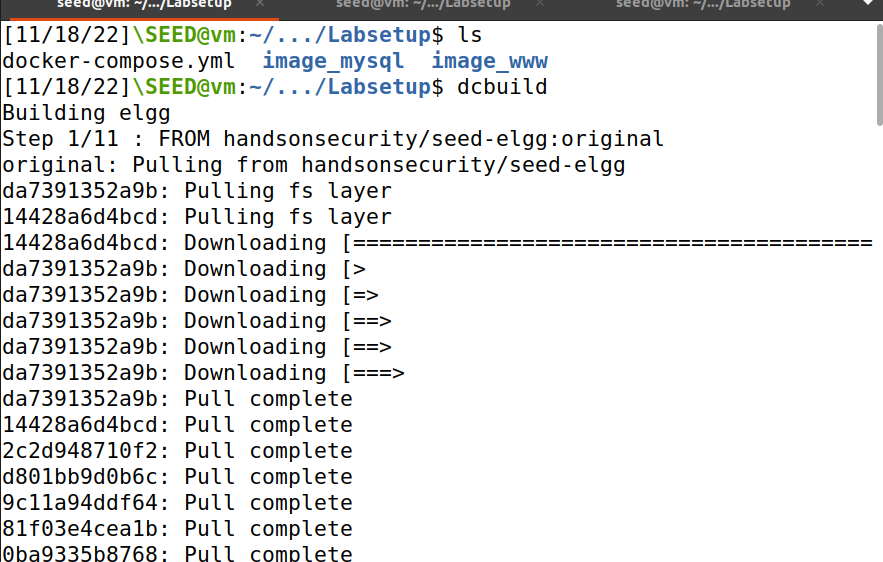
[Sub Task 1 and 2 31](#_Toc119738448)

[Subtask 3 and 4 34](#_Toc119738449)

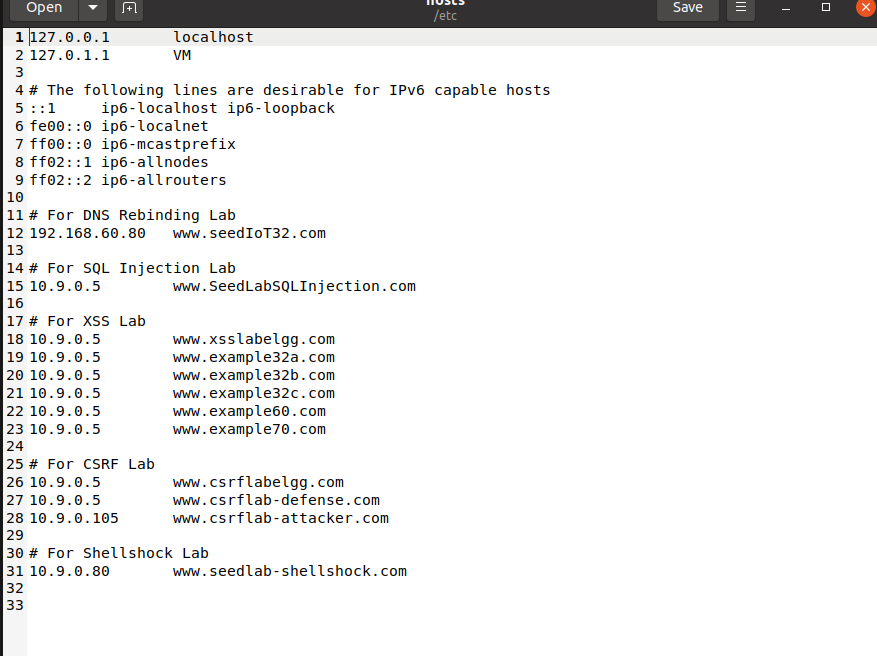
[Subtask 5 38](#_Toc119738450)

# Environment Setup

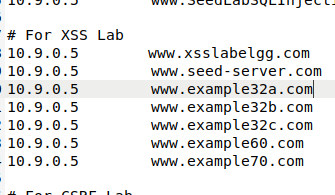
Building Dockers.



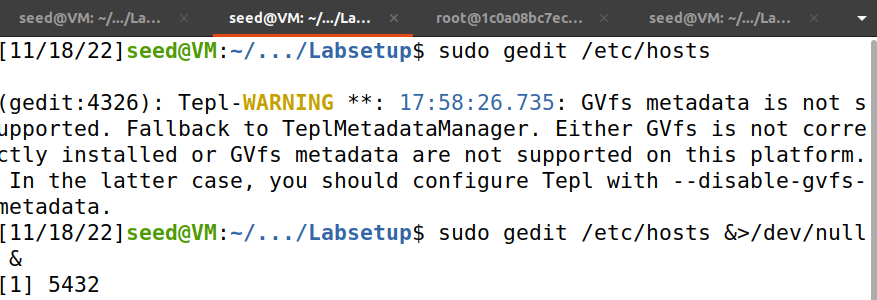
Hosts available in containers.



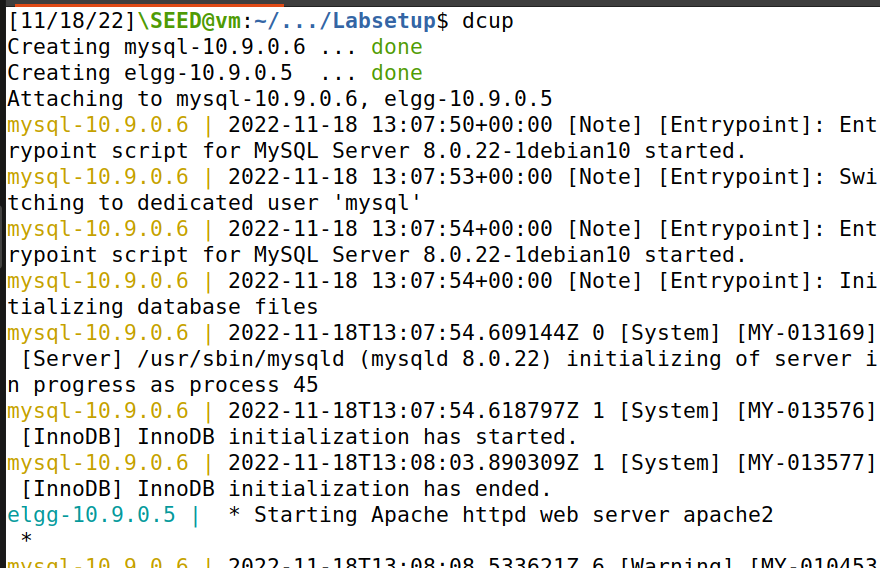
Now changing the domain.



Placing the hosts to work.



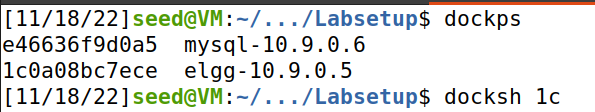
Setting up Dockers.



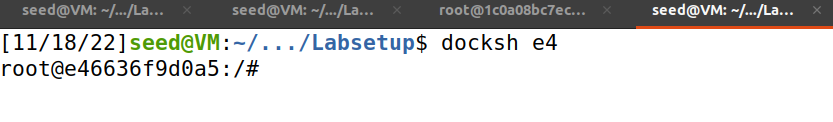
Docks available for use as provided in labsetup.



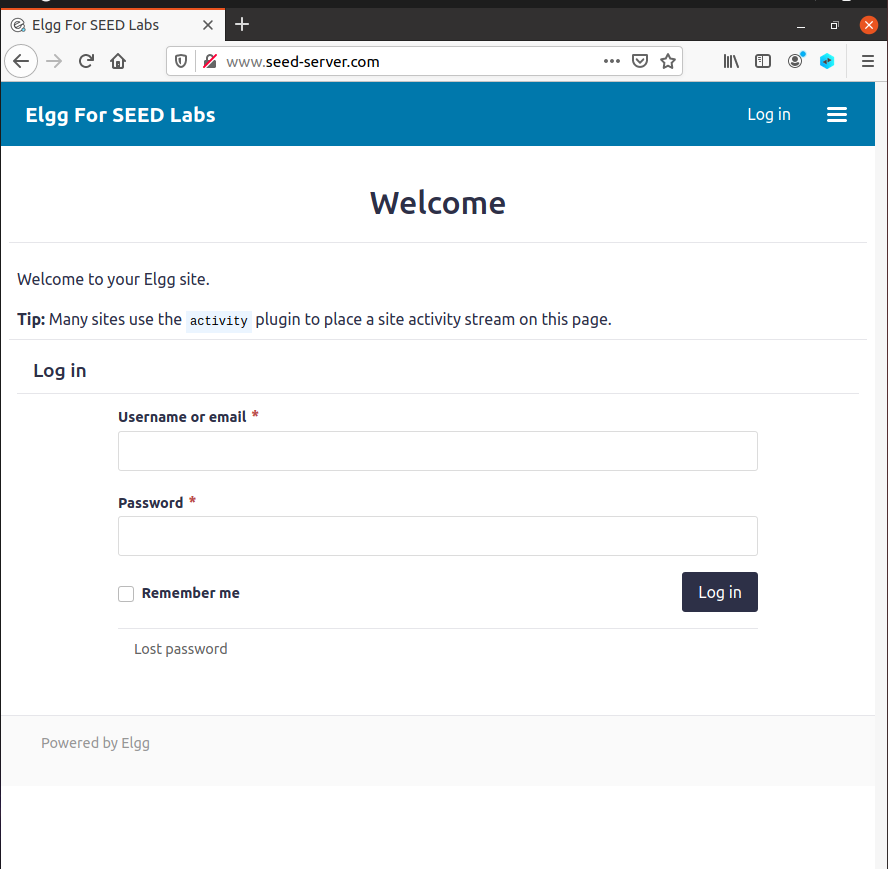
Setting up web server docker.



Setting up database docker.

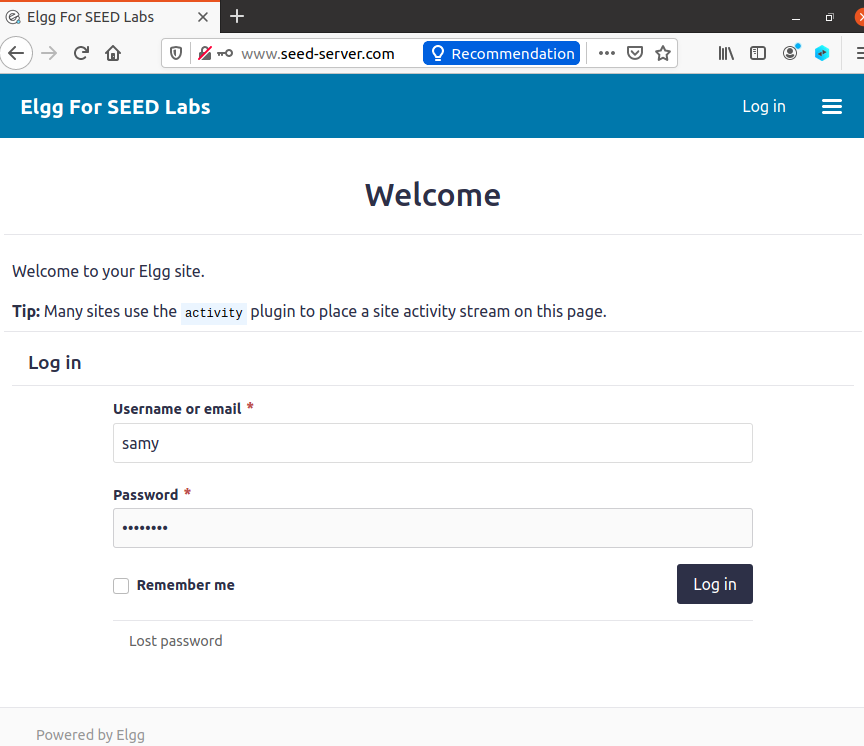


Now the machine is up and running.

