IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

A Study About Implementation of CSRF Attacks

Kamaljeet Kumar

M.Tech CST (Cyber Security), Central University of Punjab, Bathinda

Available online at: www.ijcseonline.org

Accepted: 18/Jul/2018, Published: 31/Jul/2018

Abstract- Today worldwide revolution in web application technology is changing our lives in term of the way we learn and use. Web applications fit into this because the technology has been around long enough and can provide benefits for development in this area. The main objective of this paper is to study about the CSRF attacks and implement these attacks in real world and check the success rate of these attacks. The CSRF attacks are the state changing attacks not the data stealing attacks. In this paper also discussed how various tools and frameworks that are helpful to perform the CSRF attacks works. The implementation technique of CSRF attack is discussed in fully detail. One can easily learn and understand the CSRF attacks and its implementation using this paper.

Keywords- Cross-Site Request Forgery, Log analysis, CSRF Attacks, Implementation of CSRF

I. INTRODUCTION

Today web applications are the major part of internet. Most of the daily life usage services are based on web applications and their usage is increasing day by day. For example, banking services, social networking, e-commerce service, and other services. All of these services can be accessed from anywhere and anytime. There are Hackers across the internet and they try to manipulate the functionality of web service for their benefits. In web applications lots of vulnerabilities are there which allow the hackers to break into the websites.

Vulnerability is failure or flaws in a system's design implementation, working or management that exploits the system's security objectives. The *Open Web Application Security Project* (*OWASP)* non-profitable organization categorized the web application vulnerabilities on the basis of their risk level in the name of OWASP TOP 10. One of them vulnerability is Cross Site Request Forgery (CSRF). CSRF attacks are used to change the target state request, not theft of data. If the victim is normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address and so on. If the victim is admin user of website then, successful CSRF attack can compromise the whole website. For forensics of CSRF attacks, analysing the log file is usually preferred, because users' requests and related server responses could be clearly identified. Benefits of using log files for forensics for CSRF attacks are: first reason is log file is easily available and the second reason is there is no need for expensive hardware for analysis. In addition, logs may provide successful detection specially for encrypted protocols such as Secure Socket Layer.

1.1 What is Cross Site Request Forgery?

Cross Site Request Forgery is target state changing requests. CSRF only allows for state changes to occur and therefore cannot attack those that require the attacker receiving the contents of the HTTP response. Cross Site Request Forgery (CSRF) is an attack whereby a malicious entity tricks a victim into performing actions on behalf of the attacker. The impact of the attack would depend on the level of permissions that the victim which is being exploited have.

1.2 Understanding Work of CSRF Attack

Cross-site Request Forgery (CSRF) will only be effective if a victim is authenticated. This means that the victim will need to be logged-in, in order for the attack to succeed. Since CSRF attacks are used to bypass the authentication process, there may be some elements that are not affected by CSRF attacks, even though they are not protected against it. These do not require a logged in victim to send the request, since anyone can do this. For example, let's take a contact form on a website, where visitors can send queries through the form. This does not require the victim having any privileges to submit the form, meaning that it does not matter if an administrator victim or a low privilege victim submits the form. The problem arises when a victim with additional privileges would be performing actions that are not accessible to everyone, which is when CSRF attacks are utilized.

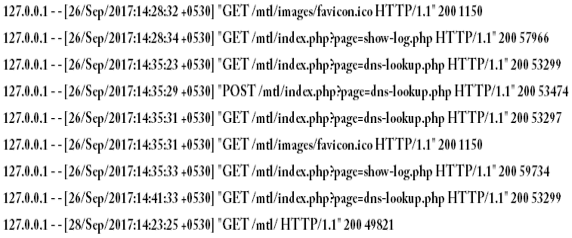
1.3 Web Server Log File

Standard web servers like Apache and IIS generate logging message by default in the Common Log Format (CLF) specification. The CLF log file contains a separate line for each HTTP request. Log file is a file which is created by the server automatically, which consists of the list of activities performed by end users of web server. A web server log file contains the information about client IP address, http/https request, date and time, page request, http status code, user agent and referrer. The whole data stored in a single log file or the data stored in separated log files such as access log file, error log file and audit log file. The log files are not available for all users but accessible to only web administrator. In the log file every entry entered following the format of Custom log Format. Below in the table Log Format strings and their description is given

