🌐 United International Journal of Engineering and Sciences 🌐
(UIJES – A Peer-Reviewed Journal); ISSN:2582-5887 | Impact Factor:8.075(SJIF)
📘 Volume 5 | Special Issue 1 | 2025 Edition
National Level Conference on "Advanced Trends in Engineering
Science & Technology" – Organized by RKCE

Violative Web Application Security Framework

MVNSL. SUKANYA[1] SK. SHAHANAZ[2]

[1,2]*Assistant Professor, Department of Computer Science and Engineering, R K College of Engineering*

Vijayawada, India

[1]Sukanya200194@gmail.com; [2]shahanaz6shaik@gmail.com

Abstract: **Offensive Web Application Security Framework (OWSAF) is an absolute offensive security framework that helps detect vulnerabilities in web apps and aids in real-time security evaluation, which can be used to detect security misconfigurations. The result helps us understand attackers' attack vectors via a web application. The main aim of OWSAF is to give a complete guide on existing threats in the application using the model.**

Keywords: **Cyber Security, Nets parker, web application**

I.INTRODUCTION

As there is an increment in the usage of web applications, static and changing. Since there are Nessus, Nets parker, and Acunetix, to find out the vulnerabilities and suggest mitigations but there are still a lot of web-based attacks that are taking place. Attackers always use complicated tools and attack frameworks so that they can attack with ease. Both Nessus and Nets parker allow auditors to configure the framework and plan scheduled scanning based on the target scope with flexible scanning methods, but in a complicated way. Other frameworks will help to scan the targets with multiple features, with or without authentication. Regrettably, all the tools fail to give the latest discovered vulnerabilities. Despite the fact that the majority of security features include logging capabilities, they are not paired with real-time monitoring and alerting capabilities unless they are in a sophisticated security transaction center environment. As a result, to fill the void created by existing access control systems, we want to propose a new framework, namely the offensive web application security framework, that can act as a complete, comprehensive security suite that is capable of performing web and network penetration testing, misconfiguration evaluations.

II.Literature Survey

In [1], the authors described the overview of Penetration Testing which contains a flow of operations to find and exploit the security vulnerabilities in the applications. Penetration testing helps us to know the strength of the security measures that are implemented in the application. The author specified the advantages and disadvantages, methodology, and strategies of performing penetration testing. The author mentioned various strategies that can be followed to find the security vulnerabilities in the applications.In [2], Information is more important than anything for an organization to take care of. For any type of technology, security is the one that is very important to consider in order to protect the data/information. The author has described the different tools, techniques, and processes to follow in the penetration testing process. This paper helped us to know about the importance, factors, and components that need to be considered while performing penetration testing. The series of activities that must be followed, starting from identifying to reporting them to the organization, has been described by the author i.e., external testing, internal testing, router penetration, firewall penetration, application penetration, and social engineering.

In [3], Systems are being implemented complexly day by day, their infrastructure is becoming very complex which can also be prone to security threats by attackers. Not only the attackers, but it can also be someone who can take advantage of the security vulnerabilities that are present in the application where there is a more chance of their presence due to the system complexity. In this paper, the author explained how one can use penetration testing as a cyber defense to be safe from any possible attack that can happen. The authors discussed the VAPT techniques like static analysis, manual testing, automation, fuzz testing, and techniques like grey box, black box, and white box testing.In [4], The authors say that vulnerability is neither a new nor a modern concept in Information Technology. Before any potential risk or hazard happens, vulnerability is the one that helps the attacker or someone to cause any attacks to the application or to an organization. Then the organization has to perform penetration testing to identify the security vulnerabilities or any threats to those resources and finally have to come up with a solution to mitigate the risk or impact that is caused by the vulnerability. By eliminating those vulnerabilities from the resources, the organization or the application can be safe from any potential cyber-attacks.In [5], the author mentioned that the many of the web applications are lack in security is because of the input validation in client side which is becoming vulnerable to attacks like Cross-Site Scripting (XSS), SQL injection (SQLI) and Buffer Overflow (BOF). The techniques that used in identifying the web application vulnerabilities are static analysis which is reviewing the source code of the application to identify any security vulnerabilities, also known as "Whitebox testing" andDynamic analysis involves the attacker analyzing the application behavior and framing the steps to pinpoint security vulnerabilities. In [6], the author discussed issues and flaws in an application arising from testers and developers who failed to notice the side effects of writing unprotected and temporary code during the application development process. Here, the author introduced a flow process for creating a threat model. To bypass it, sub-scenarios include brute-forcing the OTP case for login, exploiting the site with command injection attacks, or brute-forcing the login password via a dictionary attack. Additionally, attention must be given to dependency testing, implementation testing, user interface testing, and design testing, which is a complex task requiring developers and software architects to design secure software. In [7], the authors mentioned how penetration testing is currently useful in identifying security vulnerabilities in web applications. However, they noted that there is no guarantee of finding all the vulnerabilities present in the applications. The authors provided a simple source code example to explain the attacks and briefly outlined the methodology behind each step. In [8], the authors described how Vulnerability Assessment and Penetration Testing help access sensitive information within an application and identify potential security vulnerabilities. They also discussed types of vulnerabilities by referencing the OWASP top-10 (2013). However, they did not mention the processes for finding and exploiting these vulnerabilities. General steps included in the VAPT process were outlined, along with its advantages and disadvantages, as well as the benefits and features of VAPT. In [9], the authors discussed web application vulnerabilities such as injection attacks—including SQL injection, XSS, IDOR, CSRF—and some misconfigurations that can occur due to testers and developers. They defined these vulnerabilities theoretically but did not specify how to exploit them. They selected the best tools and noted the type, availability, price, version tested, and the functions each performs. In [10], the authors proposed a seven-step penetration testing model: Preparation -> Anonymity -> Foot printing -> Analysis -> Exploiting -> Reporting -> Advisory. This paper outlines the flow of activities that should be followed for a better understanding of the process and for ease of use. This penetration testing helps organisations identify potential security vulnerabilities before attackers do; although it has limitations, there is a need to be...

