Shane Irons

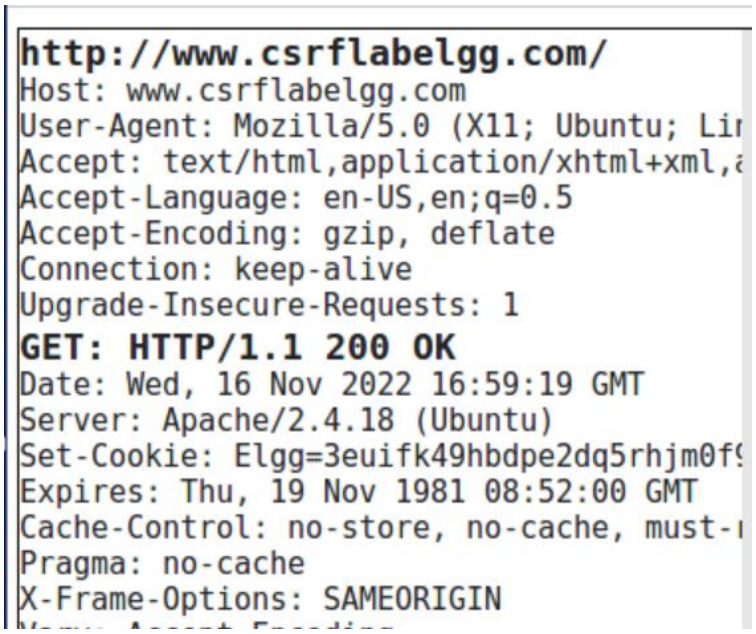
CIS 5627

Project 4: CSRF Lab

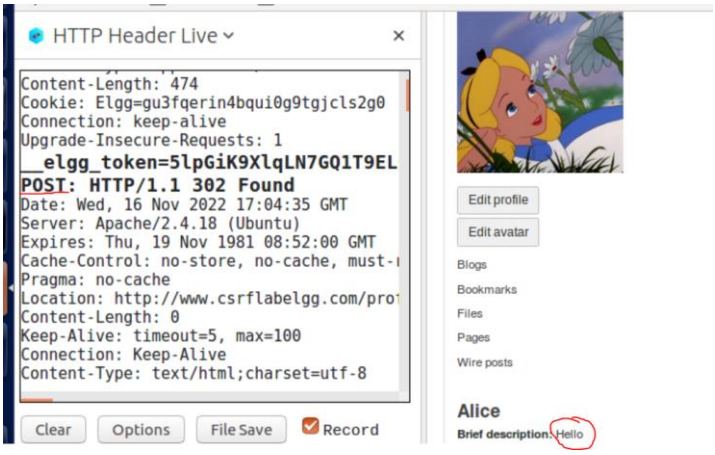
11/22/2022

Task 1:

HTTP Get:



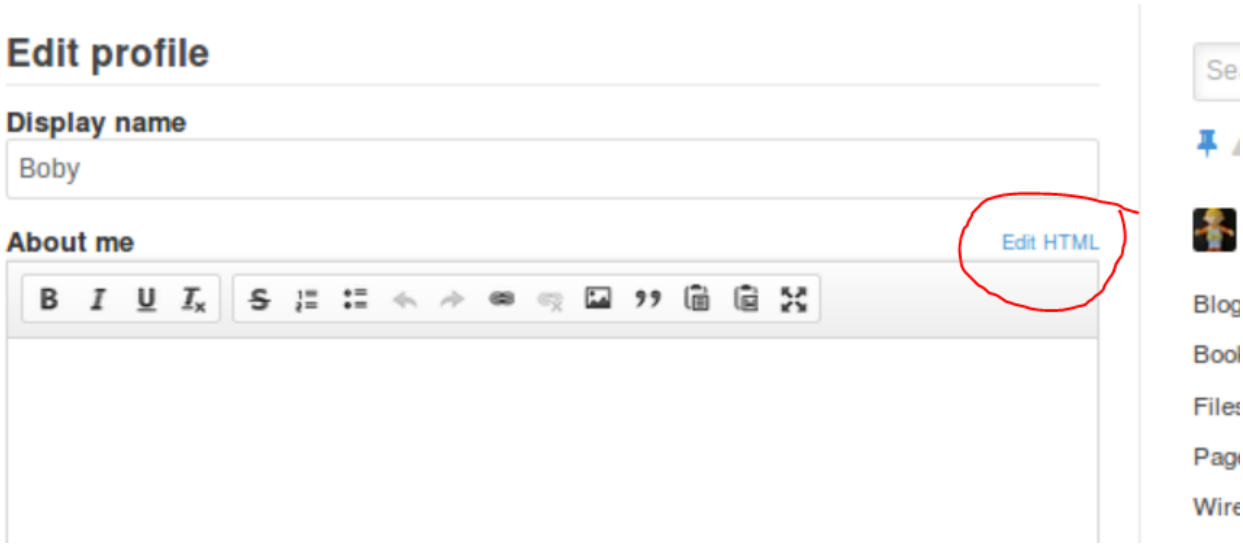
Http Post:

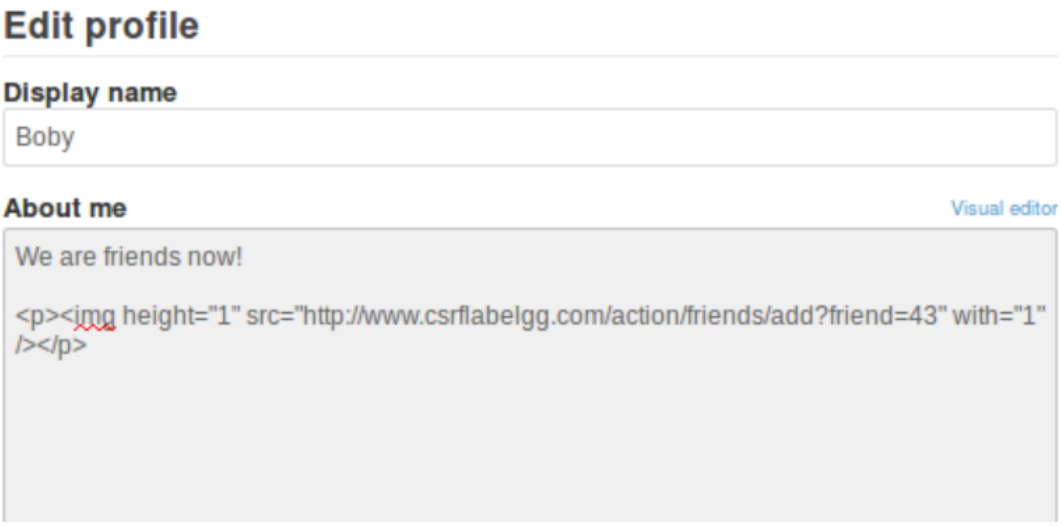


The token identified is with the post request. It is seen in the above screenshot right after "__elgg_token=…"