Table 1 Custom Log Format of Log File

Format String	Description
%%	The percent sign.
%a	Client IP address of the request
%{c}a	Underlying peer IP address of the connection
%A	Local IP-address.
%B	Size of response in bytes, excluding HTTP headers.
%b	Size of response in bytes, excluding HTTP headers.
%{*VARNAME*}C	The contents of cookie *VARNAME* in the request sent to the server.
%D	The time taken to serve the request, in microseconds.
%{*VARNAME*}e	The contents of the environment variable *VARNAME*.
%f	Filename.
%h	Remote hostname.
%H	The request protocol.
%k	Number of keepalive requests handled on this connection.
%l	Remote logname
%L	The request log ID from the error log
%m	The request method.
%{*VARNAME*}n	The contents of note *VARNAME* from another module.
%{*VARNAME*}o	The contents of *VARNAME*: header line(s) in the reply.
%p	The canonical port of the server serving the request.
%{*format*}p	The canonical port of the server serving the request or client actual port.
%P	The process ID of the child that serviced the request.
%{*format*}P	The process ID or thread ID of the child that serviced the request
%q	The query string.
%r	First line of request.
%R	The handler generating the response (if any).
%s	Status. Status of original request
%t	Time[18/Sep/2011:19:18:28 -0400].

In figure 2 the web server log file is shown. In this log file we can see the IP address of user, date and time, request method, path of requested page, protocol version, request status code and bytes received are shown.



```
127.0.0.1 - - [26/Sep/2017:14:28:32 +0530] "GET /mtl/images/favicon.ico HTTP/1.1" 200 1150
127.0.0.1 - - [26/Sep/2017:14:28:34 +0530] "GET /mtl/index.php?page=show-log.php HTTP/1.1" 200 57966
127.0.0.1 - - [26/Sep/2017:14:35:23 +0530] "GET /mtl/index.php?page=dns-lookup.php HTTP/1.1" 200 53299
127.0.0.1 - - [26/Sep/2017:14:35:29 +0530] "POST /mtl/index.php?page=dns-lookup.php HTTP/1.1" 200 53474
127.0.0.1 - - [26/Sep/2017:14:35:31 +0530] "GET /mtl/index.php?page=dns-lookup.php HTTP/1.1" 200 53297
127.0.0.1 - - [26/Sep/2017:14:35:31 +0530] "GET /mtl/images/favicon.ico HTTP/1.1" 200 1150
127.0.0.1 - - [26/Sep/2017:14:35:33 +0530] "GET /mtl/index.php?page=show-log.php HTTP/1.1" 200 59734
127.0.0.1 - - [26/Sep/2017:14:41:33 +0530] "GET /mtl/index.php?page=dns-lookup.php HTTP/1.1" 200 53299
127.0.0.1 - - [28/Sep/2017:14:23:25 +0530] "GET /mtl/ HTTP/1.1" 200 49821
```

Figure 1 Sample of Log File

Section I gives the brief introduction of the Cross Site Request Forgery and web server log files which also includes the format of a log file. Section II discusses about the work related of the CSRF and some other attacks. Section III discusses the various tools required for the forensics of CSRF attack to make it successful for web server log files. Section IV contains the implementation of the CSRF attack using ngrok. Section V concludes the research work and also gives the future directions.

II. Related Work

(Merve Bas, Seyyar Ferhat, Özgür Çatak Ensar Gül, 2017) use the access log files to detect the two-web attack on web application. They detect the Cross-Site Scripting (XSS) and Structured Queried Language Injection (SQLI) attacks from access log file of Apache HTTP server. They collect the data from log files and categorized them into three types on the basis of IP, User Agents and HTTP status codes. They also tried to understand the behavior of automated vulnerability scanners. They create a python script which runs on log files to detect the attack scans. Their proposed model working 99% for detecting the scans of XSS and SQLi vulnerabilities on web application.