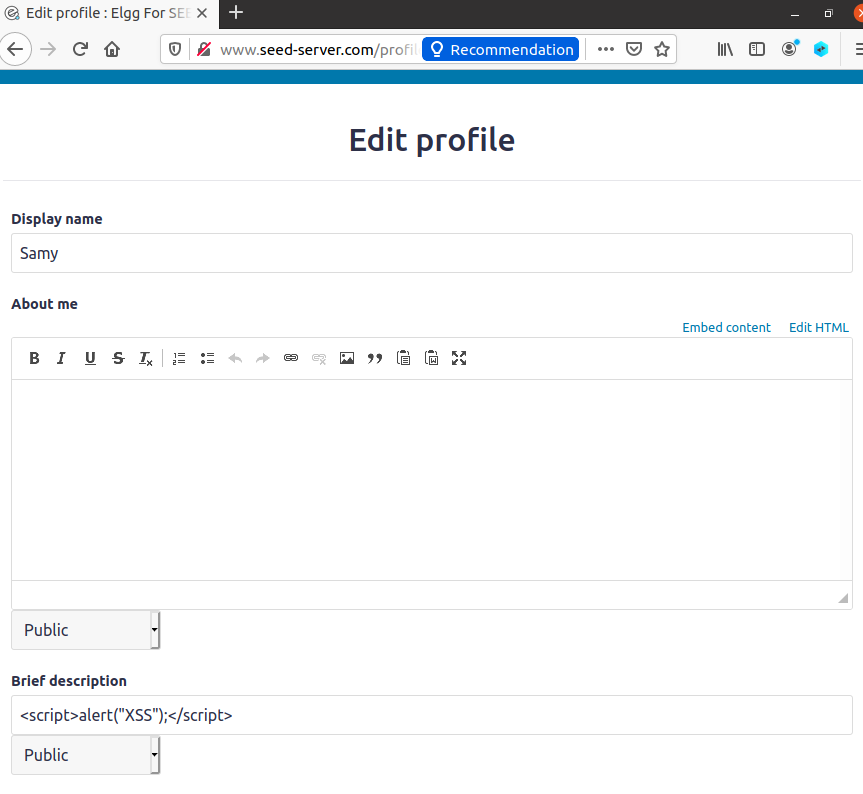


# Task 1

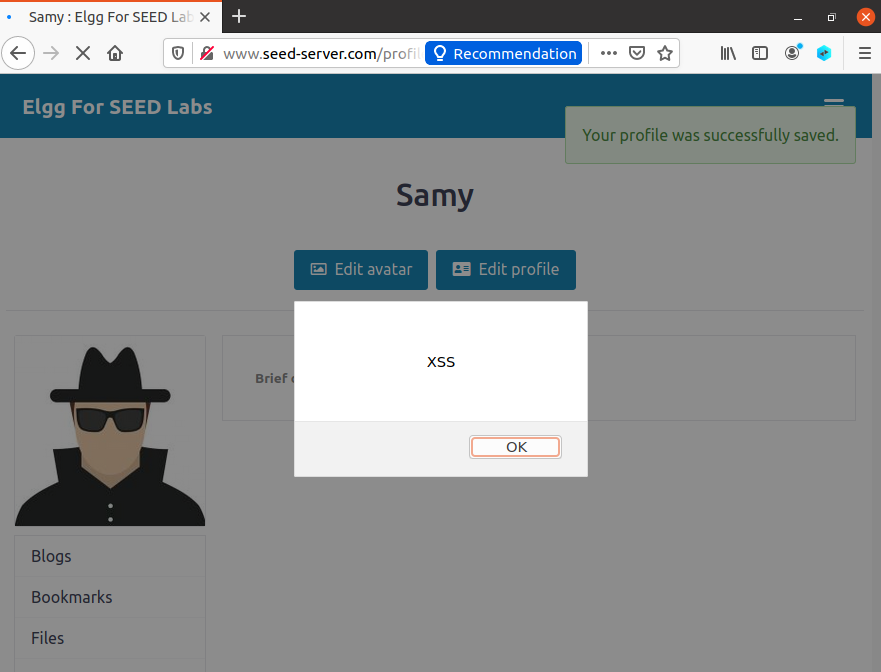
Logging in as Samy



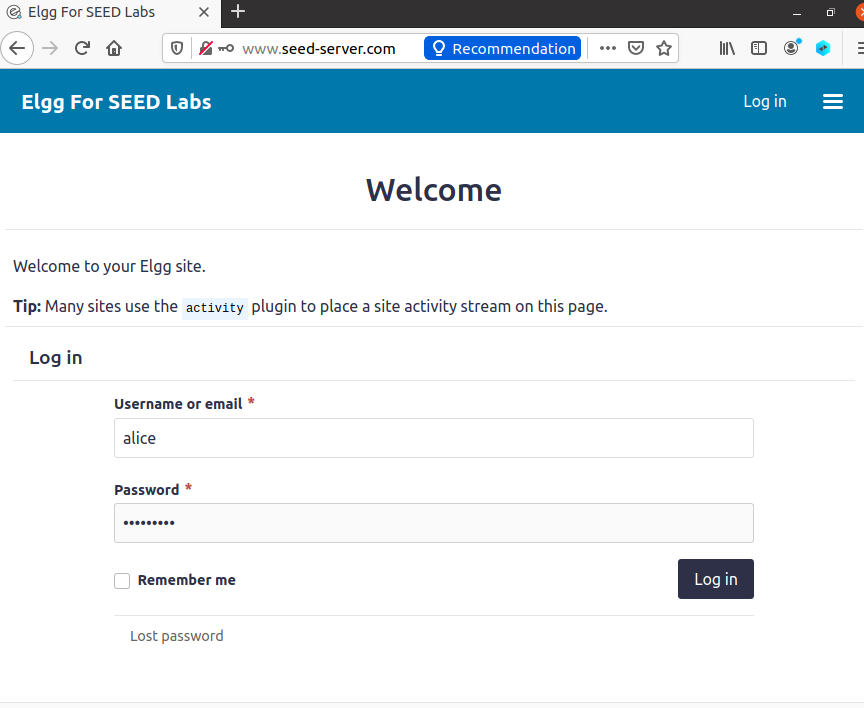
Editing the profile where I added a script.



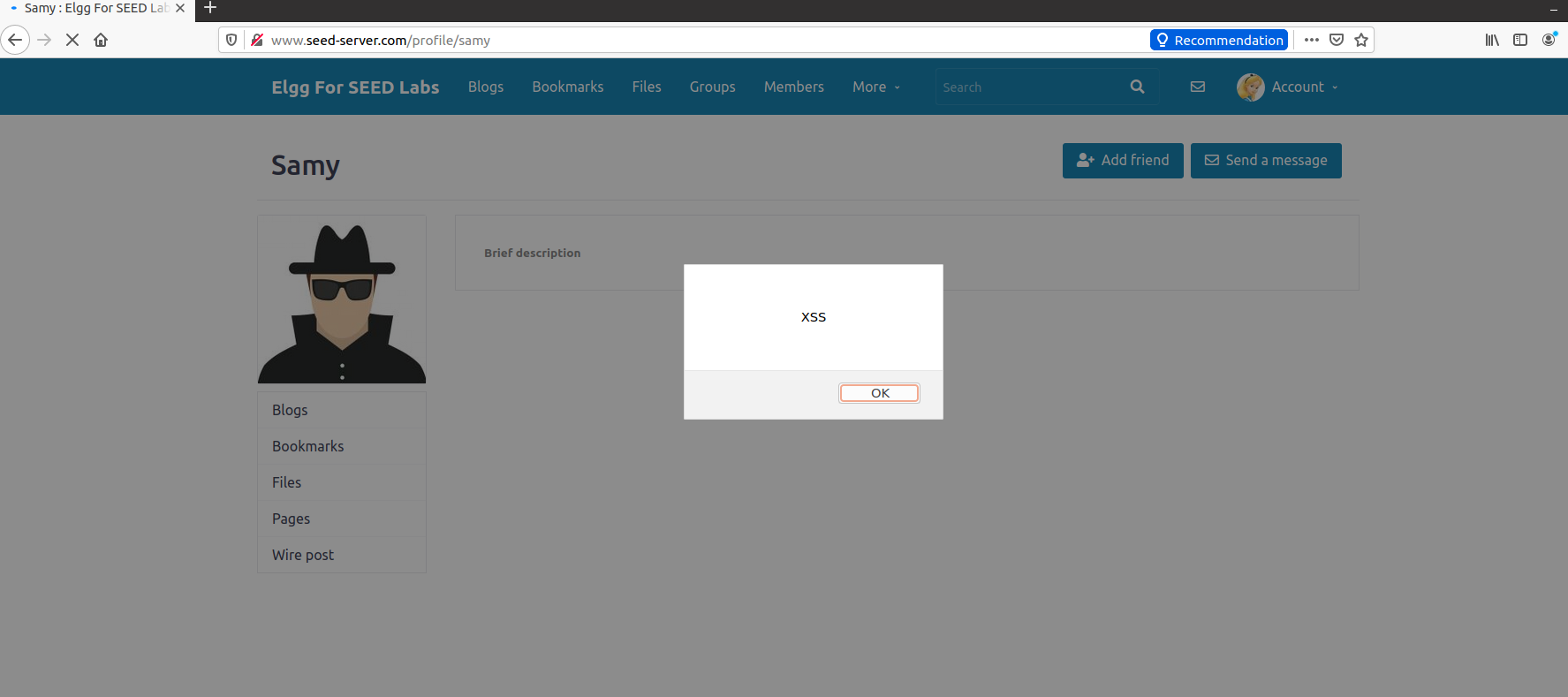
And I got the alert as soon as we save the changes.



Now logout from Samy’s profile and login as Alice.

