Design Engineering

ISSN: 0011-9342 | Year 2021
Issue: 9 | Pages: 17334 - 17342

Offensive Web Application Security Framework

Atharva Mangesh Kumar Agrawal[1], Durga Bhagavan Bolli[2], ChakkaSai Teja[3], Tushar Parmanand Budhwani[4], Lakshitaa Sehgal[5], Jash Nimesh Dharia[6], Anshal Aggarwal[7].

[1]Vellore Institute of Technology,
[2,3] BML Munjal University,
[4,6]Thadomal Shahani Engineering College,
[5,7]University Institute of Engineering and Technology Chandigarh.

ABSTRACT

Offensive Web Application Security Framework(OWSAF) is a complete offensive security framework that helps to detect vulnerabilities in the web app and helps in real-time security evaluation, which can be used for detecting security misconfigurations. The result helps us to understand the attack vectors from the attackers via a web application. The main aim of OWSAF is to give a complete guide on existing threats in the application using the templates.

Keywords: Offensive: Attack, Web Application, Framework , Security, OWSAF

I. INTRODUCTION

As there is an increment in usage of web applications static and dynamic. Since, there are nessus, netsparker, acunetix, to find out the vulnerabilities and suggest mitigations but still there are a lot of web-based attacks which are taking place. Attackers always use complicated tools and attack frameworks, so that they can attack with ease. Both Nessus and Netsparker allow auditors to configure the framework and plan scheduled scanning based on target scope with flexible scanning methods but in a complicated way. There are other frameworks that will help to scan the targets with multiple features with or without authentication. Unfortunately, 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 operations center environment. As a result, to fill the void created by existing access control systems so, we want 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 both web and network penetration testing, misconfiguration evaluations.

II. Background

The Offensive Web Application Security framework helps to scan the web application to find the potential risks and vulnerabilities associated with the application. Which is an indirectly potential threat to the organization and its users. The framework helps to gather all the internal assets of the

ISSN: 0011-9342 | Year 2021
Issue: 9 | Pages: 17334 - 17342

application for example endpoints, files, folders, API endpoints. Now, after gathering the assets the scripts start executing and performing their individual tasks. There are a set of open-source tools that help us to integrate and make them work parallelly. Attackers always use the latest tactics, techniques to obtain access to the target system and maintain persistent access. This is called exploitation, in case of further exploitation of systems either horizontally, vertically which leads to the post-exploitation. The OWASF will identify the exploitable cases in the first place itself.

III. Scanning

OWASF works as an automated information-gathering tool while performing reconnaissance during penetration testing. Scanning is done after planning the flow of the penetration testing according to the architecture of the domain. In case of a wide scope, the scanning would take some time to gather the complete information about the target as per the technologies used to develop the application. In this phase, the framework identifies the technologies, request time, status code, CNAME, IP address, subdomains, open ports, content length. After gathering the complete information about the target then the framework uses the templates to scan the obtained results. Here the scanning depends upon the type of scan engine. Scan engine includes low-level to high-level scanning of the target. Since the framework has highly configurable scan machines based on the YAML language, which actually allows security auditors to make new scan machines depending upon the requirement.

IV. Numeration

This process includes the further extraction of the details of the domain. It includes the subdomains, services, and their versions. The whole framework is made of open-source tools to make it free of cost for security auditors. The framework has the option to integrate the other tools to automate and add different tools provided the tools are written in bash. This process is further divided into active and passive. In this process, we are going to collect the passive details of the domain. While performing the enumeration with the third-party tools, they create a lot of traffic inside the network if it is active. Since once we have the passive details the rest of them can be retained by the active enum. The challenge comes once it is getting automated. In the automation the data is getting duplicated and consuming the storage, now all this can corrupt the entire enumeration results. So, the OWASF takes care of duplication of endpoints while gathering and then stores them.

V. 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. Penetration testing is different compared to functional security testing, the security functional testing helps to know whether the security measures have been taken effectively or not in an application while that of penetration testing tells us that how difficult it is for someone

ISSN: 0011-9342 | Year 2021
Issue: 9 | Pages: 17334 - 17342

to access and gain control of the organization against to unauthorized access to them. 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. Penetration testing needs to be conducted frequently to find the security risks to the application and have to manage them to accomplish higher standards of security. 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 are needed 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 has been becoming very complex which can also be prone to security threats by the 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 presence due to the system complexity. The author is describing that it is good to identify the security vulnerabilities and to prevent them before anyone can take advantage of them. In order to provide a proactive defense to security threats, the organization has to implement strong protocols in terms of security. 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. Things in vulnerability assessment are common with risk assessment, those assessments are performed based on the steps like gathering all the information and assets/resources about the target and giving a rank order to those resources based on the importance and quantifiable value to those resources.

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. The authors described the methodology that needs to be followed and management to take care of the entire penetration testing process.

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” and

ISSN: 0011-9342 | Year 2021
Issue: 9 | Pages: 17334 - 17342

dynamic analysis where the attacker analyses the application behaviour and frames the attacking steps to find the security vulnerabilities. The author mentioned these three vulnerabilities and the flow of process to identify them.