(M. Auxilia, 2010) Suggest a negative security model for intrusion detections in web applications. This method is one of the dynamic detection techniques that is anomaly-based. The authors propose to use Web Application Firewall (WAF) with a rule set protecting web applications from unknown vulnerabilities. When analyzed their rules for Hypertext Transfer Protocol (HTTP) attacks detection, the rules appear to be generated by checking the values of some important HTTP header fields, Uniform Resource Identifier (URI) strings, cookies, etc. Associating WAF, Intrusion Detection System (IDS), rule engine reasoning together makes this article interesting

(K. Goseva-Popstojanova, 2012) Propose a method to classify malicious web sessions through web server logs. Firstly, the authors constitute four different data sets from honeypots; on which several web applications were installed.

Afterwards, 43 different features were extracted from web sessions to characterize each session and three machine learning methods that are Support Vector Machine (SVM), J48 and Partial Decision Trees (PART) were used to make the classifications. The authors assert that when all 43 features used in learning period, their method to distinguish between attack and vulnerability scan sessions attains high accuracy rates with low probability of false alarms. This comprehensive research provides significant contribution in the area of web security.

III. Tools

The programming language and various related software's and frameworks are used to make the success of Forensics of CSRF attacks from web server log files.

3.1 Python

Python is a widely used, dynamic, general-purpose, high-level interpreted language. Its syntax allows to implement stuff in fewer lines of code. There is no need for type declarations of variables, functions, methods or parameters in source code. This makes the code clean, short and flexible, and you lose the compile-time type checking of the source code. Python supports multiple programming paradigms, including procedural, object-oriented or imperative and functional programming styles. Python tracks the types of all values at runtime and flags code that does not make sense as it runs. Python has a very large user base all over the world and is one of the most popular programming languages among developers.

3.2 Burp Suite

Burp or Burp Suite is a graphical tool for testing Web application security. The tool is written in Java and developed by PortSwigger Security. The tool has two versions: a free version that can be downloaded free of charge (Free Edition) and a full version that can be purchased after a trial period (Professional Edition). It was developed to provide a comprehensive solution for web application security checks

3.3 WAMP

Wamp Server refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language.

3.4 DVWA – Damn Vulnerable Web Application

Damn Vulnerable Web App *(*DVWA*)* is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web a

3.5 Kali Linux

Kali Linux is a Debian-derived Linux distribution designed for digital forensics and penetration testing. It is maintained and funded by Offensive Security Ltd. Mati Aharoni, Devon Kearns and Raphaël Hertzog are the core developers

3.6 NGROK

Ngrok allows us to run a local sever on internet with just simpe port forwarding technique.

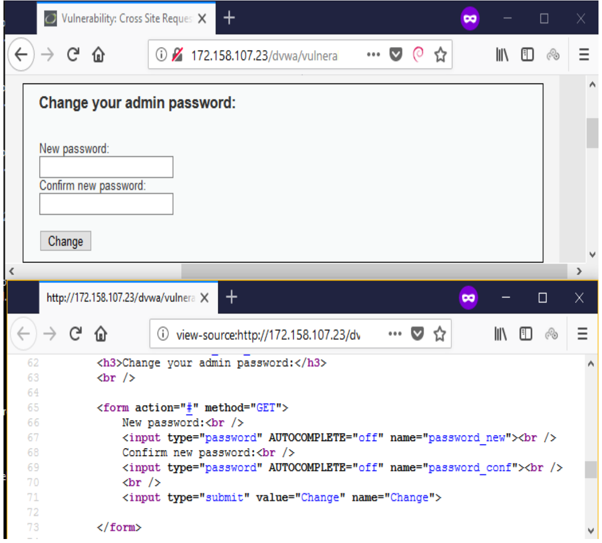
IV. IMPLEMENTATION OF CSRF ATTACK

In this section we will discuss how a successful CSRF attack performed. In this attack we will try to change password of authenticated user using CSRF attack.