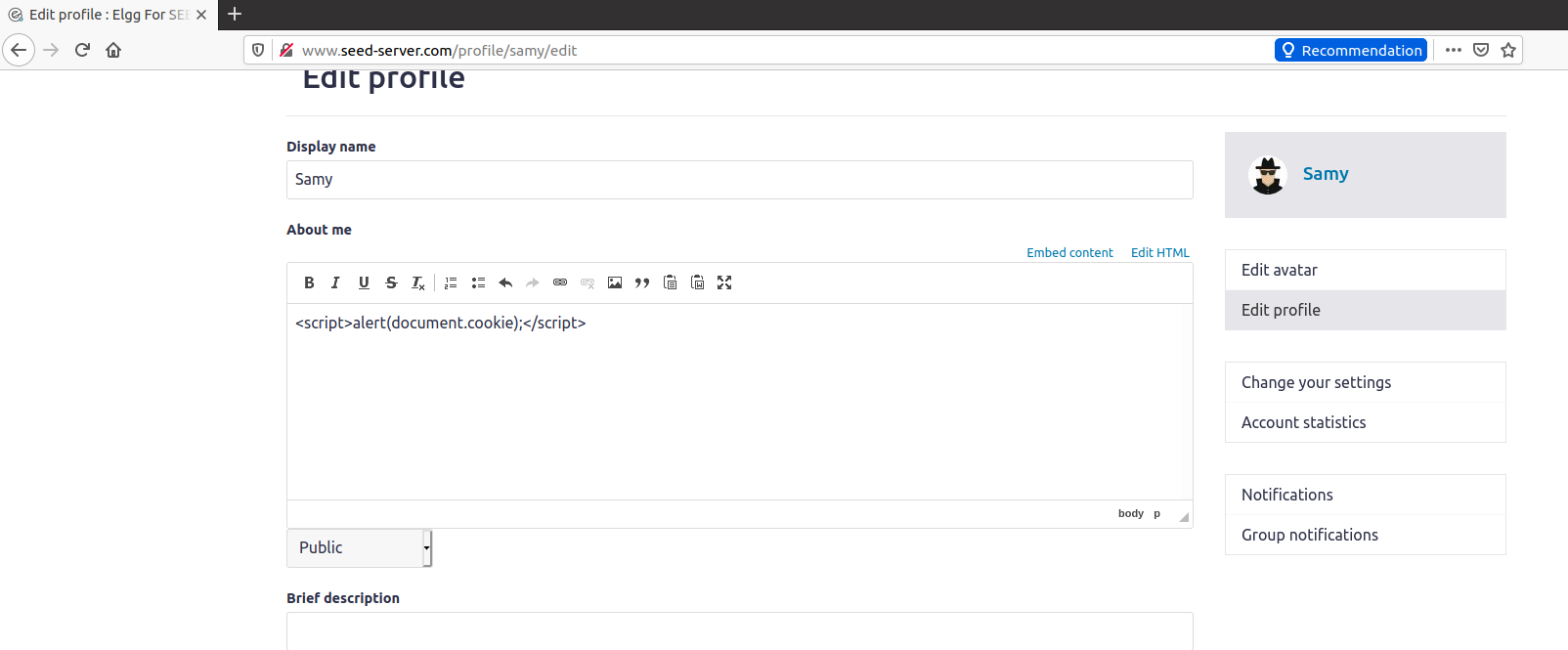


After making a search and opening Samy’s profile we receive the same script we added in brief description of Samy’s profile.

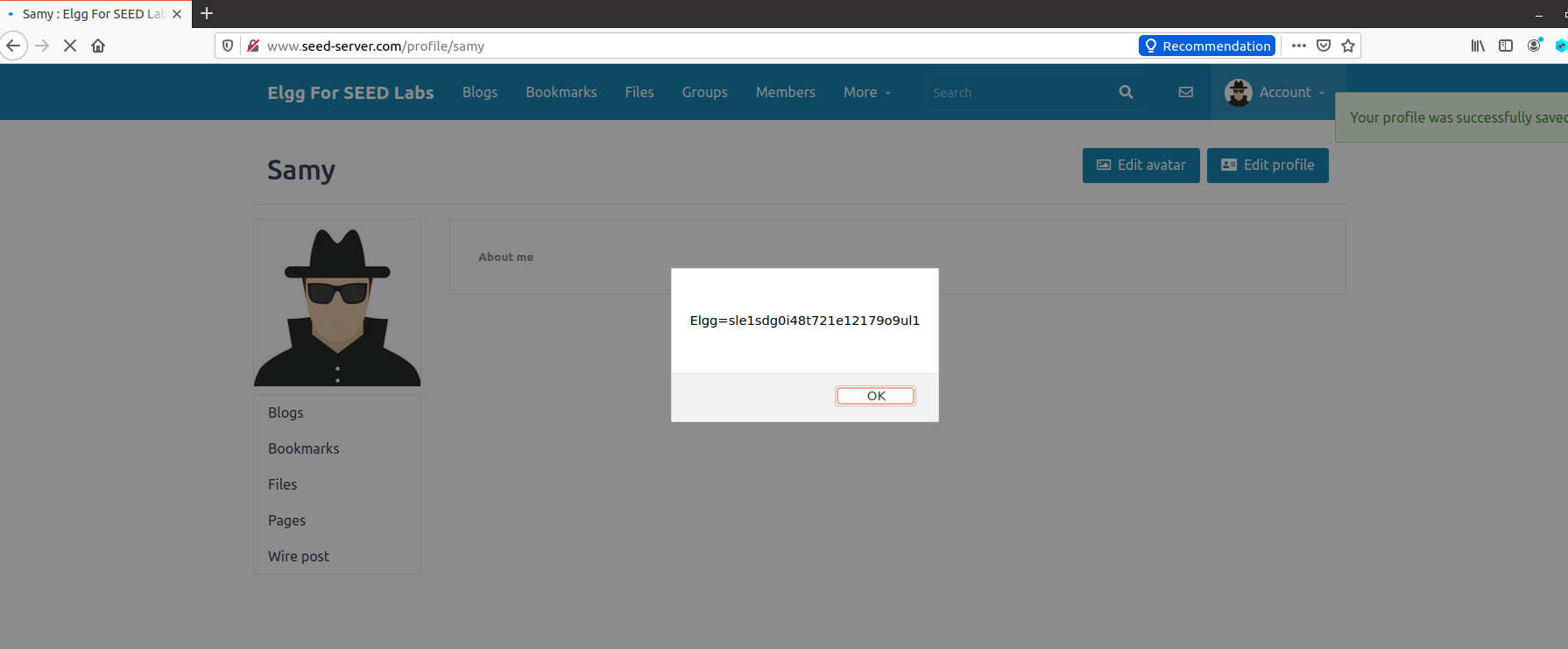


# Task 2

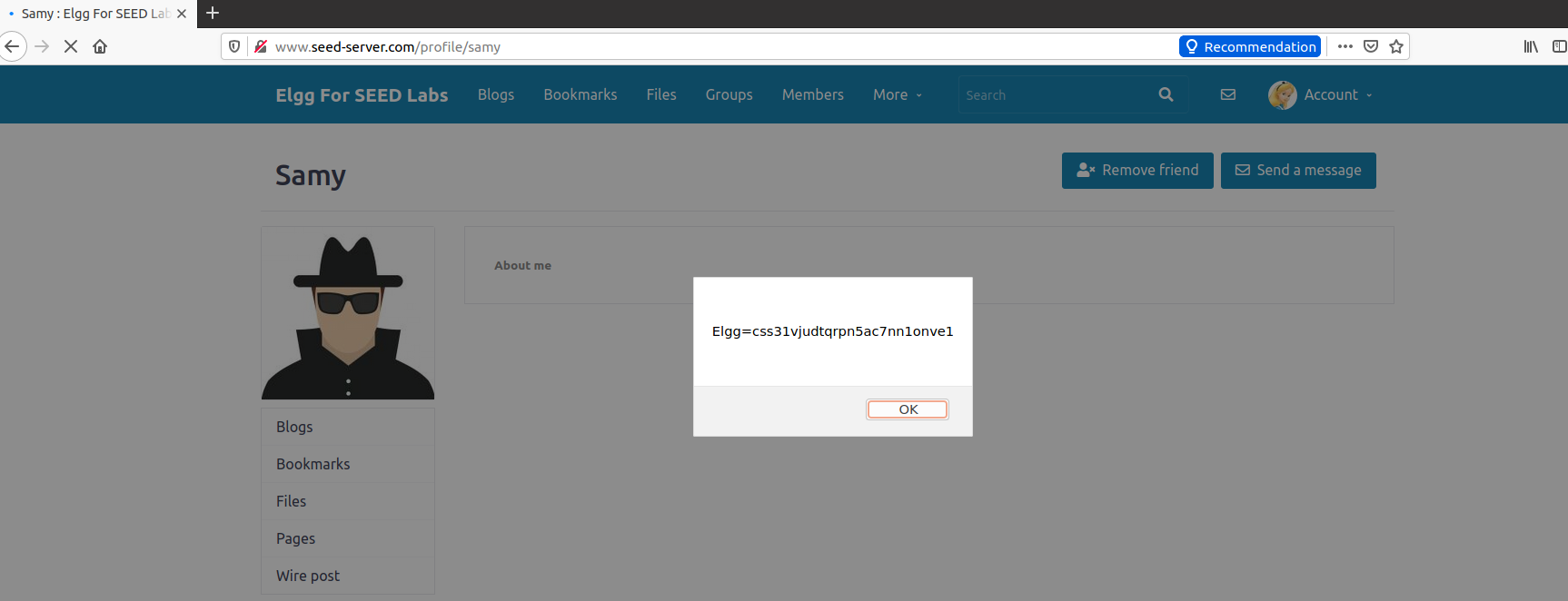
Writing script in About me.



Upon saving it is visible that the script is working.

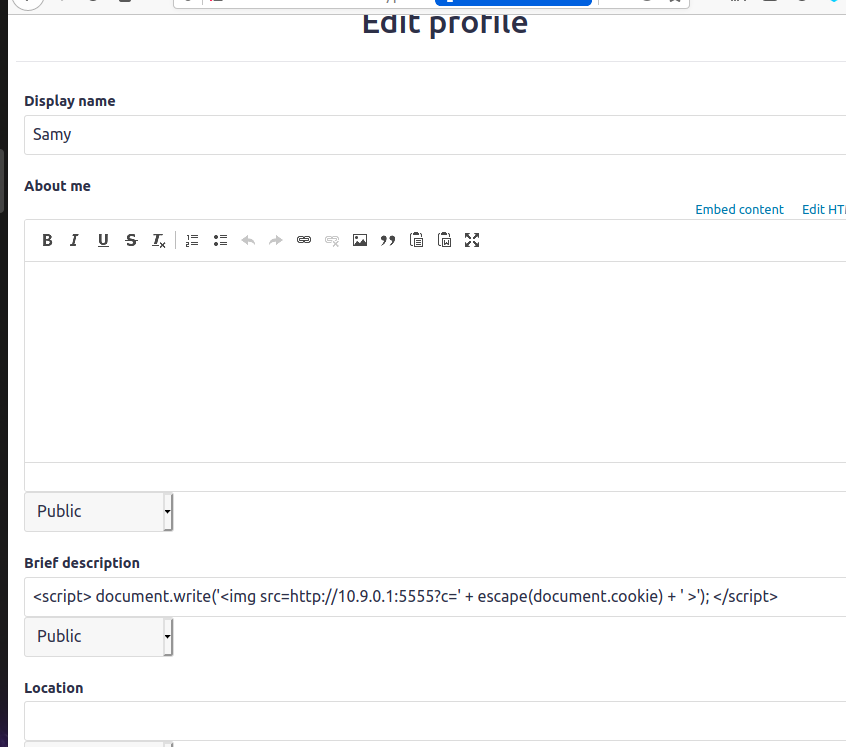


Now when we visit the Samy’s profile from Alice’s account we receive the cookie alert.

