

CSCI 401

Lab - 8

Prof. Kadri Brogi

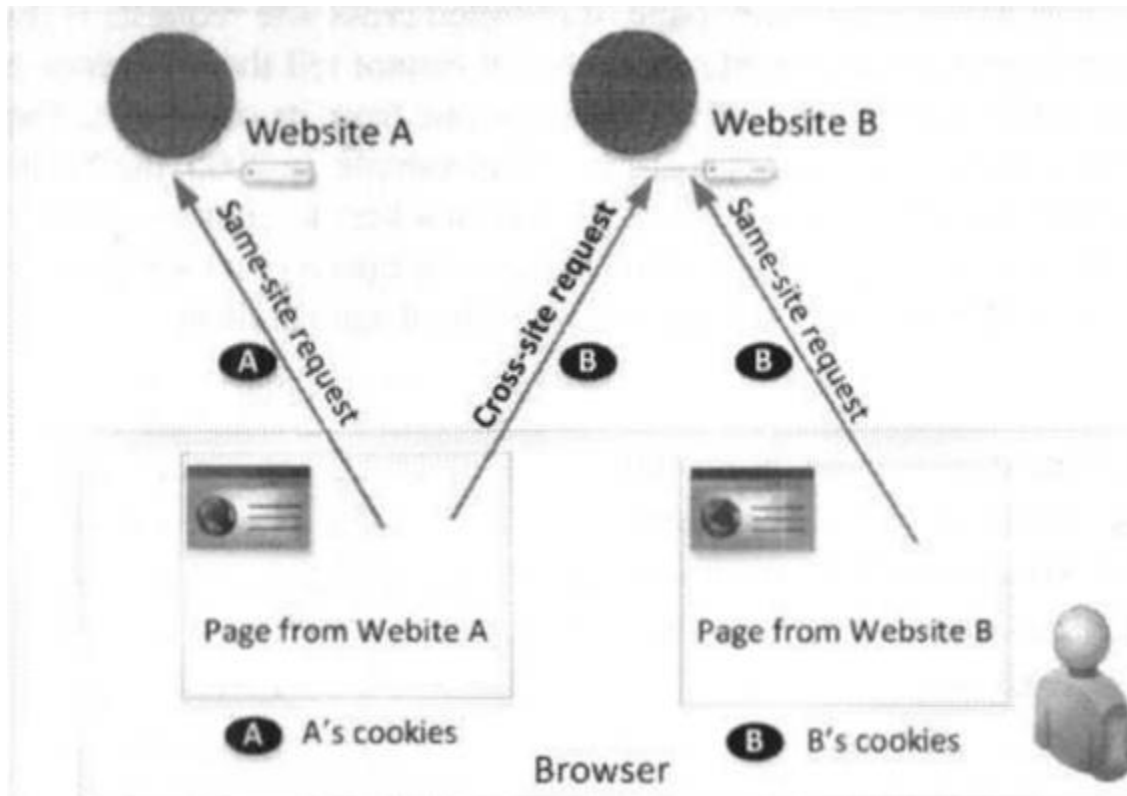
April 3, 2020

Emranul Hakim

23467834

Cross-Site-Request-Forgery Attack

Cross Site Request Forgery Lab is a very interesting and fun lab. In order to understand Cross Site Request Forgery Lab, we have to understand what Cross site request is. When a page from website sends an HTTP request back to the same website, it is called same site request. On the other hand, Cross site request is when the request is sent back to different website.



To make this attack work, we need three factions: a victim user, a targeted website and a malicious website which is controlled by an attacker. For this attack to work, the victim must be already logged in the targeted website or else the attacker can still broadcast a forged request, but the server will not process the request. The victim must detain an active session with a trusted site even though it is visiting the malicious site. The malicious site begins to inject an HTTP request for the trusted site into the victim's session, resulting in damages.

Goal:

Our goal is the lab is to perform the CSRF attack on an open source social networking web application known as Elgg which already built in Ubuntu VM image. We are provided with both the attackers website and the victims website.

Attacker Website:

URL: www.CSRFLabAttacker.com

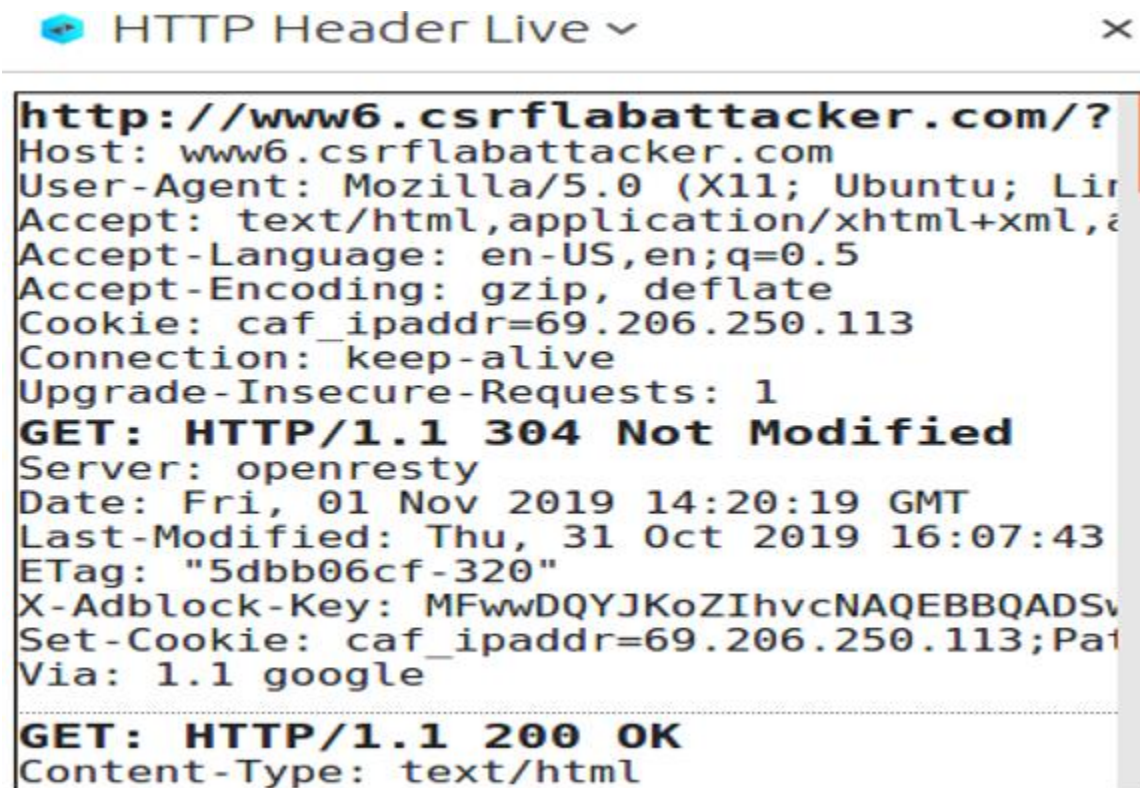
Document root: /var/www/CSRF/Attacker

Victim Website:

URL: www.CSRFLabElgg.com

Document root: /var/www/CSRF/elgg

For this lab, we will need to install an addon to firefox called "HTTP Header Live". This add-on will allow us to inspect http requests in full detail.



The screenshot shows the 'HTTP Header Live' extension interface in a browser. It displays the details of an HTTP GET request to the URL `http://www6.csrflabattacker.com/?`. The request headers include Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Cookie, Connection, and Upgrade-Insecure-Requests. The response status is `GET: HTTP/1.1 304 Not Modified`. The response headers include Server, Date, Last-Modified, ETag, X-Adblock-Key, Set-Cookie, and Via. Below this, another request is partially visible: `GET: HTTP/1.1 200 OK` with a Content-Type of `text/html`.

```

http://www6.csrflabattacker.com/?
Host: www6.csrflabattacker.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Lin
Accept: text/html,application/xhtml+xml,a
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: caf_ipaddr=69.206.250.113
Connection: keep-alive
Upgrade-Insecure-Requests: 1
GET: HTTP/1.1 304 Not Modified
Server: openresty
Date: Fri, 01 Nov 2019 14:20:19 GMT
Last-Modified: Thu, 31 Oct 2019 16:07:43
ETag: "5dbb06cf-320"
X-Adblock-Key: MFwwDQYJKoZIhvcNAQEBBQADSw
Set-Cookie: caf_ipaddr=69.206.250.113;Pat
Via: 1.1 google

GET: HTTP/1.1 200 OK
Content-Type: text/html
  
```

Scenario and Explanation:

Imagine, there is two person Alice and Samy. Samy wants to add Alice on Elgg, so he sends a friend request to Alice, but Alice refuses to add Samy. Now, Samy is mad about the refusal. Thus, he decides that he's going use a CSRF attack to add himself to Alice's friends list. In order to perform the task, he has to use the HTTP header live addon to check what a legitimate add friend http request made up of. I was not able to figure out how to make a profile for myself on the website in order to request someone as a friend. I asked for a friend help, and it turned out he does not know either. However, after discussing with him and from reading the chapter, both of us assuming that that would be the way to go about completing the task. After Samy successfully added himself to Alice's friends list. He wants to post on Alice's profile something inappropriate.

Samy planned out how he will use the CSRF attack and achieve his goal. Samy will need to take the same procedures as he added himself to Alice's friends list without Alice's consent. We will have to capture an HTTP request of an authentic HTML post request on the Elgg social network. Alice has to click on a link that Samy sends her which will take Alice to Samy's malicious website. This require an aspect of Social Engineering; the link would have to be very appealing and trustable to Alice.

Summary:

In a Cross-Site Request Forgery Attack, the victims are tricked to visit an attacker's website. The attacker can send a fake request to the targeted website while the victim is viewing the attacker's website. If the targeted website cannot identify whether not the request is coming from the same website or the third-party website, there will be a huge problem. Forged requests from attackers can cause major damage and security breach. There are many countermeasures we can take to prevent this attack. The simple solution as book explained is to include secret tokens and same site cookies, which helps website to identify whether or not a request came from its own website or a different website.