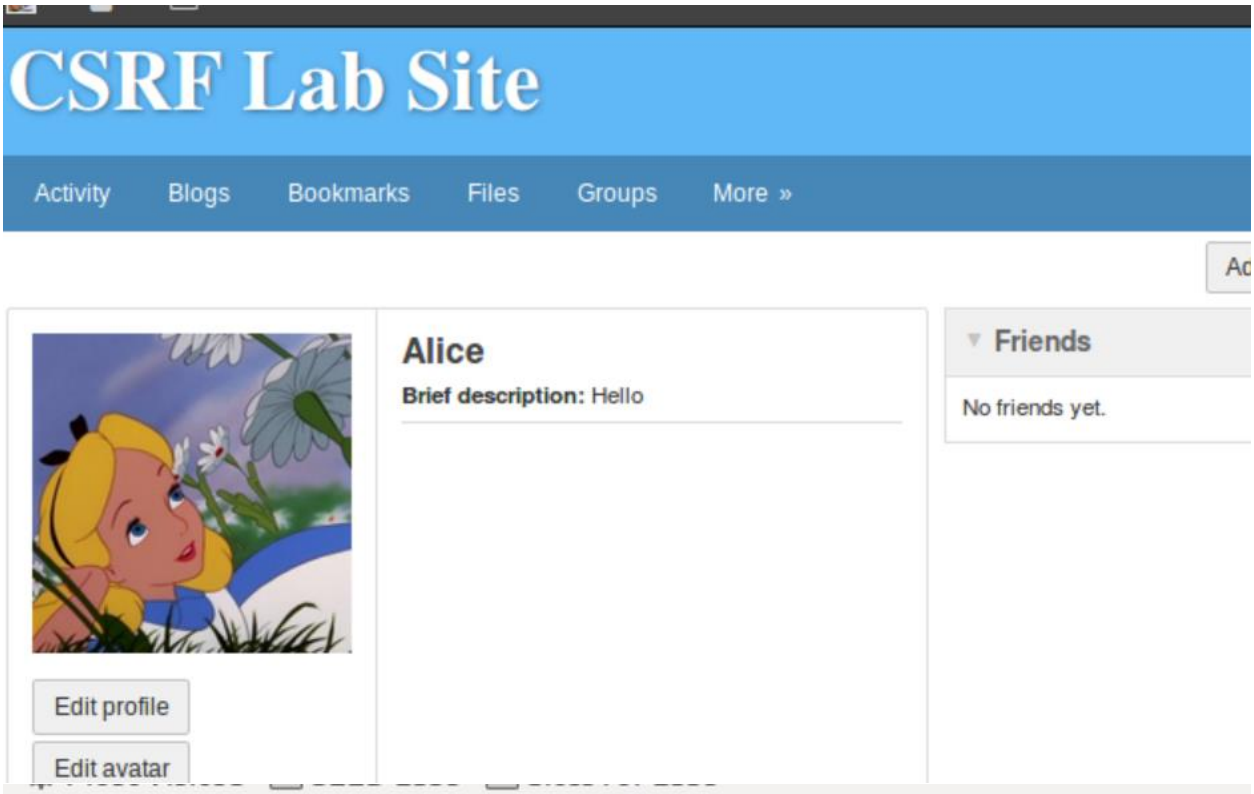
Task 2:

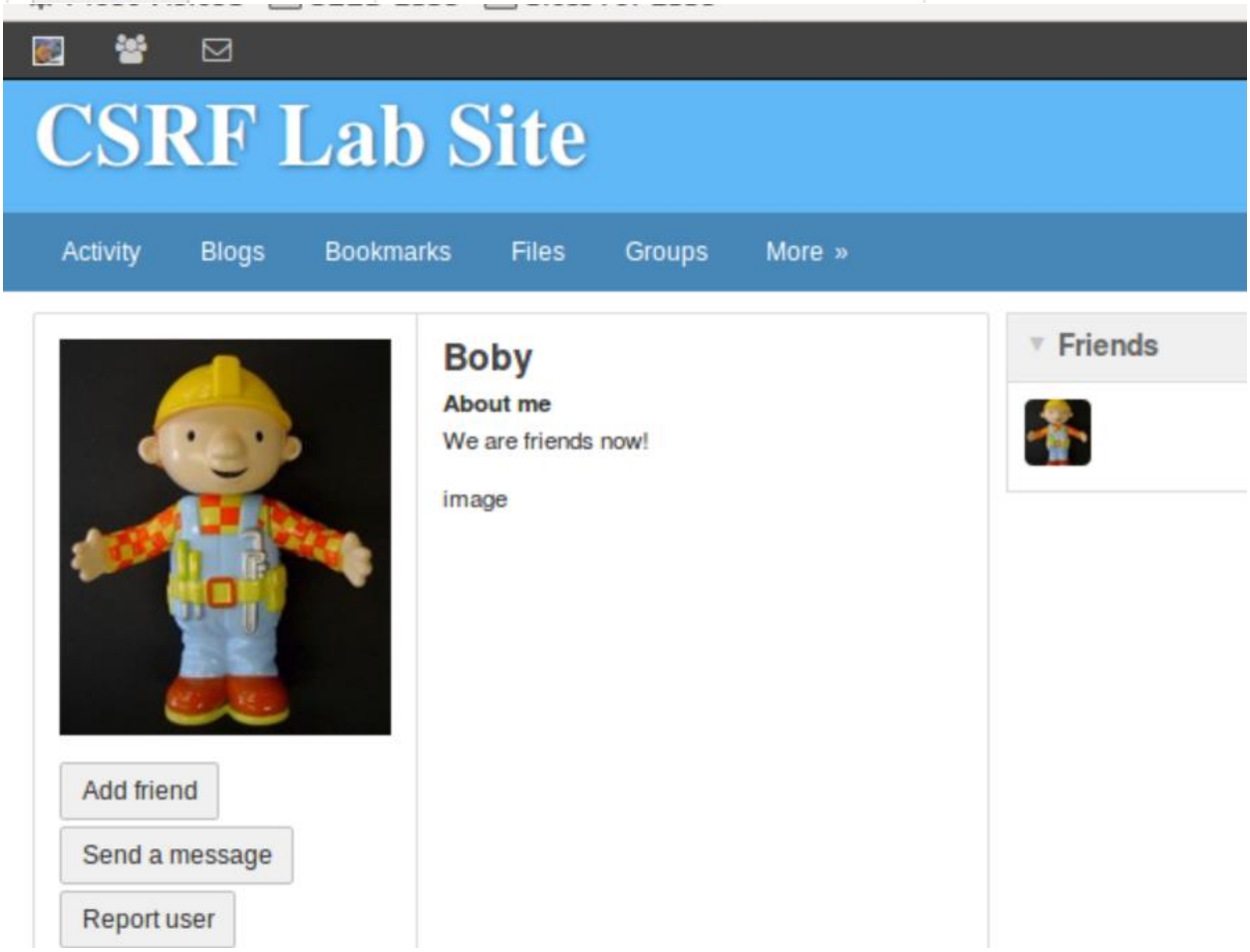
Boby user guid is 43. Knowing this, I can construct the get request attack. First, edit boby's profile:

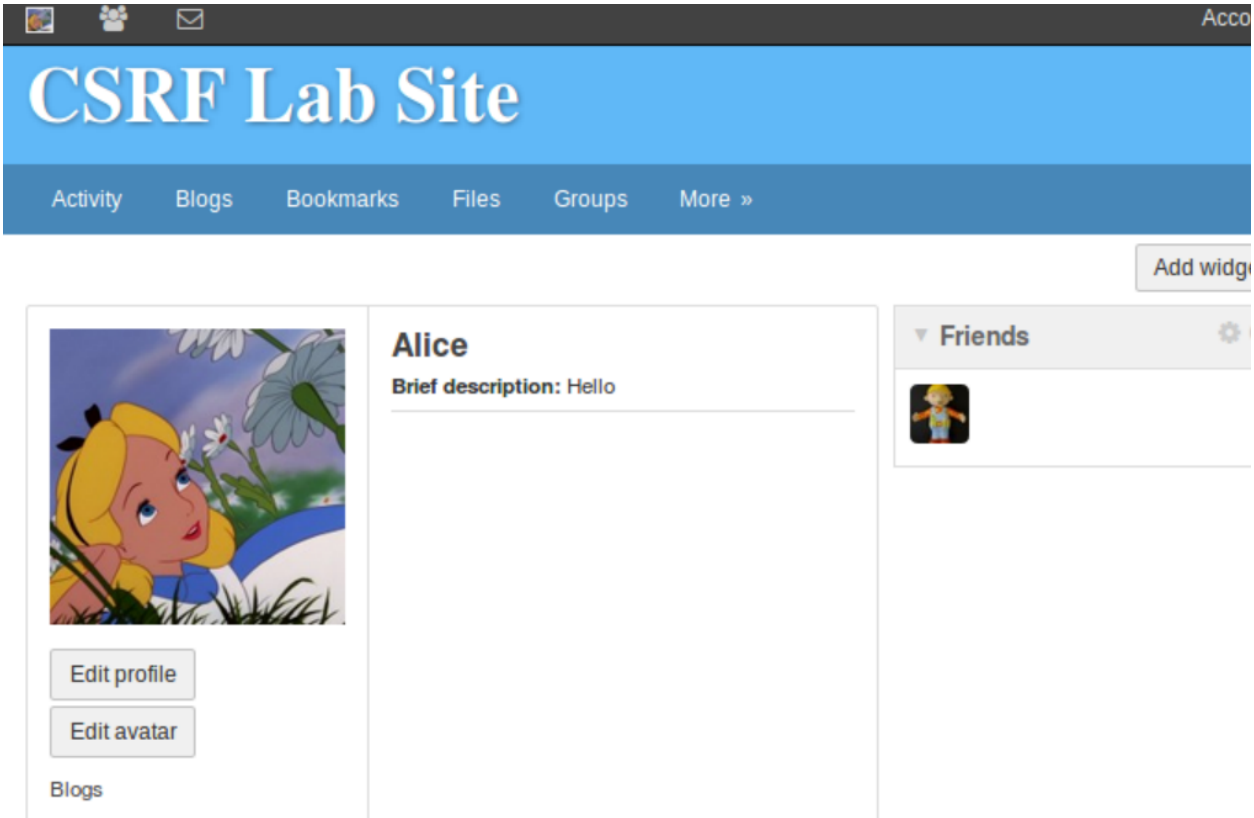




Above is my constructed attack that is embedded into Boby's profile. When Alice visits his page, he will automatically show up on her friends list. I use the img src approach for this attack.


CSRF Lab Site
Activity
Blogs
Bookmarks
Files
Groups
More »
Alice
Brief description: Hello
Friends
No friends yet.
Edit profile
Edit avatar

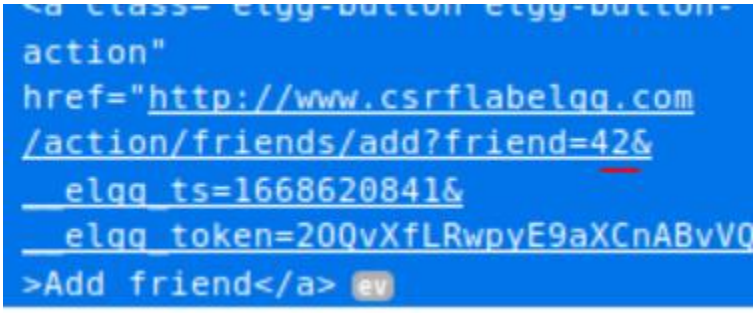

CSRF Lab Site
Activity
Blogs
Bookmarks
Files
Groups
More »
Boby
About me
We are friends now!
image
Friends
Add friend
Send a message
Report user



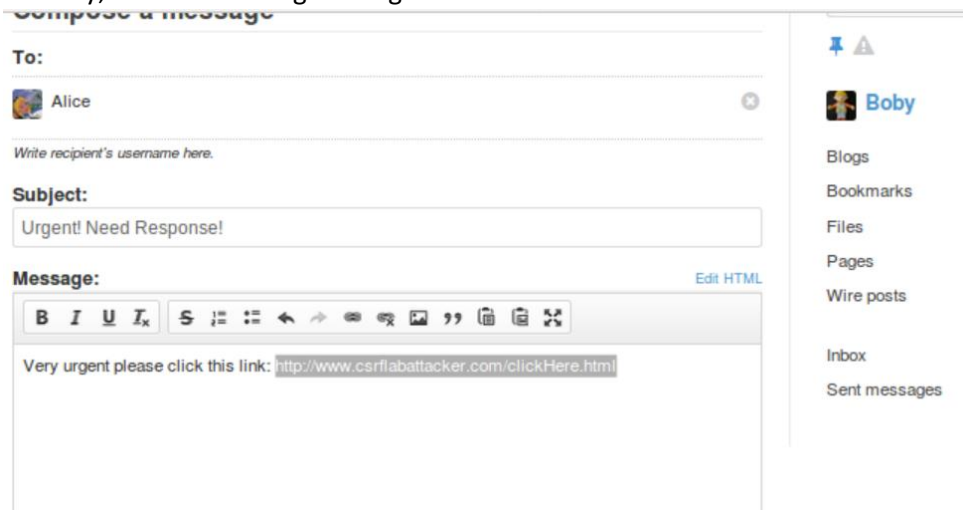
With this attack, simply visiting Boby's profile page will add him to the user's friends list without any warning. You can also see Boby is friends with himself after this attack.