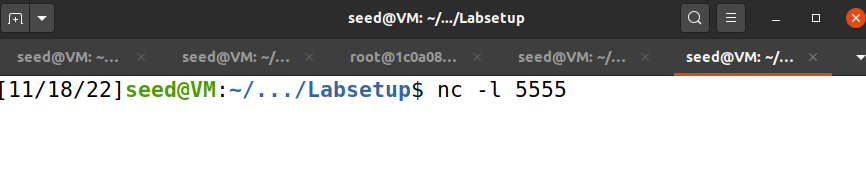


# Task 3

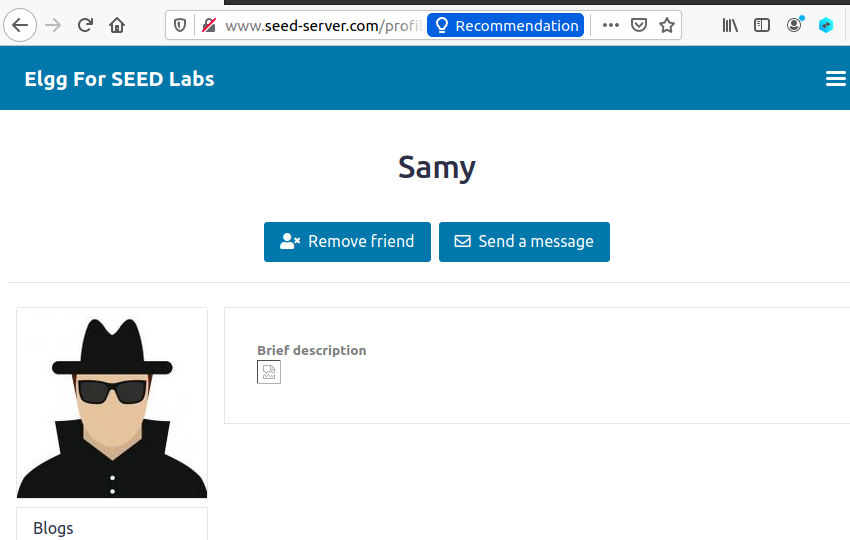
Enter the script provided in the manual.



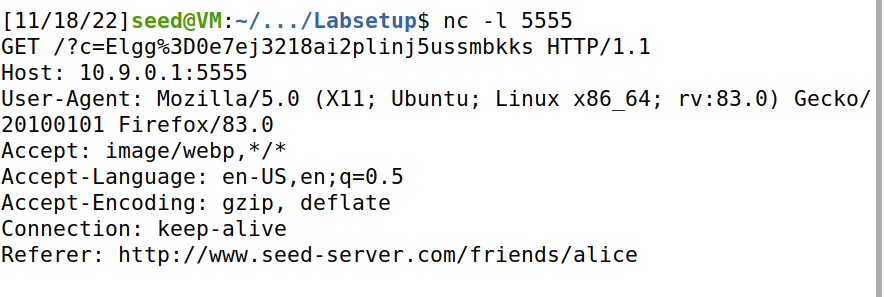
Listening on port 5555.



Visiting Samy’s profile as Alice.

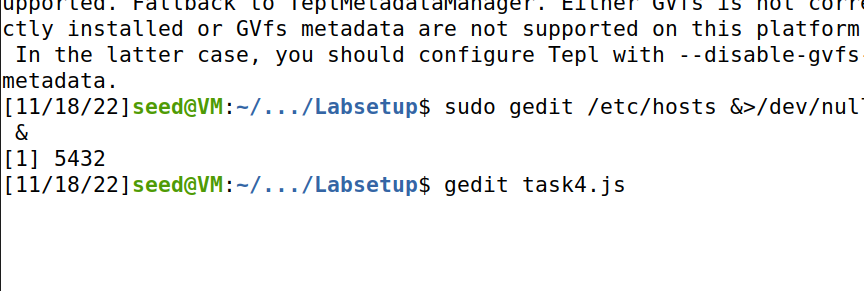


And when we look back at the terminal we can see the captured cookie details caught.

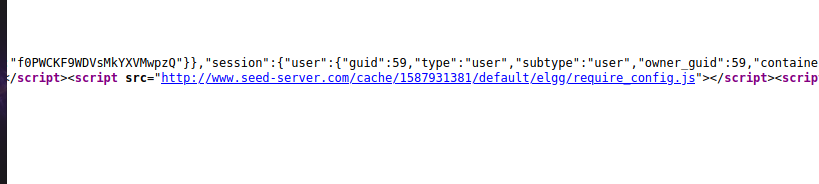


# Task 4

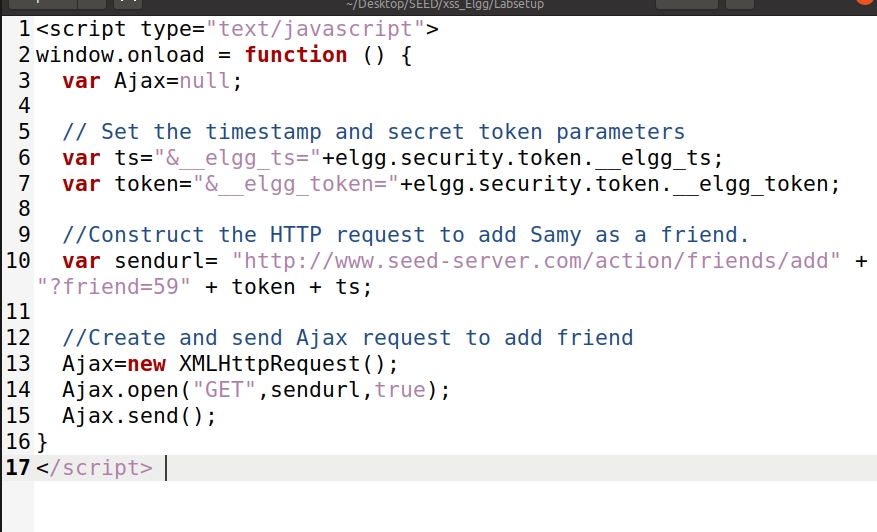
Creating js file for the script.



Checking the id for Samy.



The script now looks like this.



Now I added the modified script in the About me section of Samy’s profile.



And after checking friend’s list and switching between accounts the add friend doesn’t work except for adding himself as friend.

## Question 1

In order to send a valid HTTP request, we need to have the secret token and timestamp value of the website attached to the request, or else the request will not be considered legitimate or will probably be considered as an untrusted cross-site request and hence will throw out an error with our attack being unsuccessful. These desired values are stored in JavaScript variables and using the lines 1 and 2, we are retrieving them from the JS variables and storing in the AJAX variables that are used to construct the GET URL.

## Question 2

If that were the case, then we will not be able to launch the attack anymore because this mode

Encodes any special characters in the input. So, the < is replaced by &lt and hence every special

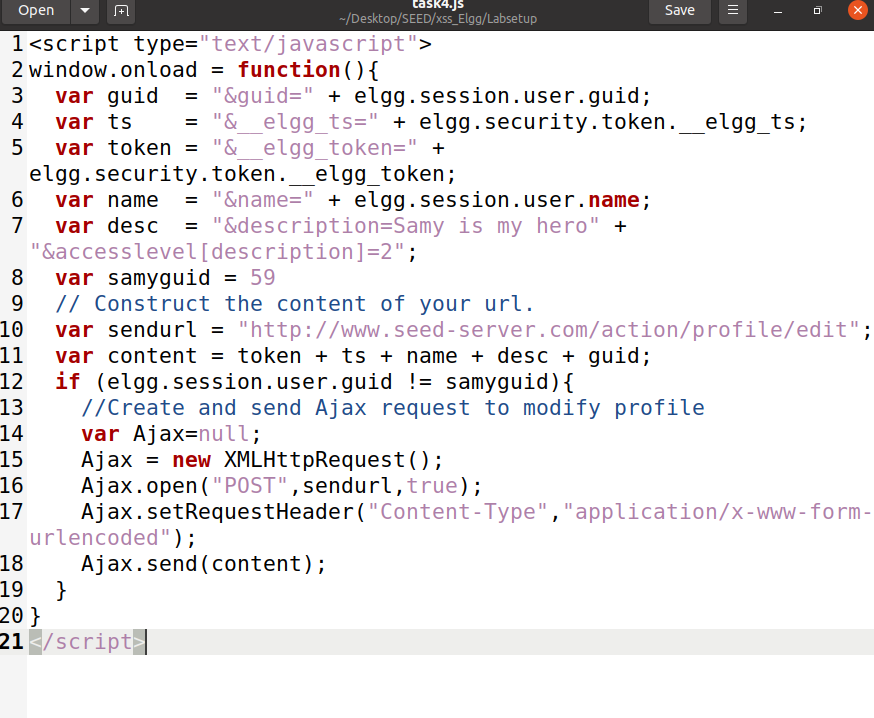
Character will be encoded. Since, for a JS code we need to have <script> & </script> and various

Other tags, each one of them will be encoded into data and hence it will no more be a code to be

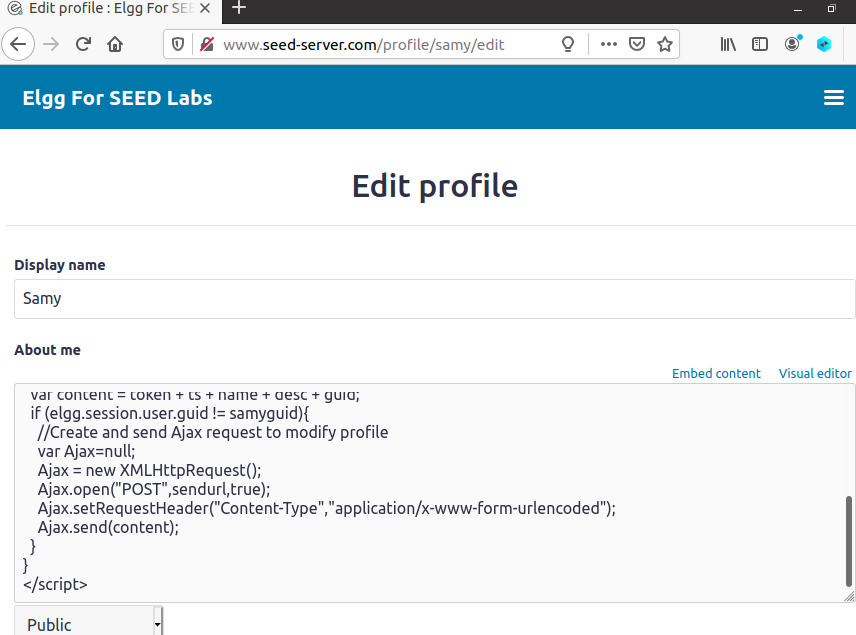
Executed.

# Task 5

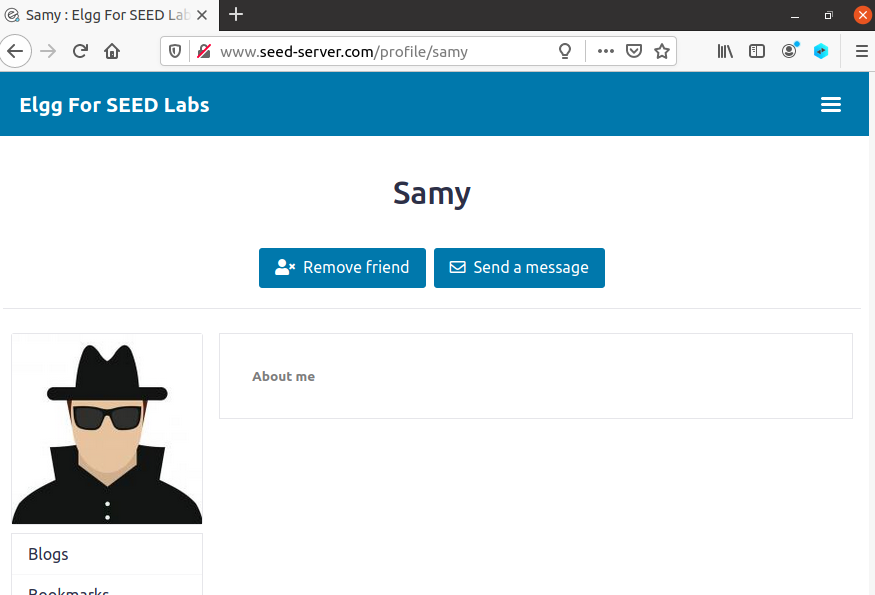
Modifying the code as per instructions.