III. Proposed Work

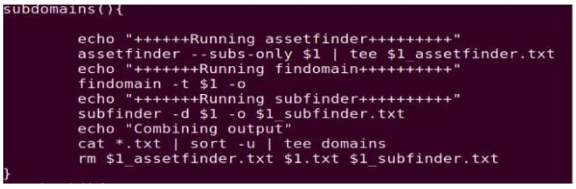
The tools start to gather the information from the domain name, Once the collection is done then the individual folders are automatically created and placed on the main recon folder. After execution of the script, a few tools take the API keys to clear the subdomains, hosts after scanning. We have used open-source tools to make a security framework. which helps to change the process of finding the vulnerabilities in a short period of time by increasing the number of threads. All the tools are written in the bash language so that it becomes easy to integrate and execute the script in the Debian-based systems. There are very few prerequisites to execute the script in the environment. There are a few tools that need to be executed in the Golang environment. The installation of Go is dependent on the system architecture. We have majorly divided the used tools into certain categories. Tools which are responsible for collecting the subdomains, resolving the collected domains, Port scanning, IP"s collection, sorting unique & new target assets (ASN).

3.1 Port Scanning

A port scan is a technique for discovering whether network ports are open. Port scanning is taking to knocking on doors to determine whether somebody is home since ports on a computer are where information is transferred and received. A port scan on a network or server indicates which ports are open and listening (receiving data), as well as the presence of security mechanisms such as firewalls between the sender and the destination. These are referred to as fingerprints. It's also useful for checking network security and the effectiveness of the system's firewall. Because of its feature, it is also a popular reconnaissance tool for attackers looking for a weak point of access to a machine. The tool used is Naabu.

3.2 Subdomain Gathering

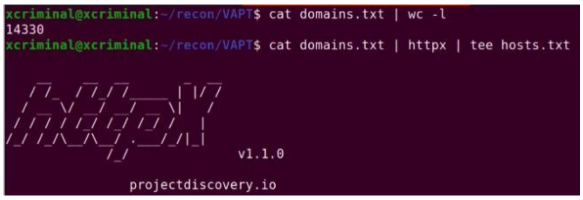
Here we have used the 4 open-source tools to scan the target and get information about subdomains. In some cases, there might be a chance the tool can miss the asset. so, we ll be cross-checking the results with CRT. The tools used are Asset Finder, Sub Finder, Amass, and Fin domain.



```
subdomains(){

        echo "++++++Running assetfinder+++++++++"
        assetfinder --subs-only $1 | tee $1_assetfinder.txt
        echo "+++++++Running findomain++++++++++"
        findomain -t $1 -o
        echo "+++++++Running subfinder++++++++++"
        subfinder -d $1 -o $1_subfinder.txt
        echo "Combining output"
        cat *.txt | sort -u | tee domains
        rm $1_assetfinder.txt $1.txt $1_subfinder.txt
}
```

