Environment setup:

Text

Description automatically generated with medium confidence

Building the docker server using dcbuild



Starting the server using dcup

Graphical user interface, text

Description automatically generated

Mapping the addresses in /etc/hosts for attacker, example and seed-server

Task1:

Graphical user interface, text, application

Description automatically generated

Using HTTP header live add-on to track all the requests and responses in the webpage

Graphical user interface, text, application, email

Description automatically generated

We use the HTTP header live tool to capture a HTTP post request being sent to elgg website

The post request has a body to it, which is used to sent along with the post request the elgg server

Graphical user interface, text, application, email

Description automatically generated

Here We use the HTTP header live tool to capture a HTTP get request being sent to elgg website

And the HTTP get request is used to get the requested data from the server

Task2

Graphical user interface, text, application

Description automatically generated

Here we use HTTP get request to get the url for our attack which can be seen in the src part of the next image

 this is the id for samy which we get from the source page of samy’s profile

A picture containing website

Description automatically generated

Restart server and we have got the friend id form the guid of samy and the url by using add friend button using the http header live which we see in the above image

Graphical user interface, application

Description automatically generated

Login as samy so that we can set up the attack

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Samy writes a post that has the attacker website and publishes it on the blog so that alice can open it when she sees it

Graphical user interface, text, application, chat or text message, email

Description automatically generated

Here is the preview of the post that Sammy created

Background pattern

Description automatically generated with low confidence

Here we login as alice and see that she has no friends right now

Graphical user interface, text, application, email

Description automatically generated

Alice reads samy’s post and opens the url which redirects alice to the melecious website (here the get request is being forget) where our attack runs which make our attacker (samy) friend to anyone who views the website being logged in

A picture containing diagram

Description automatically generated

This is the malicious website preview

Graphical user interface, application

Description automatically generated with medium confidence

When we see alices’s friend list after the attack is done, we can see that samy is now friend to alice

Task3:

Graphical user interface, text, application

Description automatically generated

In this task we use the HTTP header live to capture the post request that is generated when edit profile is saved

Graphical user interface, text, application, email

Description automatically generatedwe get the guid from the post request’s body which is 56 here

Graphical user interface, text

Description automatically generated

We are given the skeleton of the code for editing the profile as editprofile.html

We edit the .html file according to the details we have got from the previous steps

That is adding the value for name which is Alice, value for brief description which is “samy is my hero” and value for guid which is 56

We also change p.action to the url which is in the above post request

Graphical user interface, text, application, email

Description automatically generated

Then samy adds another post so that alice can open it and the attack can be executed

Graphical user interface, text, application, email

Description automatically generated

Then alice opens the malicious website by clicking on it

Graphical user interface, application

Description automatically generated

Here we can see that when alice opens the malicious website the malicious javascript that exists in the .html (the forged post request send it to the malicious website) file gets executed and prints “samy is my hero” on alice’s profile

Question1:

Boby can learn alice guid bu inspecting the add friend request on alice. The HTTP header live will give bob the url of the get request and he can fiend the id that alice used which will be alices guid

Question2:

This is not possible as bob will have to know the euid of the victim. Which means he cannot attack everyone and can attack only one person whose guid is known

Task4:

A picture containing text

Description automatically generated

Opening Csrf.php at it location

Text

Description automatically generated

Commenting out the return statement in in validation function to turn on counter measures

Graphical user interface, text, application

Description automatically generated

here we try to do the same attack as in previous task

Graphical user interface, text, application

Description automatically generated

We are directed to the malicious webpage where we can’t understand anything so we press the back button and go back to the elgg webpage

Graphical user interface, application

Description automatically generated

When we come back to the page we see there are multiple error messages on the screen as the counter measures are turned on and doesn’t allow the attack to execute but the attack keeps running in background until it gets executed and because of the counter measure is turned on the attack can never execute as the token and ts values are missing