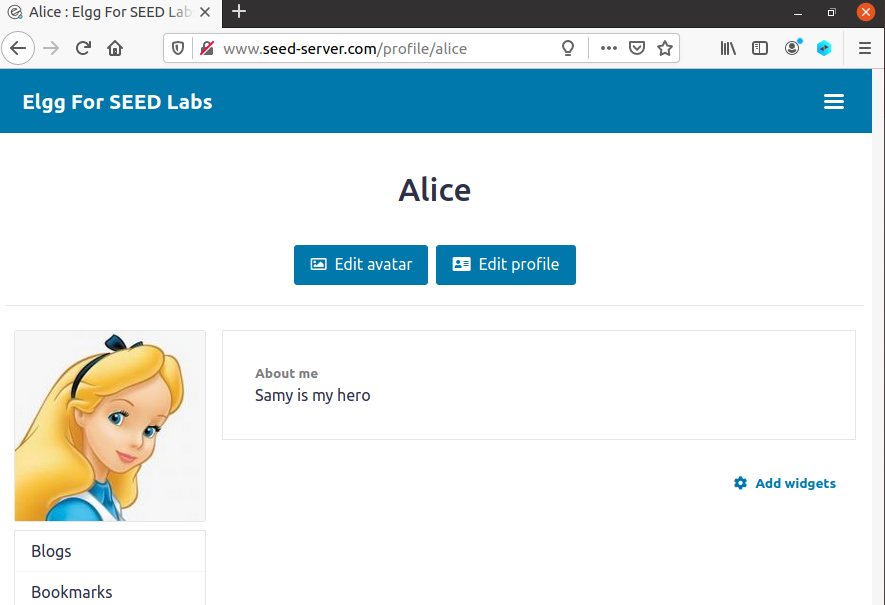
Adding the script to About me section of Samy’s profile.



Now I logged in as Alice and went to Samy’s profile and then noticed Samy as added as friend but that doesn’t display in Samy’s logged in profile that Alice is a friend.



Moreover, It is noticeable on Alice’s profile that the script worked as the string, “Samy is my hero” is being displayed in her About me section.

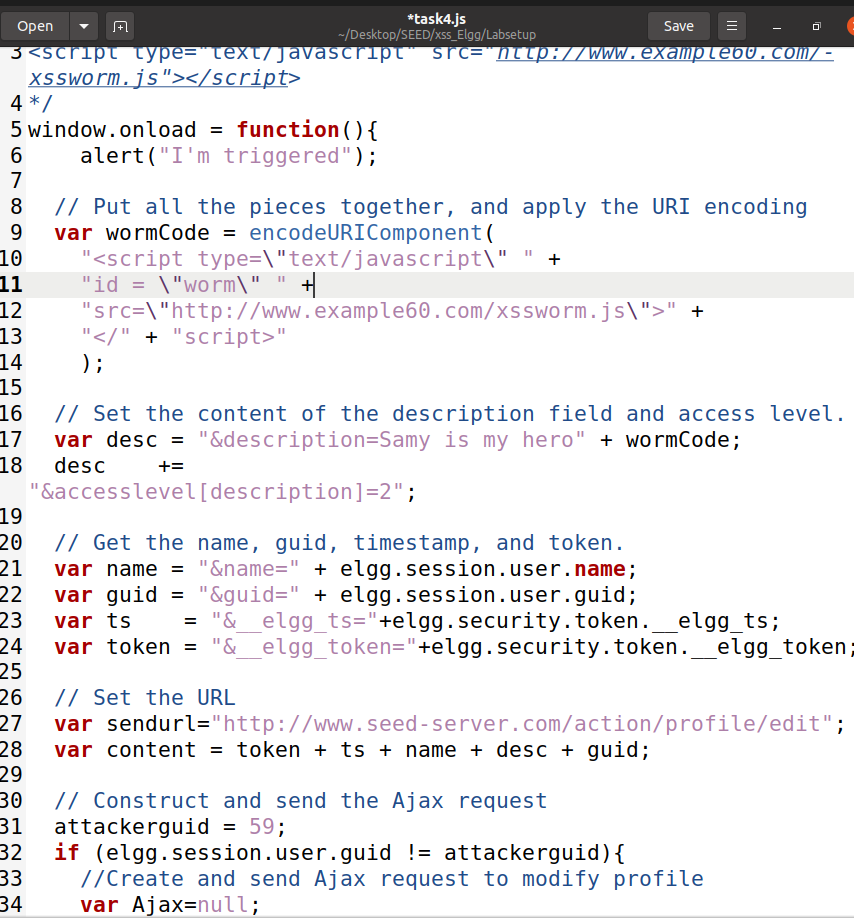


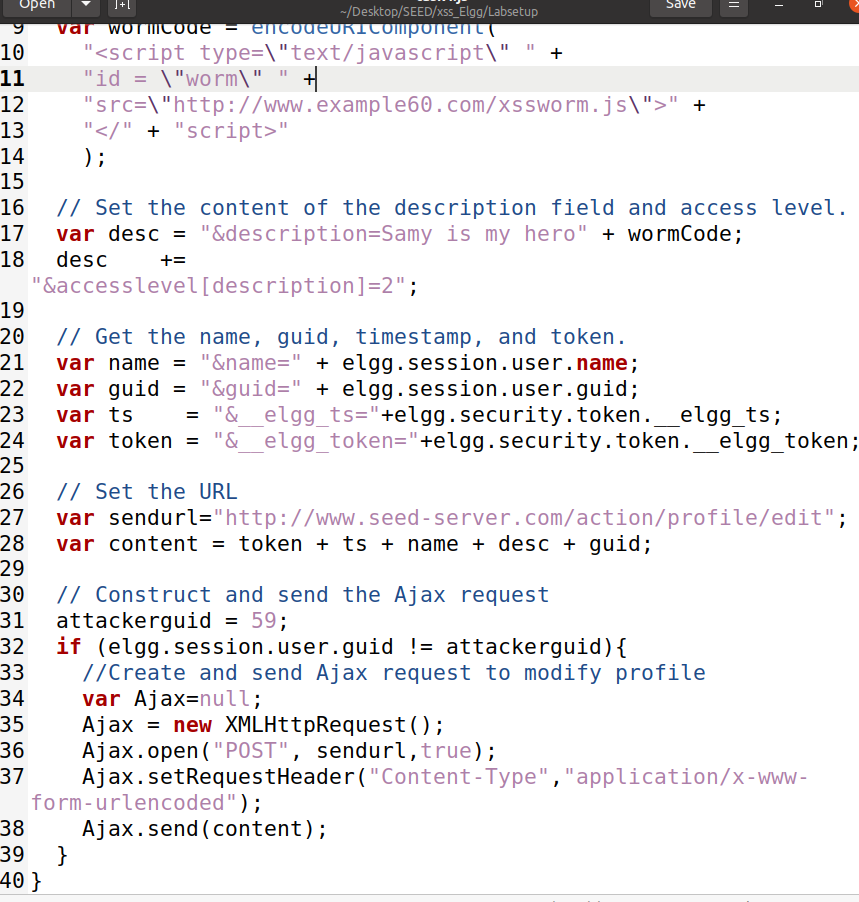
## Question 3

Line 1 is needed so that Samy does not attack himself and we can attack other users. The JS code obtains the current session’s values and stores a string named “Samy is my hero” in the about me section. Now, since we have the JS code in about me section, and if we did not have that line, as soon as the changes are saved, the JS code is executed and this JS code will enter “Samy is my hero” in the About me field of the current session i.e. Samy. This will basically replace the JS code with the string, and hence there won’t be any JS code to be executed whenever anyone visits Samy’s profile.

# Task 6

Writing the self-propagating worm.

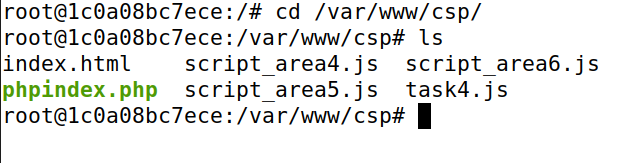




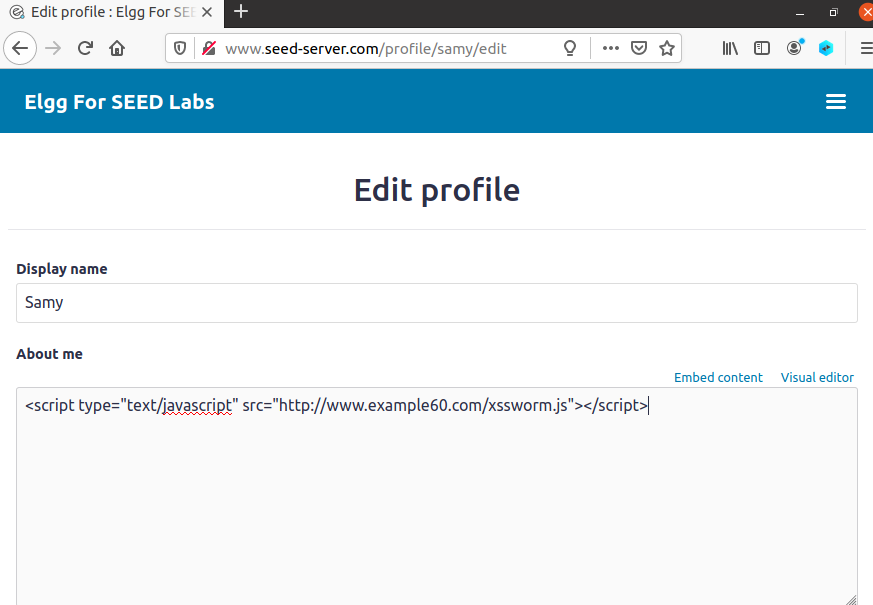
Copying the code to Web Server Docker.



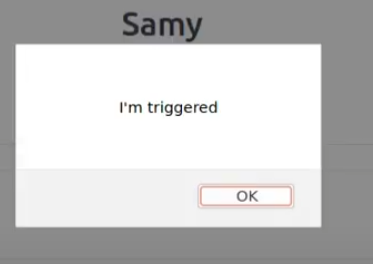
And it is visible in the Web Server Docker in the twitter directory.



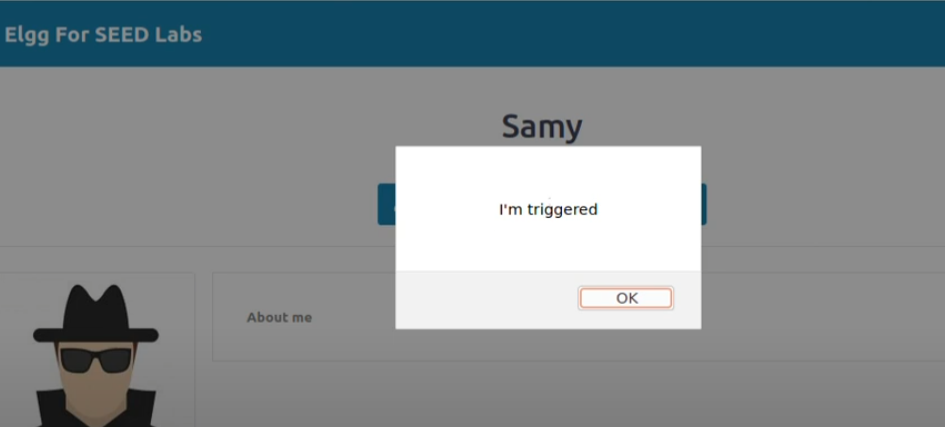
Now putting the script in About me Section of Samy’s profile.