3.4 Resolving Live Hosts:

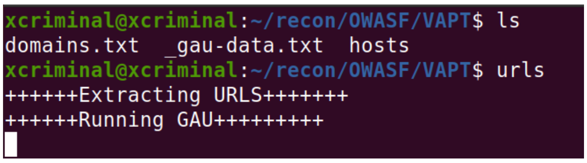
After collecting the domains from the target, Filter-resolved will help us to resolve the domains and help to gather the live hosts from the domains. It is very common that tools use the Brute force mechanism to discover the possibility of new subdomains by using the existing wordlist. We have used HTTPX, Filter-resolver.



```
xcriminal@xcriminal:~/recon/VAPT$ cat domains.txt | wc -l
14330
xcriminal@xcriminal:~/recon/VAPT$ cat domains.txt | httpx | tee hosts.txt

    __    __  __       _  __
   / /_  / /_/ /_____ | |/ /
  / __ \/ __/ __/ __ \|   /
 / / / / /_/ /_/ /_/ /   |
/_/ /_/\__/\__/ .___/_/|_|
             /_/              v1.1.0

                projectdiscovery.io
```

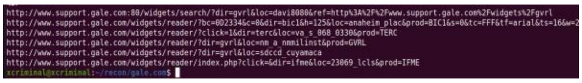
3.5. Gathering Endpoints

The endpoints play an important role in finding out the vulnerability in web applications. All these endpoints are tested according to the number of parameters present in the URL. These URLs are taken into the testing environment where the temp /opt. Now, they are tested against the wordlists that are present in the nuclei database. The tools used are Wayback URLs.



```
xcriminal@xcriminal:~/recon/OWASF/VAPT$ ls
domains.txt  _gau-data.txt  hosts
xcriminal@xcriminal:~/recon/OWASF/VAPT$ urls
++++++Extracting URLS+++++++
++++++Running GAU+++++++++
```

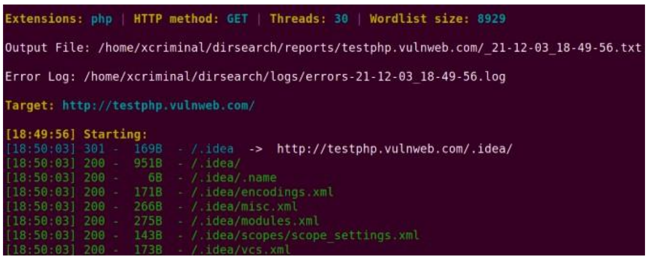
After a short period of time, the tool will initiate the collection of all endpoints and sort them according to parameters.



```
http://www.support.gale.com:80/widgets/search/?dir=gvrl&loc=davi8080&ref=http%3A%2F%2Fwww.support.gale.com%2Fwidgets%2Fgvrl
http://www.support.gale.com/widgets/reader/?bc=0D2334&c=0&dir=bic1&h=125&loc=anaheim_plac&prod=BIC1&s=0&tc=FFF&tf=arial&ts=16&w=2
http://www.support.gale.com/widgets/reader/?click=1&dir=terc&loc=va_s_068_0330&prod=TERC
http://www.support.gale.com/widgets/reader/?dir=gvrl&loc=nm_a_nmmilinst&prod=GVRL
http://www.support.gale.com/widgets/reader/?dir=gvrl&loc=sdccd_cuyamaca
http://www.support.gale.com/widgets/reader/index.php?click=&dir=ifme&loc=23069_lcls&prod=IFME
xcriminal@xcriminal:~/recon/gale.com$ 
```

3.6 Directory Brute force

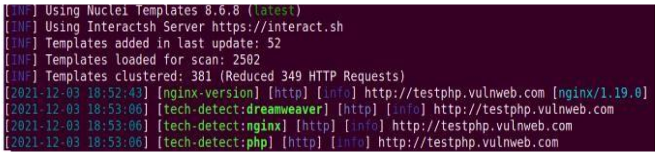
This module helps us to return or extract URLs all the subdirectories of a particular domain that has sensitive data or may have vulnerabilities along with those HTTPS status codes. The tool used is Dirsearch.



```
Extensions: php | HTTP method: GET | Threads: 30 | Wordlist size: 8929

Output File: /home/xcriminal/dirsearch/reports/testphp.vulnweb.com/_21-12-03_18-49-56.txt

Error Log: /home/xcriminal/dirsearch/logs/errors-21-12-03_18-49-56.log

Target: http://testphp.vulnweb.com/

[18:49:56] Starting:
[18:50:03] 301 -  169B  - /.idea  ->  http://testphp.vulnweb.com/.idea/
[18:50:03] 200 -  951B  - /.idea/
[18:50:03] 200 -    6B  - /.idea/.name
[18:50:03] 200 -  171B  - /.idea/encodings.xml
[18:50:03] 200 -  266B  - /.idea/misc.xml
[18:50:03] 200 -  275B  - /.idea/modules.xml
[18:50:03] 200 -  143B  - /.idea/scopes/scope_settings.xml
[18:50:03] 200 -  173B  - /.idea/vcs.xml
```