In [6], the author discussed the issues and flaw in an application that arises because of the testers and developers who were failed in noticing the side-effects of writing the unprotected and temporary codes while developing the applications. Here, the author introduced a flow of process to create a threat model. The first and basic scenario is to bypass the authentication and then gaining the access to the user's account. In order to bypass it, the sub scenarios are brute-forcing the OTP case of login or exploiting the site with any command injection attacks or by brute-forcing the login password via dictionary attack. Also, need to look upon the dependency testing, implementation testing, User Interface testing, and design testing which is a complex task where the developers and software architectures have to design a secured software.

In [7], the authors mentioned that how the penetration testing is useful nowadays in identifying the security vulnerabilities in web applications. Although they said that there is no guarantee in finding out the all the vulnerabilities that are present in the applications. Here, the penetration testing process mentioned is after selecting the target -> Information gathering -> Attack generation on the application and then analysis should be done on the responses of the attacks, the pentesters will examine it to know whether the attack is successful or not and finally reporting the identified/discovered vulnerabilities. The authors took a simple source code to explain the attacks that how it is working and briefly explained the methodology of the above each steps.

In [8], the authors described how the Vulnerability Assessment and Penetration Testing helps in accessing the sensitive information of an application and in finding the potential security vulnerabilities. They also discussed the types of vulnerabilities by referencing the OWASP top-10 (2013); they haven't mentioned the process of finding and exploiting the vulnerabilities. They only described the types of vulnerabilities that can exist in an application and the general steps that included in the process of VAPT and its advantages & disadvantages and also the benefits and features of VAPT.

In [9], the authors discussed about the web application vulnerabilities like injection attacks such as SQL injection, XSS, IDOR, CSRF and some misconfigurations which can be encountered because of the testers and developers. The author targeted a vulnerable application to perform attacks in identifying the vulnerabilities, they performed with all the available tools in the market and mentioned the some of the best tools among them for penetration testing. They defined the vulnerabilities what exactly as they are theoretically, not defined how to exploit them. They selected the best tools and mentioned about the type, availability, price and version tested and what the work it does.

In [10], the authors proposed the penetration testing model in seven steps: Preparation -> Anonymity ->Footprinting -> Analysis -> Exploiting -> Reporting -> Advisory. This paper describes the flow of activities that need to be followed for the better understanding of the process and for ease of use. Also, about the black box & white box testing and the limitations that exist in the present penetration testing. This penetration testing helps an organization in identifying the potential security vulnerabilities before any attacker does, although it has limitations but need to be

ISSN: 0011-9342 | Year 2021
Issue: 9 | Pages: 17334 - 17342

performed with well experienced pentesters to secure the applications in the organization.
In [11], the authors proposed the work which involves the usage of wireshark in the information gathering phase of penetration testing. They demonstrated their work by considering a vulnerable application, they captured the packets and filtered the POST method packets and examined them. They observed that entire data is clearly seen in clear text which is the traffic transmitting from sender to receiver. Here, the entire conversation and credentials too between them can be seen by any third party simply by analyzing the traffic. So, it's always important to encrypt the traffic too, as nowadays the usage of Internet and online work is increasing where the payment transactions are also happening, even that can also be manipulated by the attacker.
In [12], the authors demonstrated how they were able to exploit the SQL Injection vulnerability in one of the financial based web applications in the Bangladesh region. Their process is to attempt the vulnerability exploitation and then to extract the data from the application, as SQL is related to the database which contains all the data about the application and its users which is very confidential as it contains their personal data. Here, they showed the SQLi attack with IDs and without IDs, as they are using the black box approach they won't know whether the application contains IDs or not. Firstly, they passed the single quote(') and the application responded with the error which verifies the presence of SQLi vulnerability. After that they crafted their payload according to the application response and then injected it. They showed step by step how they made the payload based on the response.
In [13], the authors have mentioned the SQL Injection attack and its types: first order attack, second order attack and lateral injection. They explained the attack stepwise using the HAVIJ tool. Here, we don't need to inject any payloads as it was an automated tool. Just copy and paste the URL in the target field and the results are in the Tables tab and can see the databases that are available in that application. There is a case where the passwords are encrypted, so in that case decrypt using various algorithms which can give the clear text. There are other tools which are available for similar work like BSQL Hacker, SQL Power Injector and Marathon Tool.

In [14], the authors discussed the web application architecture then explained about the detection of security vulnerabilities in web applications using black box automation and manual penetration testing. Also, mentioned about the OWASP Top 10(2010) and the vulnerability rankings. They proposed a framework with four phases i.e., selection of tools and target application for testing -> performing the automated black box testing -> manual pentesting -> analyzing the results obtained from the previous phases. With the help of their proposed framework, they were able to find out the five different types of vulnerabilities.

In [15], the authors selected six pentesting frameworks/methodologies with reference to ISSAF and OWASP's OTG and evaluated based on the six categories: Standard or Guide, Methodology, Framework, Application Suite, Framework encapsulates Methodology and Manual or Resource. They noticed that most of the frameworks are not generalized across the problem domains as like a generic pentesting framework, their next step is to test these frameworks in the real world to understand better results wise.