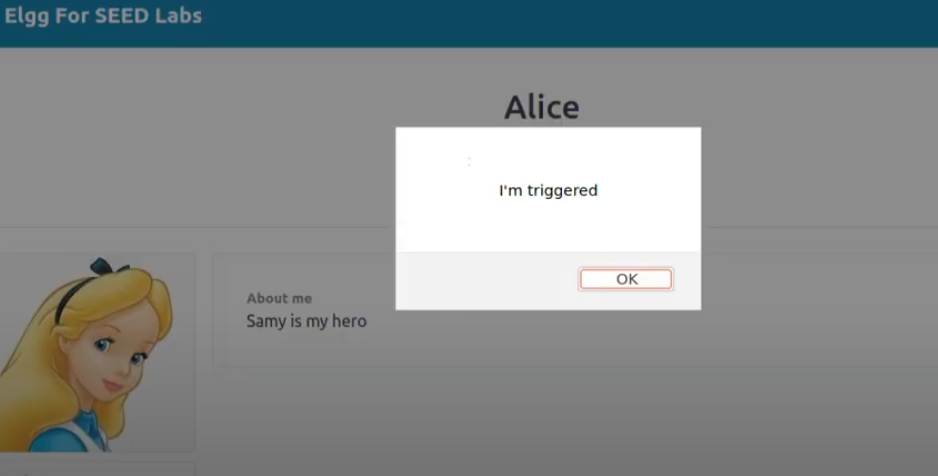
Now we get the alert which means script is active and won’t attack Samy.



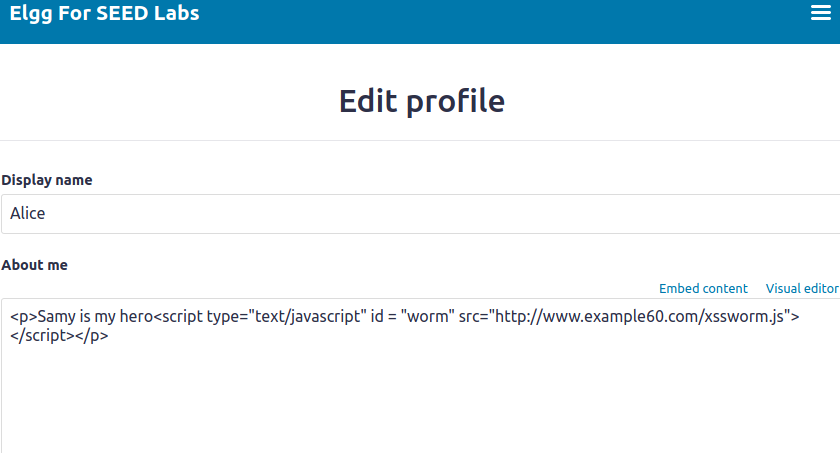
Now logging in as Alice I entered Samy’s profile and this happened.



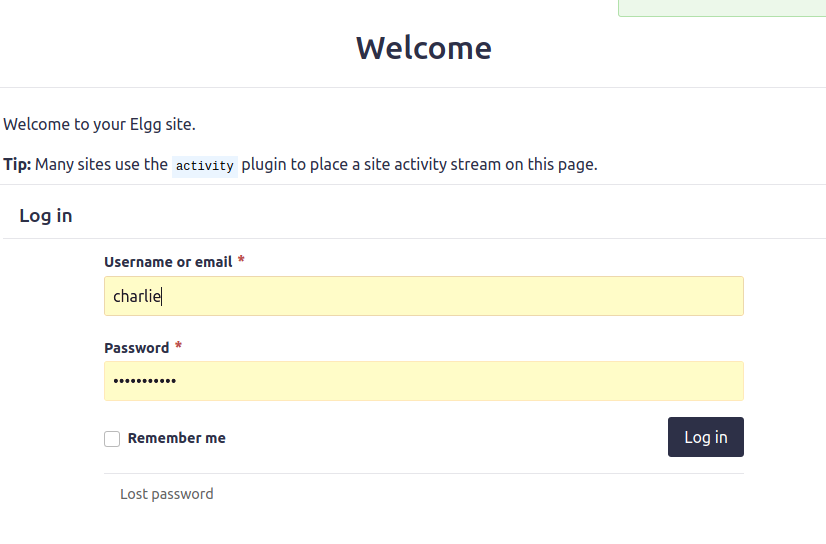
When I checked back Alice’s profile logged in as Alice. Now it seems like the script is active here as well.



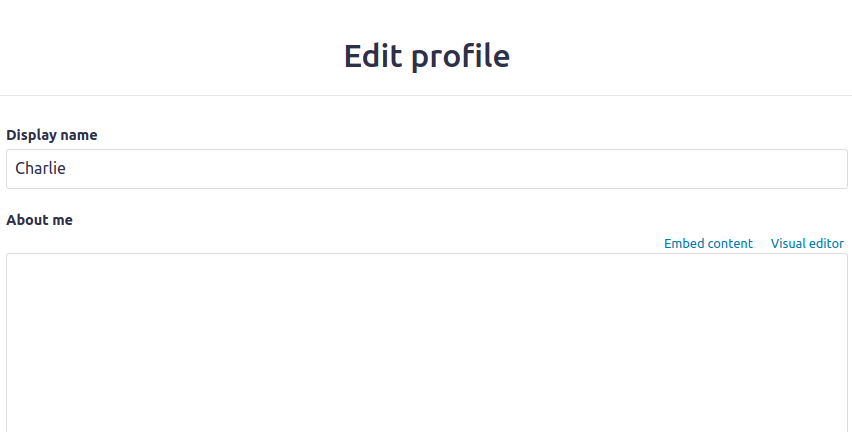
While checking the profile I found in HTML editing that the script made its way here.



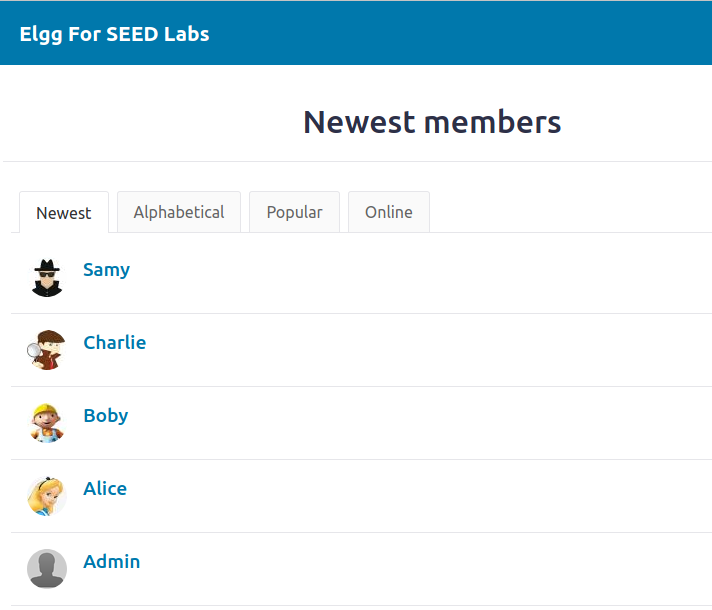
For confirmation logging in as Charlie.



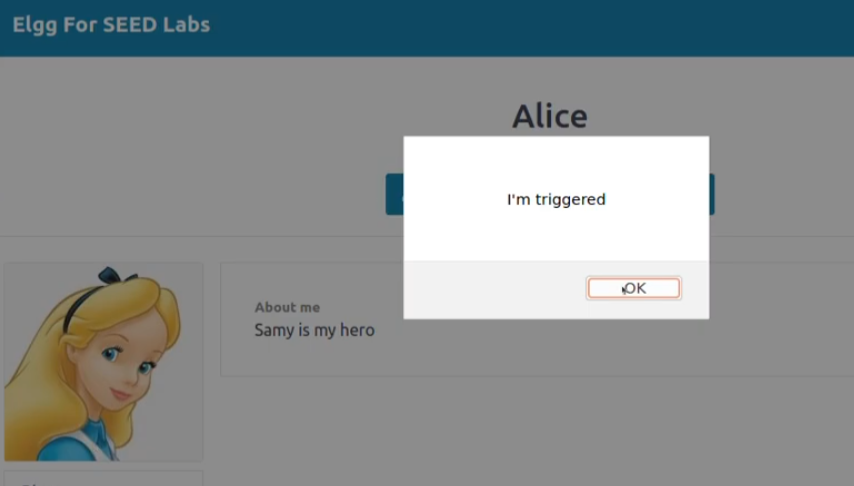
Profile here is clear of any content.



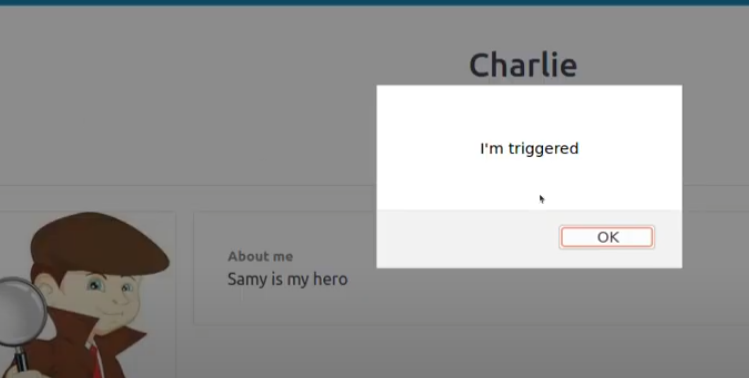
Checking Members.



When entering Alice’s profile a trigger appeared.



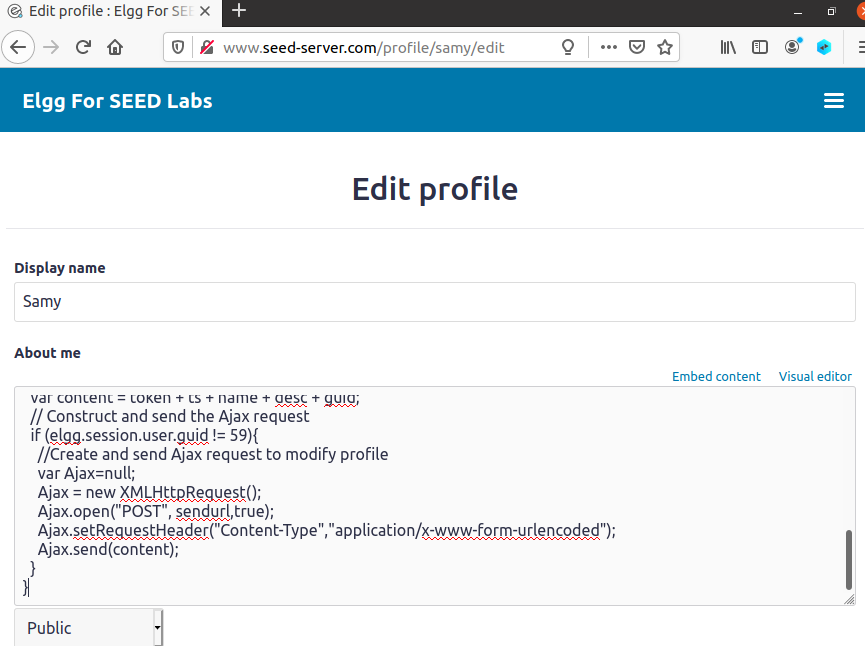
When heading back to the profile of Charlie we can see he is also effected by the worm. Moreover, the message, “Samy is my hero” and the worm script is also present.



