**CS336 Lab 5 – Cross site scripting (XSS) attack**

**Due: Nov 10th, 2:30pm**

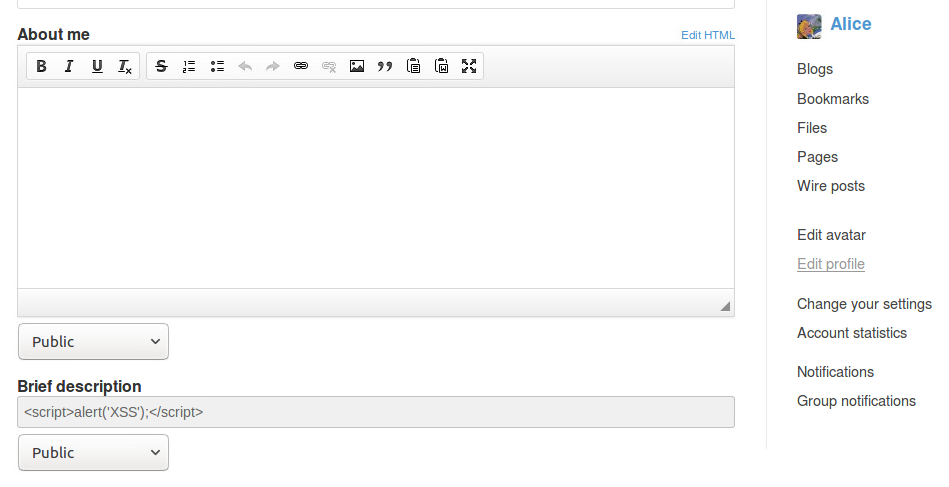
**Turn in: a lab report**

**Points: 100 pts**

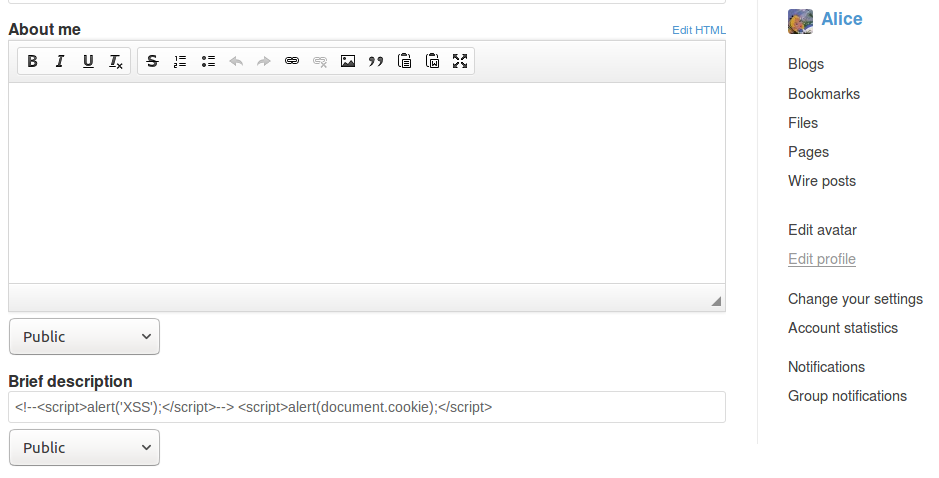
Note: see lab handout for detailed lab instructions. You don’t need to answer the questions on the handout, just need to answer the questions listed on this report. Sample code for Tasks 4 and 5 can be found on Bblearn.

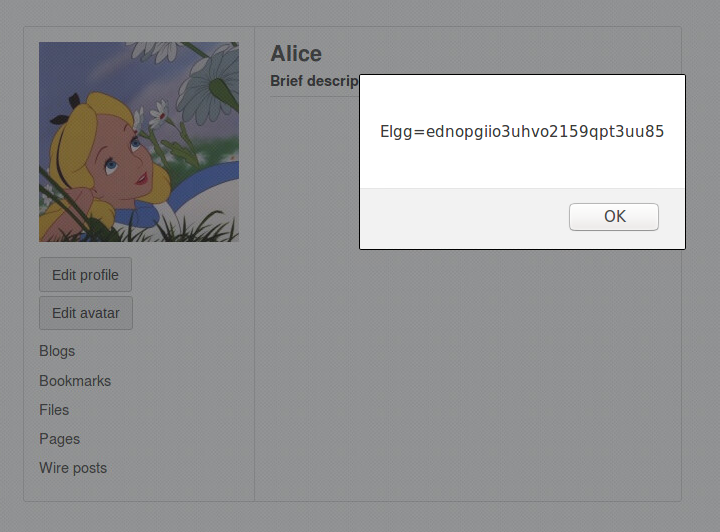
(5 pts) Task 1: Posting a malicious message to display an alert window.

