Lybryant Wright
CUNY John Jay College
CSCI 401
Lab 8

Cross Site Request Forgery

The attack we explore this week is Cross Site Request Forgery. This attack requires a victim, a trusted website and a malicious website. Basically the user will believe they're connected to a trusted website however it's just a trusted session while they connect to the malicious website that'll inject HTTP requests to cause damage to the user. In this instance we attack the social network application, Elgg.

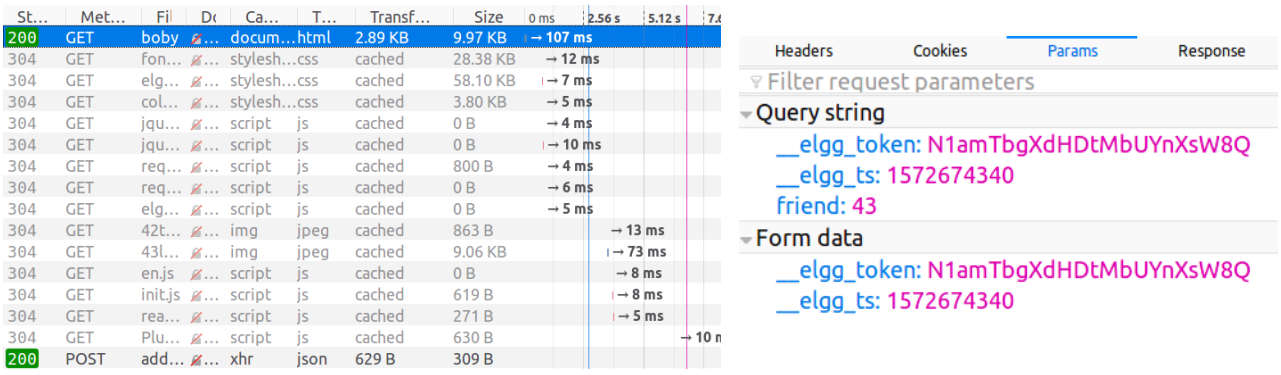
```
[10/26/19]seed@VM:~$ cd /var/www/CSRF
[10/26/19]seed@VM:.../CSRF$ ls
Attacker  Elgg
[10/26/19]seed@VM:.../CSRF$
```

Above is the location for the Elgg and attacker URLs on the VM.

```
<VirtualHost *:80>
        ServerName http://www.csrflabelgg.com
        DocumentRoot /var/www/CSRF/Elgg
</VirtualHost>
<VirtualHost *:80>
        ServerName http://www.csrflabattacker.com
        DocumentRoot /var/www/CSRF/Attacker
</VirtualHost>
```

Above is the virtualhosts enabled on the apache server

3.1)



Using Elgg I captured a HTTP GET and HTTP POST. The HTTP GET didn't have any parameters but the HTTP POST parameters are _elgg_token, and _elgg_ts.

3.2 & 3.3)

To perform this GET attack, the URL created when adding boby as a friend has to be inserted into the attacking website. This URL carries the token and timestamp parameters above to authenticate the friend request.

```
http://www.csrflabelgg.com/action/friends/add?friend=43&__elgg_ts=1572674340&__elgg_token=N1amTbgXdHDtMbUYnXsW8Q
```

Insert above url in a tag on the attacker websites html code.

I couldn't perform the POST attack properly but what should have occurred was that in the html file, the function created to forge a HTTP POST should have posted a blog post on Alices page stating Boby is he hero.

Question 1: One way Boby could have got Alice's account password would have been to sql inject the password attribute to pull a table of Alice's information.

Question 2: No, random users going to the attack website wont be affected by the attack because the GET URL has a timestamp and token that's unique per user.