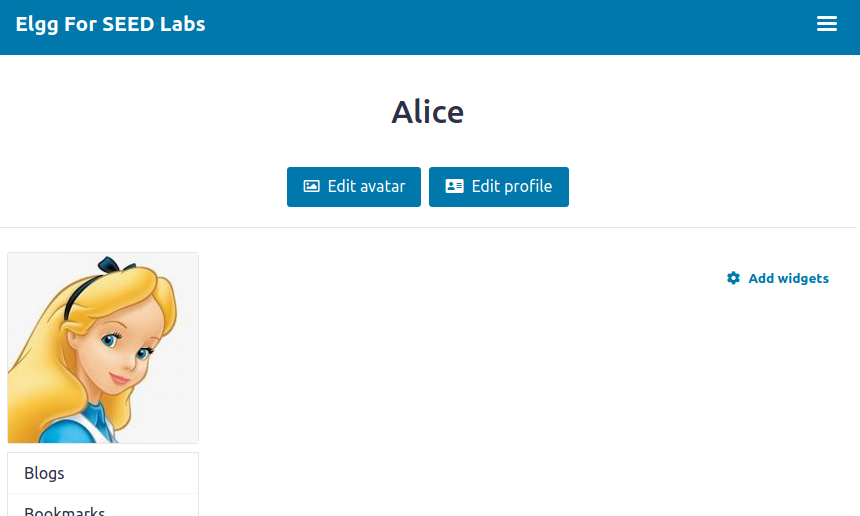
Now going for DOM Approach and here I modified the code.



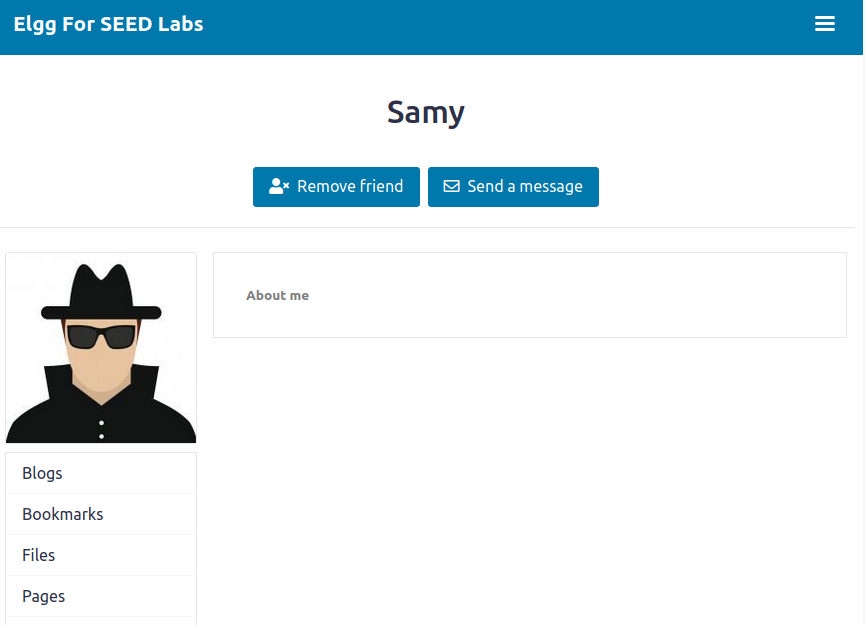
Now Adding the Script to the About me Section of Samy’s account profile.



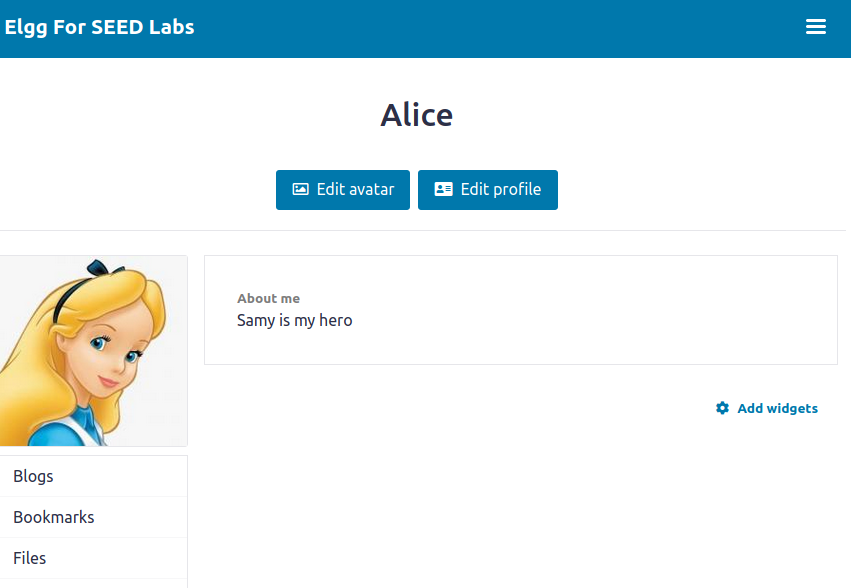
Logging in and clearing previous changes on Alice.



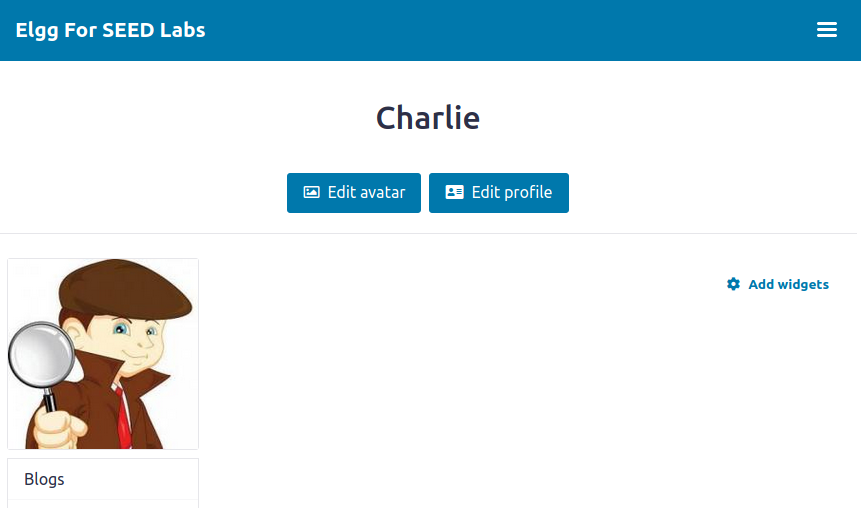
Now visiting Samy’s profile as Alice.



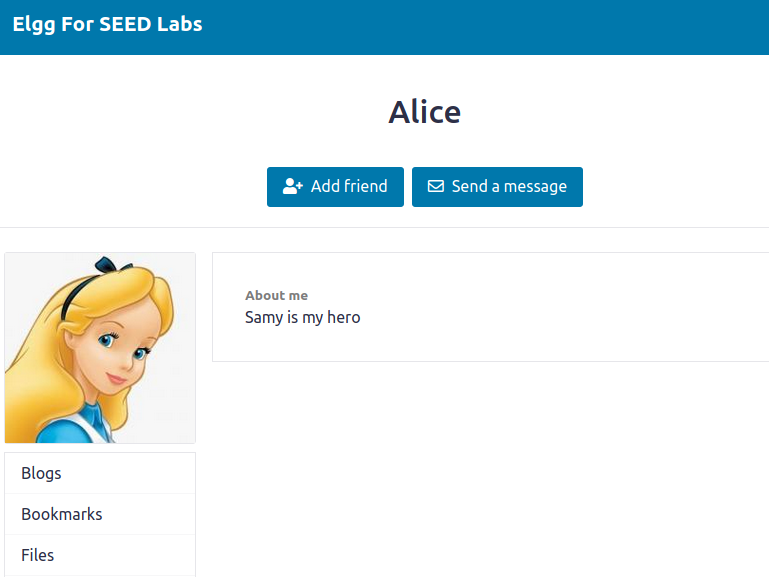
Coming back to Alice’s profile it is noticeable that the worm worked and Alice is infected as the string “Samy is my hero” is being displayed on Alice’s profile.



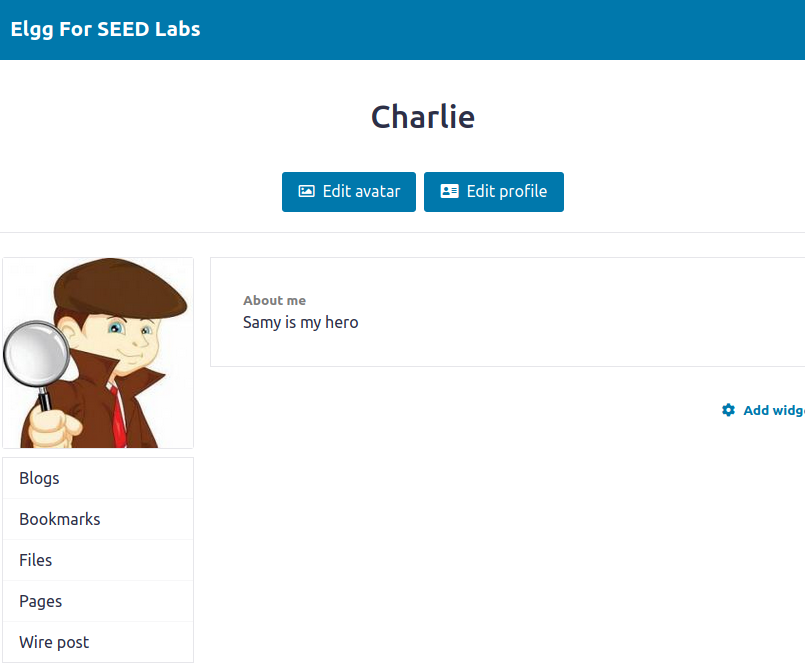
For more verification logging in as Charlie.