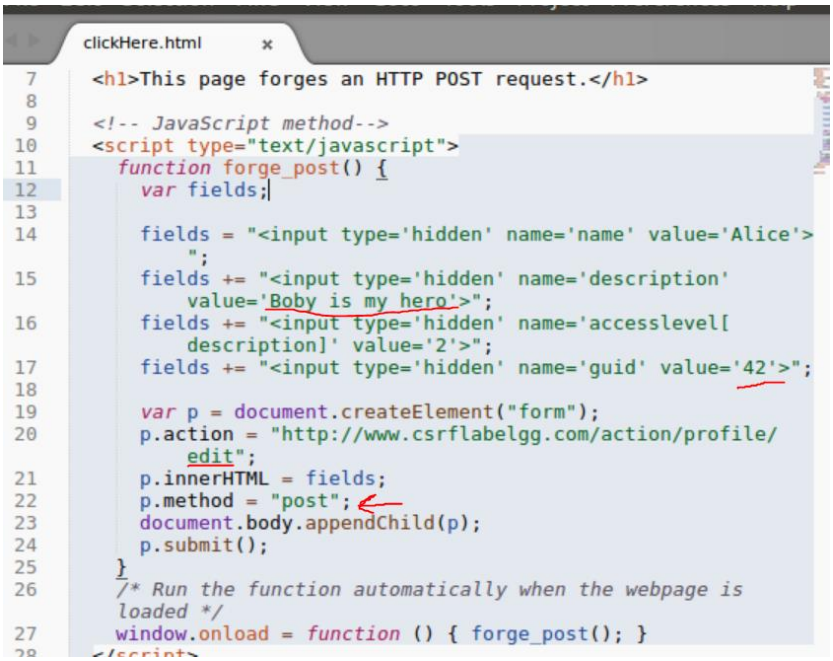
Task 3:

Making the post request… Alice GUID is 42

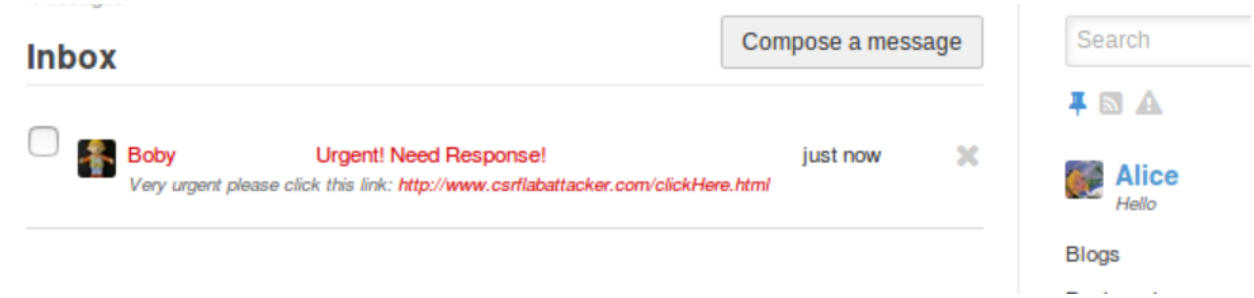


As Boby, send an enticing message to Alice:

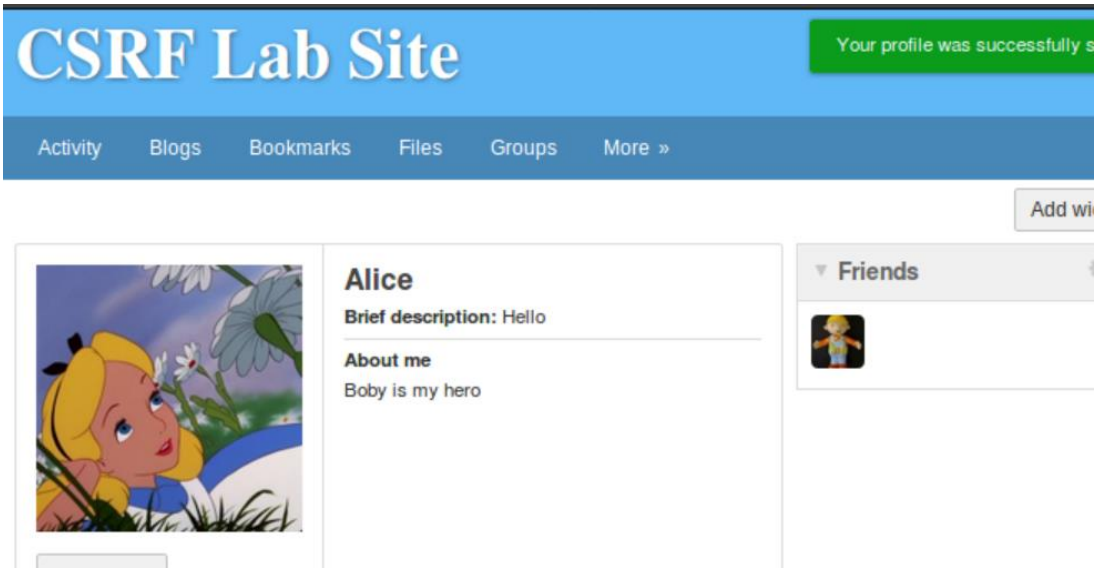




Above is the clickHere HTTP POST attack.



Message showing up for Alice.



As soon as I click the link, a web page opens up for a second then returns to Alice's home page with the about me post "Boby is my hero".

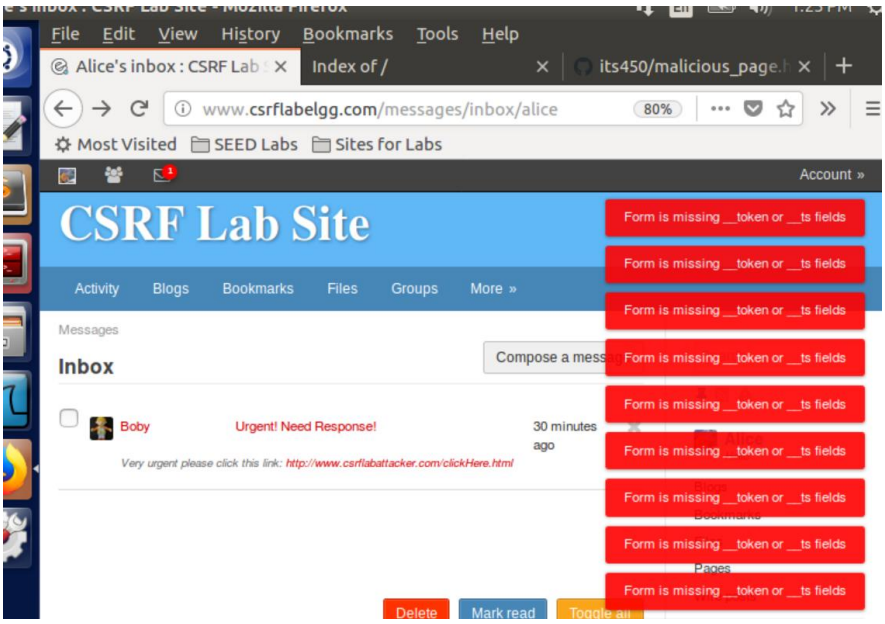
Question 1: Boby solves the issue of not being able to log in to Alice's account to get the guid by using other methods to attain it. The method I was able to use was to view the page source code by right clicking on Alice's page (from being logged in as Boby) and CTRL-F for "guid", then searching around for the correct code. Another method you can use is simply hovering over the "Add Friend" button. On this website, hovering over this button displays a url at the bottom of the page (that reflects what will happen when the button is clicked). This url displays Alices' guid.

Question 2: Yes, Boby could theoretically launch this attack on anyone who visits the page. The code would have to be modified to automatically search for the targets guid and input it into the guid value field. This could probably be done in a script automatically as soon as the link is clicked.

Task 4:

```
 */
public function gatekeeper($action) {
    //return true;

    if ($action === 'login') {
        if ($this->validateActionToken(
            false)) {
            return true;
        }
```



Upon retrying the previous POST attack, the webpage remains open (does not automatically return to Alice user page) and when going back, these errors are thrown onto the screen. This shows that the attack now does not work when the countermeasures are enabled.

The secret tokens in the HTTP request are shown here:

```
,"security":{"token":{"__elgg_ts":1668623263,"__elgg_token":"rkaPKXqHrvOMxoJ9HeRRlQ"}},"
ipt><script src="http://www.csrflabelgg.com/cache/1549469429/default/jquery-ui.js"></scr
```

These are the elgg tokens associated with Alice. These can be found using the Firefox HTTP inspection tool.