ISSN: 0011-9342 | Year 2021
Issue: 9 | Pages: 17334 - 17342

VI. 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 require the API keys to validate the subdomains, hosts after scanning. We have used open-source tools to make a security framework. which helps to automate 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 golang 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).

VII. 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.

VIII. 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, Tools used are Assetfinder, Subfinder, Amass, and findomain.

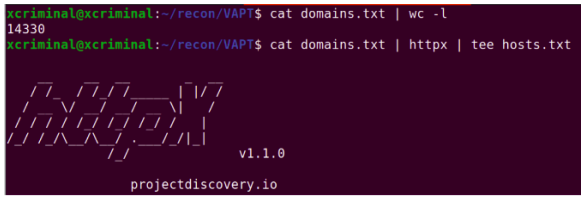
```
subdomains(){

        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
}
```

IX. Resolving Live Hosts:

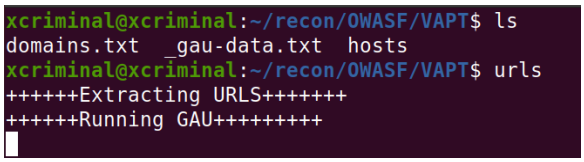
After collecting the domains from the target, Filter-resolved will help us to resolve the domains and help to gather the live hosts from the domains. It is very common that tools use the Bruteforce mechanism to discover the possibility of new subdomains from using the existing wordlist. We have used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

ISSN: 0011-9342 | Year 2021
Issue: 9 | Pages: 17334 - 17342

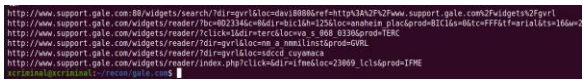


X. Gathering Endpoints

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 which are present in the nuclei database. The tools used are Wayback URLs.

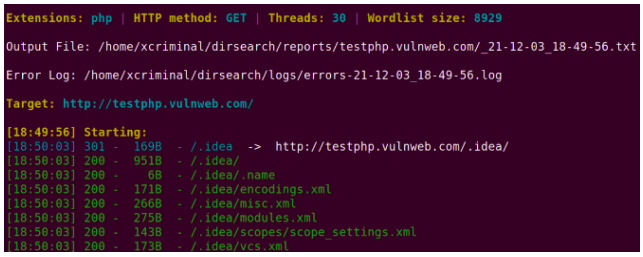


After taking a short period of time, the tool will initiate the collection of all endpoints and sort them according to parameters.



XI. Directory Bruteforce

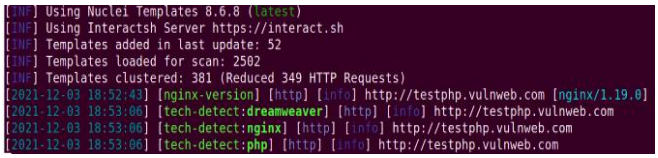
This module helps us to return or extract URLs all the subdirectories of a particular domain that has sensitive data or may have vulnerabilities along with those HTTPS status codes. The tool used is Dirsearch.



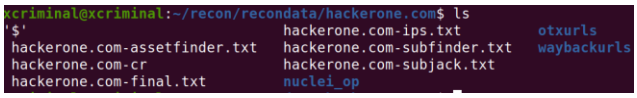
XII. Vulnerable Template Scanning

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 the 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 web interface interact which is developed by projectdiscovery.

ISSN: 0011-9342 | Year 2021
Issue: 9 | Pages: 17334 - 17342



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



XIII. Conclusion

OWASF serves as a comprehensive framework that can serve as a backup when access control measures and misconfigurations occurs in development cycle. It immediately notifies the security auditor about the vulnerabilities by doing the template scanning. As previously mentioned the web vulnerabilities keeps rising depending upon 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 proceeds into the testing stage.So, that 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 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 defense 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] Halfond, 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. 346-355). IEEE.

[8] Shinde, P. S., &Ardhapurkar, S. B. (2016, February). Cyber security 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.

ISSN: 0011-9342 | Year 2021
Issue: 9 | Pages: 17334 - 17342

[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.

[12] Farah, T., Alam, D., Kabir, M. A., & Bhuiyan, T. (2015, October). SQLi penetration testing of financial Web applications: Investigation of Bangladesh region. In *2015 World Congress on Internet Security (WorldCIS)* (pp. 146-151). IEEE.

[13] Nagpal, B., Singh, N., Chauhan, N., & Panesar, A. (2015, May). Tool based implementation of SQL injection for penetration testing. In *International Conference on Computing, Communication & Automation* (pp. 746-749). IEEE.

[14] Awang, N. F., &AbdManaf, A. (2013, September). Detecting vulnerabilities in web applications using automated black box and manual penetration testing. In *International Conference on Security of Information and Communication Networks* (pp. 230-239). Springer, Berlin, Heidelberg.

[15] Shanley, A., & Johnstone, M. N. (2015). Selection of penetration testing methodologies: A comparison and evaluation.