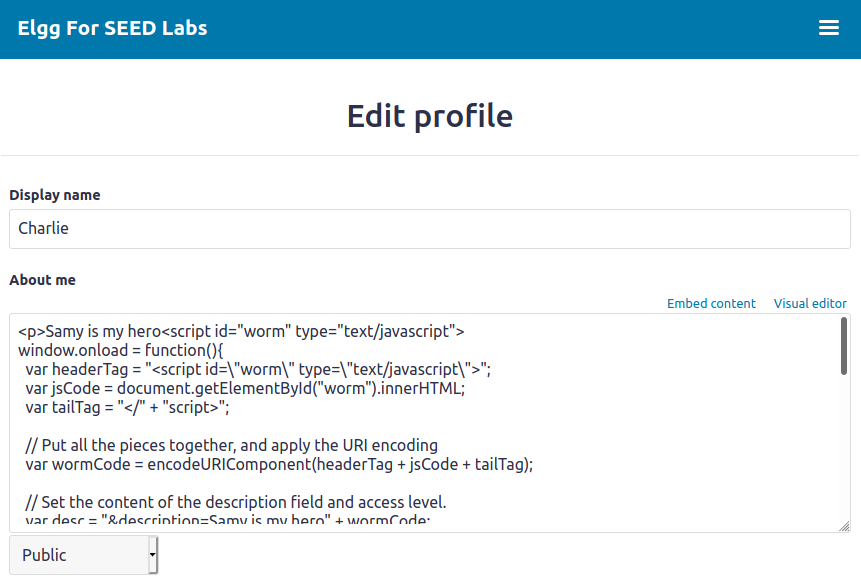
Visiting Alice’s profile.



And when coming back to Charlie’s own profile it got infected.

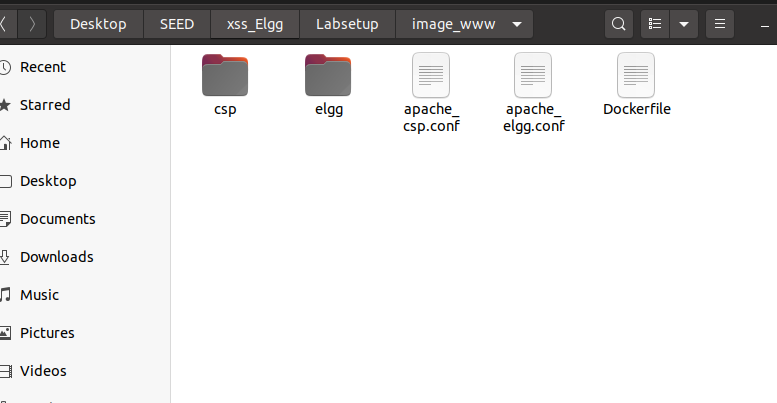


Even when checking the HTML editor in Charlie’s profile the worm script is discovered there.



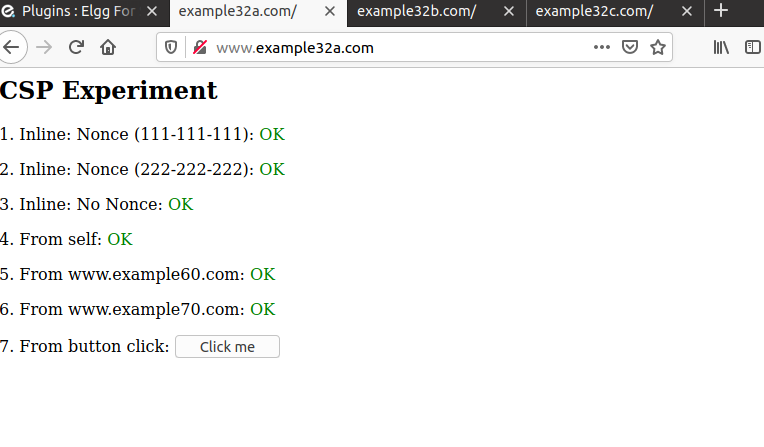
# Task 7

Open the file Apache.conf from File Explorer and the files in csp folder in an editor.

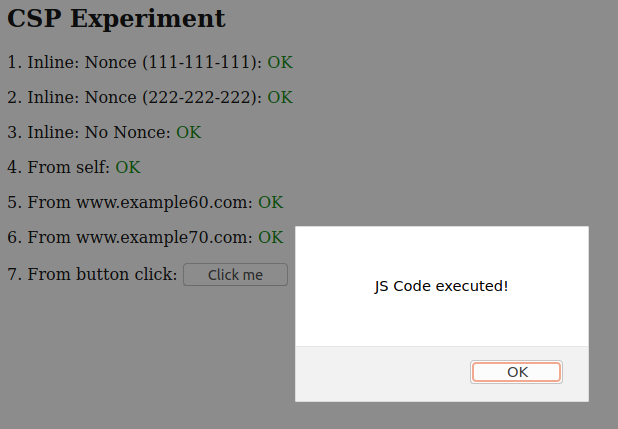


### Sub Task 1 and 2

For [www.example32a.com](http://www.example32a.com) we can see everything is clear.



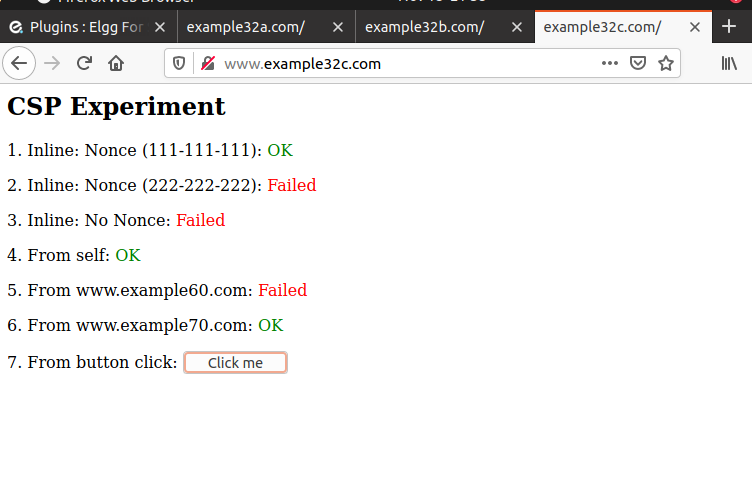
When pressing the button an alert pops up.



For [www.example32b.com](http://www.example32b.com) we can see only area 4 and 6 is clear and when clicked on “Click me” button nothing happens.



For [www.example32c.com](http://www.example32c.com) we can see only area 1, 4 and 6 is clear and when clicked on “Click me” button nothing happens.

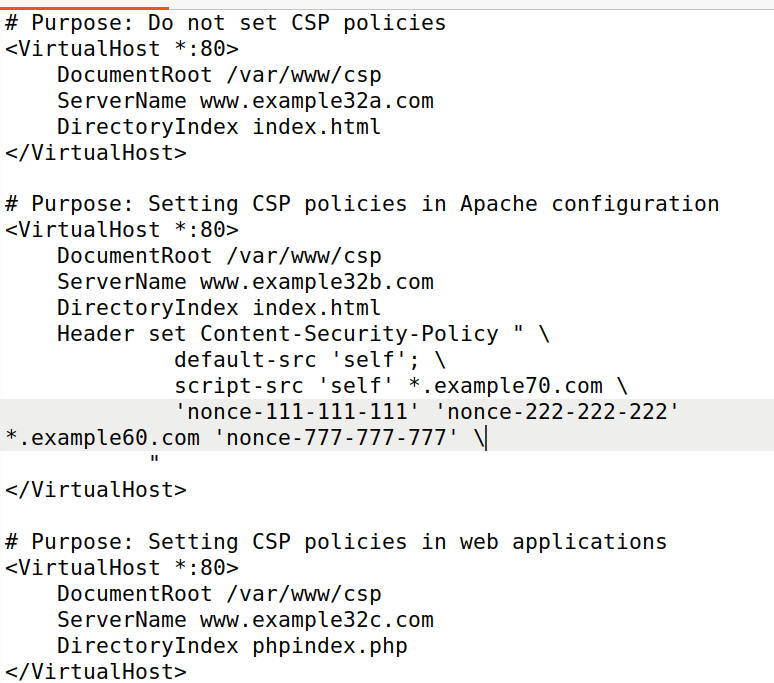


### Subtask 3 and 4

Making some changes in index.html.



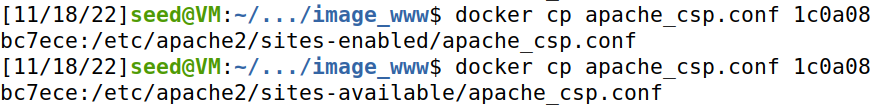
Making some changes in apache\_csp.conf file.



Also making changes in phpindex.php.



Copying the file to Web Server Docker to enabled and available sites..



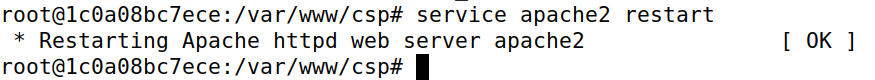
Copying index.html to Web Server Docker.



Copying phpindex.php to Web Server Docker.



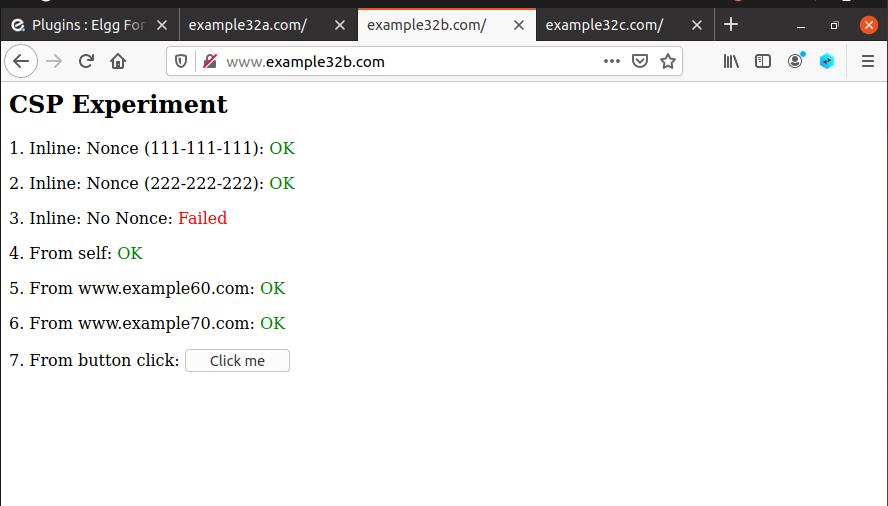
Restarting the Apache2 Service.



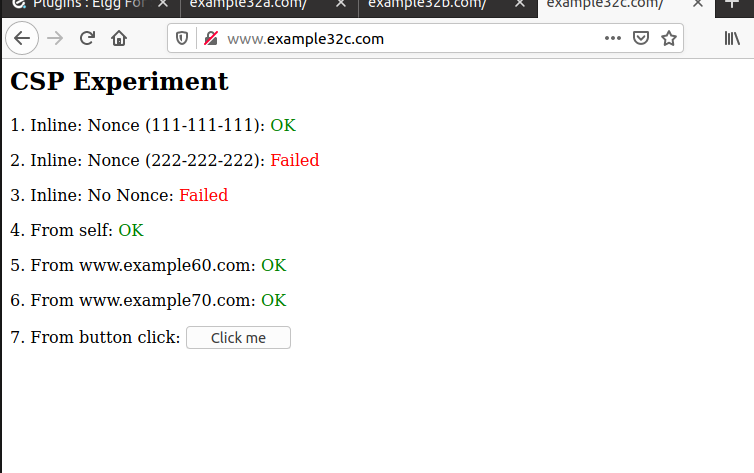
Moreover, checking changes which confirms the copy of the file successful

Now after restarting Apache 2 services I checked the webs again.

[www.example32b.com](http://www.example32b.com) has got much of problems fixed.



[www.example32c.com](http://www.example32c.com) has got much of problems fixed but lesser than [www.example32b.com](http://www.example32b.com)



### Subtask 5

CSPs mitigate cross-site scripting (XSS) attacks because they can block unsafe scripts injected by attackers.