4.1 Requirements

- ✓ For successful CSRF attack a user must be logged in.
- ✓ Web hosting and domain or localhost with port forwarding using ngrok.

To perform CSRF attack first of all we open a web page and get the form code with parameters.



Vulnerability: Cross Site Reques
172.158.107.23/dvwa/vulnera
http://172.158.107.23/dvwa/vulnera
view-source:http://172.158.107.23/dv

Change your admin password:

New password:

Confirm new password:

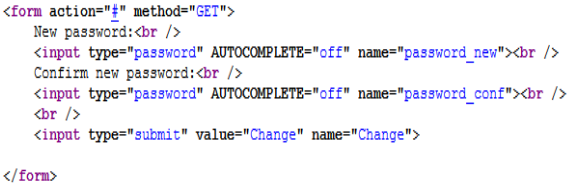
Change

```
<h3>Change your admin password:</h3>
<br />

<form action="#" method="GET">
    New password:<br />
    <input type="password" AUTOCOMPLETE="off" name="password_new"><br />
    Confirm new password:<br />
    <input type="password" AUTOCOMPLETE="off" name="password_conf"><br />
    <br />
    <input type="submit" value="Change" name="Change">

</form>
```

Figure 2 Login form and Source code of form

Now copy the form code and create a html form with this and do some change which are given below:

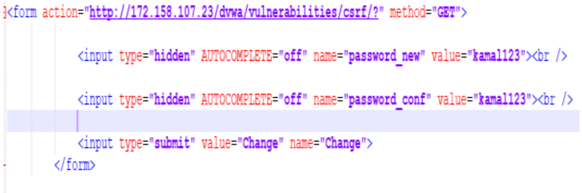
Original code



```
<form action="#" method="GET">
    New password:<br />
    <input type="password" AUTOCOMPLETE="off" name="password_new"><br />
    Confirm new password:<br />
    <input type="password" AUTOCOMPLETE="off" name="password_conf"><br />
    <br />
    <input type="submit" value="Change" name="Change">

</form>
```

Figure 3 Original Code of form

Modified Code for CSRF Attack



```
<form action="http://172.158.107.23/dvwa/vulnerabilities/csrf/?" method="GET">

        <input type="hidden" AUTOCOMPLETE="off" name="password_new" value="kamal123"><br />

        <input type="hidden" AUTOCOMPLETE="off" name="password_conf" value="kamal123"><br />

        <input type="submit" value="Change" name="Change">
    </form>
```

Figure 4 Modified Code for CSRF attack

Host this file using wamp server and put the wamp server online using ngrok.

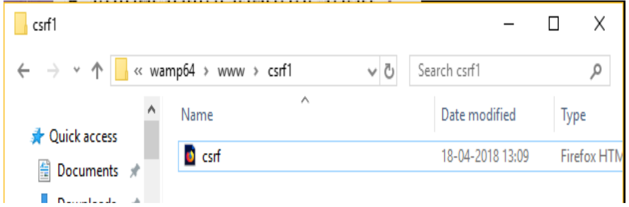
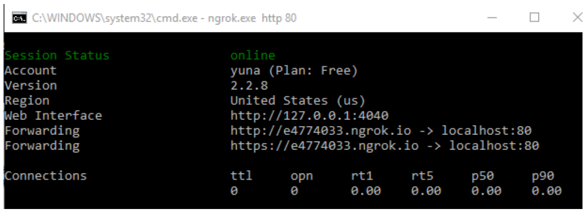