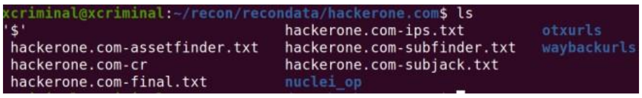
3.7 Vulnerable Template Scanning

Template scanning will help us to scan for existing vulnerabilities, which range from low severity to high. It becomes easy for the security auditors to scan and fix the vulnerabilities according to their severity of the vulnerabilities. Finally, all the vulnerabilities are saved into individual files, with a unique name. Template scanning works with the existing model of URL schemes or predefined directories in the web architecture. Now, if there are any pingbacks from the server, then the pings will be received from the web interface interact, which is developed by Project Discovery.



```
[INF] Using Nuclei Templates 8.6.8 (latest)
[INF] Using Interactsh Server https://interact.sh
[INF] Templates added in last update: 52
[INF] Templates loaded for scan: 2502
[INF] Templates clustered: 381 (Reduced 349 HTTP Requests)
[2021-12-03 18:52:43] [nginx-version] [http] [info] http://testphp.vulnweb.com [nginx/1.19.0]
[2021-12-03 18:53:06] [tech-detect:dreamweaver] [http] [info] http://testphp.vulnweb.com
[2021-12-03 18:53:06] [tech-detect:nginx] [http] [info] http://testphp.vulnweb.com
[2021-12-03 18:53:06] [tech-detect:php] [http] [info] http://testphp.vulnweb.com
```

Since the nuclei_op folder consists of vulnerabilities and sensitive information. We"ll not be posting it here and obeying the company s rules of engagement.



```
xcriminal@xcriminal:~/recon/recondata/hackerone.com$ ls
'$'                            hackerone.com-ips.txt          otxurls
 hackerone.com-assetfinder.txt   hackerone.com-subfinder.txt    waybackurls
 hackerone.com-cr                hackerone.com-subjack.txt
 hackerone.com-final.txt         nuclei_op
```

4. Conclusion

OWASP serves as a comprehensive framework that can serve as a backup when access control measures and misconfigurations occur in the development cycle. It immediately notifies the security auditor about the vulnerabilities by doing template scanning. As previously mentioned, the web vulnerabilities keep rising depending on the added features and functionalities. OWASF is unique in the way that it can exist as a multi-purpose framework used for offensive purposes. Red teamers and penetration testers can run this framework on the target system to uncover potential misconfigurations in the system that they could exploit. Further auditors can use this to mitigate the vulnerabilities as soon as the applications proceed into the testing stage. So, it helps them to keep the security updates up to date.

References

[1]. Bacudio, A. G., Yuan, X., Chu, B. T. B., & Jones, M. (2011). An overview of penetration testing.

International Journal of Network Security & Its Applications, 3(6), 19.

[2]. Al Shebli, H. M. Z., & Beheshti, B. D. (2018, May). A study on the penetration testing process and tools. In 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT) (pp. 1-7). IEEE.

[3]. Goel, J. N., &Mehtre, B. M. (2015). Vulnerability assessment & penetration testing as a cyber defence technology. Procedia Computer Science, 57, 710-715.

[4]. Kovacs, S., & Darabont, A. (n.d.). SYMPOSIUM SERIES NO 159 Vulnerability assessment - one step further towards better safety. https://www.icheme.org/media/8972/xxiv-poster-07.pdf

[5]. ĐURIĆ, Z. (2014). WAPTT-Web application penetration testing tool. Advances in Electrical and Computer Engineering, 14(1), 93-102.

[6]. Thompson, H. H. (2005). Application penetration testing. IEEE Security & Privacy, 3(1), 66-69.

[7].Hal fond, W. G., Choudhary, S. R., &Orso, A. (2009, April). Penetration testing with improved input vector identification. In 2009 International Conference on Software Testing, Verification and Validation (pp. 346-6355). IEEE.

[8]. Shinde, P. S., &Amrapurkar, S. B. (2016, February). Cybersecurity analysis using vulnerability assessment and penetration testing. In 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave) (pp. 1-5). IEEE.

[9].Ferreira, A. M., & Kleppe, H. (2011). Effectiveness of automated application penetration testing tools.

[10].Ami, P., & Hasan, A. (2012). Seven phrase penetration testing model. International Journal of Computer Applications, 59(5), 16-20.

[11].Sandhya, S., Purkayastha, S., Joshua, E., & Deep, A. (2017, January). Assessment of website security by penetration testing using Wireshark. In 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 1-4). IEEE.