* For this task, we just put an alert script into the “Brief description” field on Alices page. Anytime someone visits the page, the browser will run the alert script:

(5 pts) Task 2: Posting a malicious message to display cookies

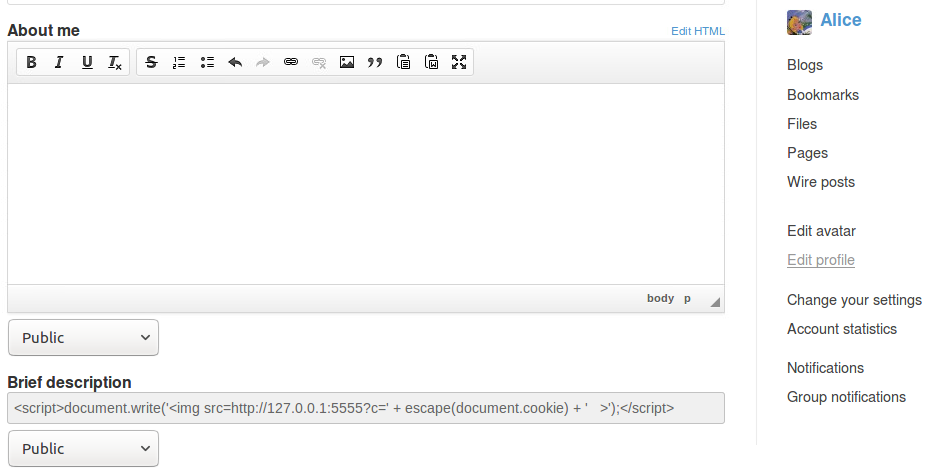
* This is similar to the last task, but this time we display the document.cookie object.