Figure 5 CSRF Attack file host on Local Server



```
Session Status                online
Account                       yuna (Plan: Free)
Version                       2.2.8
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://e4774033.ngrok.io -> localhost:80
Forwarding                    https://e4774033.ngrok.io -> localhost:80

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Figure 6 Local Host Online Using ngrok

Now send this link to victim and when victim click on it password will successfully change.

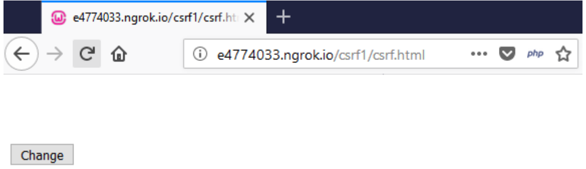
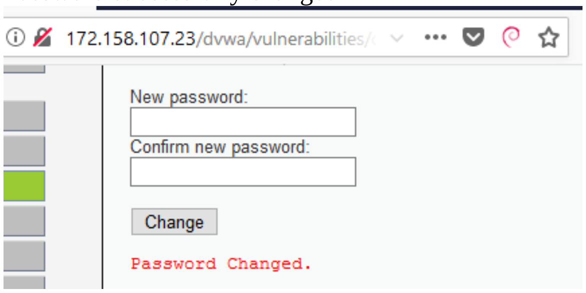

e4774033.ngrok.io/csrf1/csrf.ht
e4774033.ngrok.io/csrf1/csrf.html
Change

Figure 7 Forged Link

Password successfully changed



New password:

Confirm new password:

Change

Password Changed.

Figure 8 Successful CSRF attack

V. CONCLUSION

In this paper, the CSRF attack are studied and was performed successfully and also discussed that how these attacks can be performed. Implementing these attacks, an attacker just changes the state of requests. As it is seen that implementation of CSRF attacks is very easy and a normal user can become victim of this attack very easily. In the future work, we will work on detection of CSRF attacks using log files.

REFERENCES

[1]. CSRF Attacks, XSRF or Sea-Surf. (n.d.). Retrieved from https://www.acunetix.com/websitesecurity/csrf-attacks/

[2]. Getting Started With Burp Suite. (n.d.). Retrieved from https://portswigger.net/burp/help/suite_gettingstarted

[3]. K. Goseva-Popstojanova, G. A. (2012). *Classification of malicious web sessions*. Retrieved from 21st International Conference on Computer Communications and Networks (ICCCN): http://dx.doi.org/10.1109/ICCCN.2012.6289291

[4]. Kali Linux Tutorials. (n.d.). Retrieved from https://www.kali.org/category/tutorials/

[5]. M. Auxilia, D. T. (2010). "*Anomaly detection using negative security model in web application*". Retrieved from International Conference on Computer Information Systems and Industrial Management Applications (CISIM): http://dx.doi.org/10.1109/CISIM.2010.5643461

[6]. M. Zolotukhin, T. Hämäläinen, T. Kokkonen, J. Siltanen. (2014). "*Analysis of http requests for anomaly detection of web attacks*". Retrieved from IEEE 12th International Conference on Dependable, Autonomic and Secure Computing: http://dx.doi.org/10.1109/DASC.2014.79

[7]. Merve Bas, Seyyar Ferhat, Özgür Çatak Ensar Gül. (2017). "*Detection of attack-targeted scans from the Apache HTTP Server access logs*". Retrieved from Applied Computing and Informatics: https://www.sciencedirect.com/science/article/pii/S2210832717300169

[8]. N. Singh, A. Jain, R.S. Raw, R. Raman. (2014). "*Detection of Web-Based Attacks by Analyzing Web Server Log Files*". Retrieved from Springer India: http://dx.doi. org/10.1007/978-81-322-1665-0_10

[9]. OWASP Zed Attack Proxy Project. (n.d.). Retrieved from https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

Authors Profile

Mr. Kamaljeet Kumar pursued Master of Technology in Computere Science and Technology with specialization in Cyber Security from Central University of Punjab, Bathinda in year 2018. He is an independent Security Researcher and Web Application Penetration Tester. He is former Network Administrator and Ethical Hacking Trainer at Infowiz – A Software Solution. He has delivered 50+ workshops in various educational insititue and other organization on topics Cyber Security, Ethical Hacking and Network Security. He has Presented his Poster named as TOR: Friend or Foe in National Conference on Computer Engineering Problems Optimization – 2016 held at Central University of Punjab. He is also zonal level winner of National Network Security Championship – India organized by IIT Delhi and Network Bulls. His main research work focus on Cyber Security, Digital Forensics, Web Attacks, Network Security, Cloud Security and IoT and Artificial Intelligence.