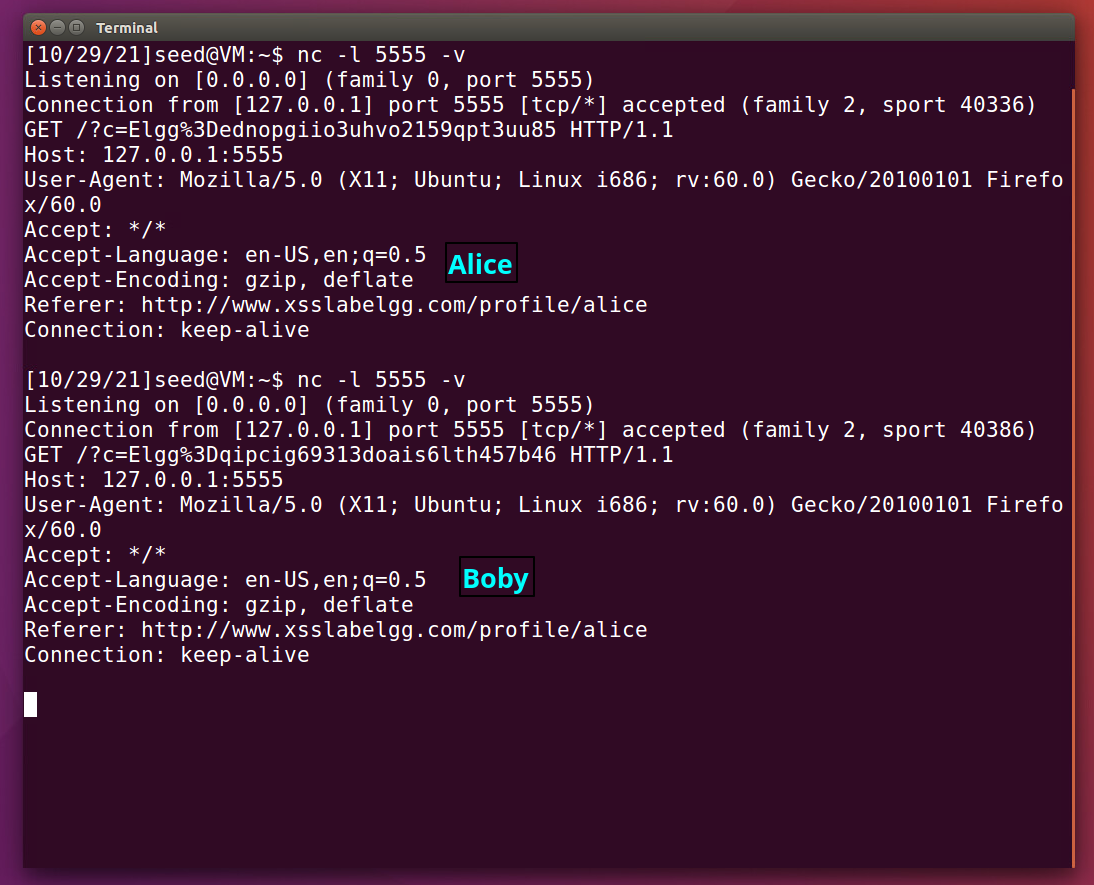


(10 pts) Task 3: Stealing cookies from the victim’s machine

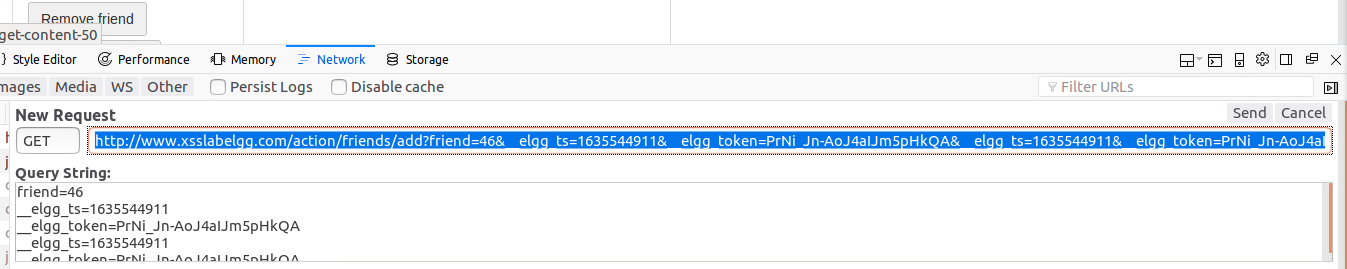
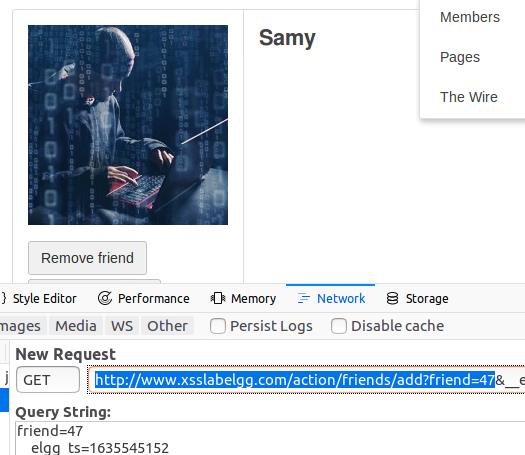
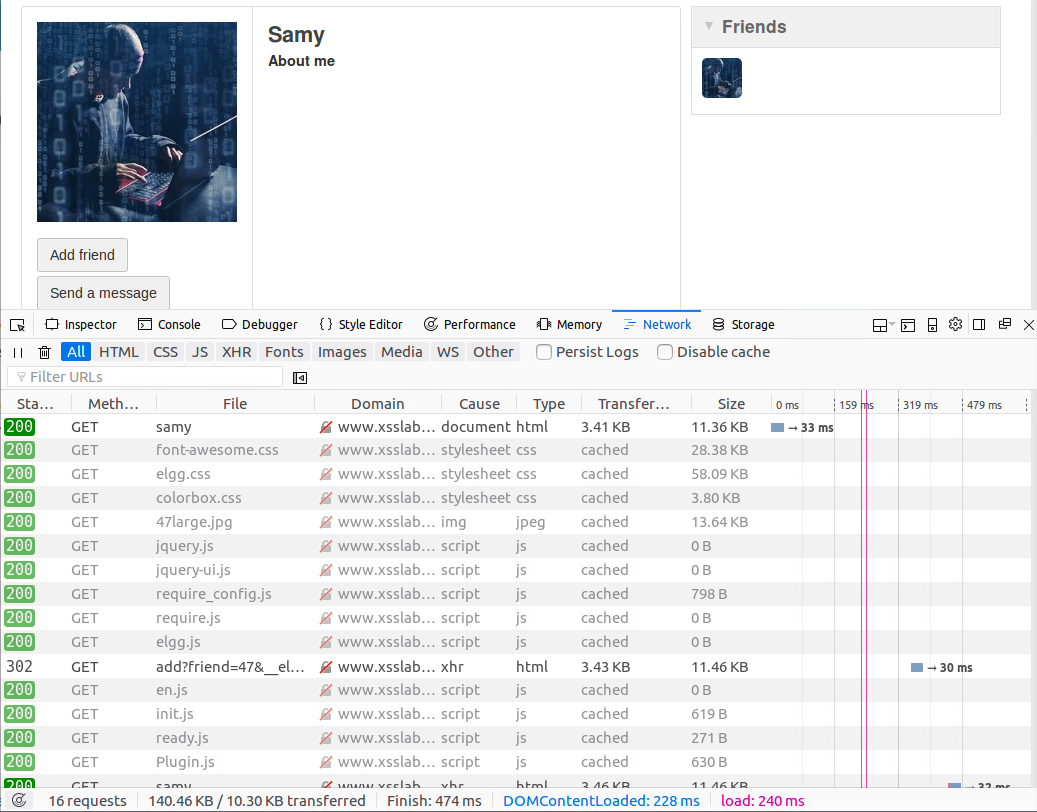
* For this task, we tell the victim to write to the attackers computer, which is listening for input:



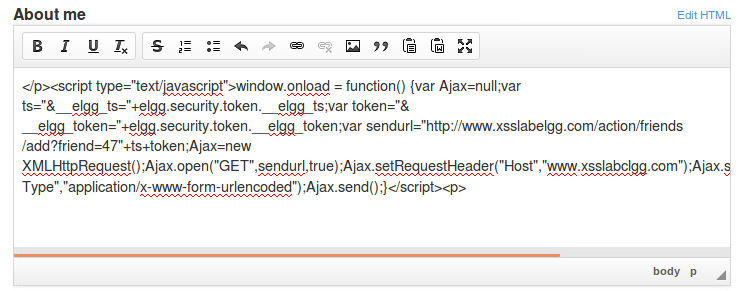
Here you can see the different cookies we receive for Alice and Boby:



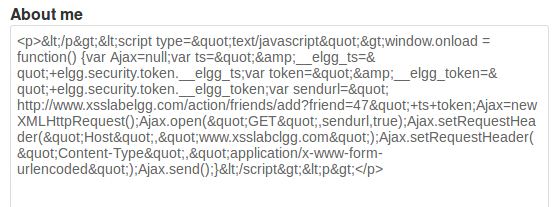
(30 pts) Task 4: Becoming the victim’s friend

* (5) *When you examining the HTTP GET request for adding a friend, take a screenshot to show the URL for adding a friend.* Here is the GET Request from the browser’s HTML inspector.
* (15) Have a screenshot to show the code you injected on Samy’s profile page.
* *How did you get the correct URL?* I added Samy as a friend, then inspected the request sent.
* *How did you figure out the friend id?* This is listed above^ in the highlighted section.
* (10) Answer this question: 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?
* I tried a couple different methods, but had no luck.

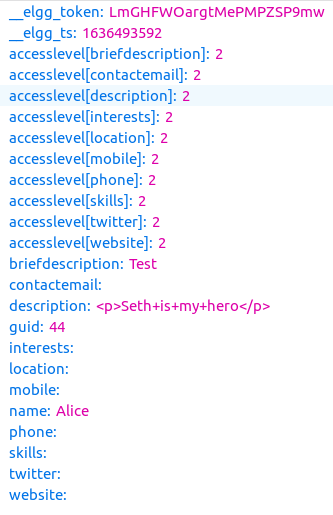
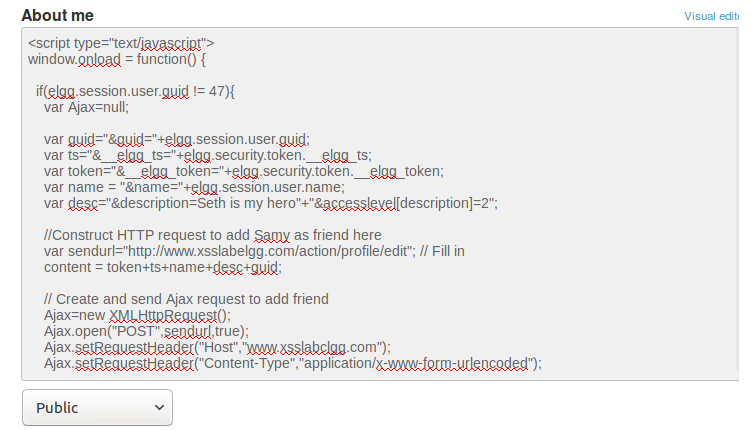
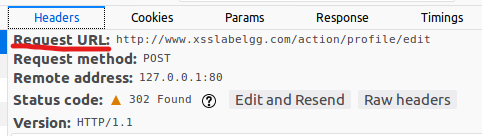
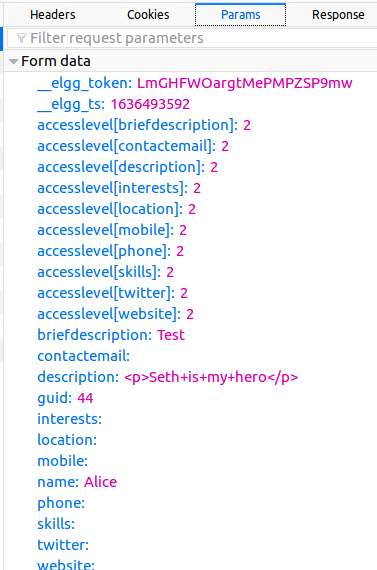
First I tried something similar to the SQL injection commenting method. I started the script with a closing </p>, and ended it with a <p> tag, hoping to insert the script between them.



This was not successful, as the Elgg application converts the symbols like the < and > to &lt; and &gt;, then wraps the code with p tags:



(40 pts) Task 5: Modifying the Victim’s Profile

* (5) **For this task, instead of using “samy is my hero”, let’s modify the message to be “[your name] is my hero” (put your name in the message ☺).**
  + **Done below!**
* (5) Take a screenshot to show a real HTTP POST request when editing a profile page (just the content, not the header of the request)
  + I’m not 100% sure what you are asking for here, here is the content that is sent when editing Alice’s page:
* (20) Have a screenshot to show the code you injected on Samy’s profile page.
* 
* (I modified the friending script, that’s why the friend comments are still there)
* How did you get the correct URL?
  + 
  + The inspector made it very easy to find
* How did you figure out the content of the request?
  + This was on the slides, as well as here:
  + 
* (10) Answer this question: See the picture below. Why do we need Line 1? Remove this line and repeat your attack. Report and explain your observation.
  + If we didn’t have this, as soon as we applied our changes and it reloaded our page, our own script would overwrite itself. So, we make sure that it only runs on other users.

Graphical user interface, text, application

Description automatically generated

(10 pts) Conclusion: Write a summary of the XSS lab.

* + What is a XSS attack? What is the main reason that causes this problem?
    - An attacker injects a script into a website that runs on a victim’s machine when they visit the infected page. Since it is running on *the victim’s* computer, it has access to their cookies, and can then impersonate their actions.
    - The main cause of this problem seems to be sites not validating form input. There should be no way a user can inject HTML into the existing site, because they can then exploit this to run malicious scripts, as we saw in this lab.
  + What have you learned in this lab?
    - I was unaware that you were able to do so much from a simple input box. I didn’t realize you could inject code in such a way that the browser would actually run it; that was interesting to me. Like most, if not all, of the other labs, this one again highlights the importance of validating and sanitizing user input.