Web Application Scanning Framework

Prof. Priyanka H. Shingate[1], Hrishikesh P. Jadhav[2], Rushikesh S. Jagdhane[3], Akash S. Trimbake[4], Mayur M. Sawant[5]

[a]Zeal College of Engineering Research, , Pune, 411041, Maharashtra, India

Abstract

This paper introduces a framework that utilizes open-source tools for conducting reconnaissance and vulnerability assessment in web-based systems. The framework is hosted as a website and offers a user-friendly interface for cybersecurity practitioners to identify potential security risks. By integrating various open-source tools, the framework enables efficient and effective information gathering and vulnerability scanning. This unified solution contributes to the field of cybersecurity and provides a valuable resource for practitioners.

Keywords: Cybersecurity, web-based systems, reconnaissance, vulnerability assessment, open-source tools, information gathering, security risks, framework, user-friendly interface, flexibility customization.

1. Introduction

The increasing dependence on web-based systems in vari-ous aspects of modern life, such as e-commerce, banking, so-cial networking, and communication, has given rise to concerns about cybersecurity. Cyberattacks, which can result in informa-tion theft, data breaches, system disruption, and ransomware attacks, pose significant risks to both organizations and indi-viduals. To combat these threats, cybersecurity practitioners require effective tools and techniques for gathering information and assessing vulnerabilities in web-based systems, in order to identify potential weaknesses and vulnerabilities.

. In response to these challenges, the authors propose a frame-work that aims to assist cybersecurity practitioners in conduct-ing reconnaissance and identifying vulnerabilities in web-based systems. The framework is built on open-source tools that are widely used in the industry, providing flexibility, extensibility, and cost-effectiveness, making it accessible to a wide range of users. It is designed to be hosted as a website, providing a user-friendly interface that simplifies and optimizes the overall pro-cess. One of the key advantages of this framework is the in-tegration of various open-source tools, leveraging their unique capabilities and functionalities to enable efficient and effective reconnaissance and vulnerability assessment in web-based sys-tems. By integrating these tools into a single framework, the authors provide cybersecurity practitioners with a consolidated and streamlined approach to conducting reconnaissance and vulnerability assessment.

. One of the key advantages of this framework is the integra-tion of various open-source tools, leveraging their unique capa-bilities and functionalities to enable efficient and effective re-connaissance and vulnerability assessment in web-based sys-tems. By integrating these tools into a single framework, the authors provide cybersecurity practitioners with a consolidated and streamlined approach to conducting reconnaissance and vulnerability assessment.

. The user-friendly interface of the framework allows cyberse-curity practitioners to easily input their target systems, config-ure scanning options, and initiate the reconnaissance and vul-nerability assessment process. The framework then orchestrates the execution of the integrated tools, automating the scanning and assessment tasks, and providing consolidated results for analysis. The results can include information about potential vulnerabilities and their severity. Furthermore, the framework allows for extensibility, enabling cybersecurity practitioners to add or customize tools based on their specific requirements.

. Overall, the authors' framework offers a practical and com-prehensive solution for organizations and individuals looking to enhance their cybersecurity measures and protect their on-line assets.

2. RELATED WORK

The top 10 web security vulnerabilities, as summarized by the Open Web Application Security Project (OWASP)[9], in-clude the SQL injection and the cross-site scripting (XSS). These two vulnerabilities are considered the most common and harmful among the ten listed. OWASP Top Ten for 2021:

1. Injection
2. Broken Authentication and Session Management
3. Cross-Site Scripting (XSS)
4. Broken Access Control
5. Security Misconfiguration
6. Insecure Cryptographic Storage
7. Insufficient Transport Layer Protection
8. Unvalidated and Unsanitized Inputs
9. Insufficient Logging and Monitoring

Preprint submitted to Astronomy & Computing

10. Using Components with Known Vulnerabilities

Haibo Chen, Junzuo Chen, Jinfu Chen, Shang Yin, Yiming Wu, Jiaping Xu in An Automatic Vulnerability Scanner for Web Applications [1] proposed a system to find vulnerabilities in a system. They mainly focus on SQL injection and Cross-Site Scripting.

. Ahana Roy, Louis Mejia, Paul Helling, Aspen Olmsted in Automation of Cyber-Reconnaissance [4]. The project implementation involves the use of Jsoup, a Java library that provides an API for extracting and manipulating data using DOM, CSS, and jQuery-like methods. The tool scrapes and parses HTML from the company URL provided by the user and uses various meth-ods to extract information such as domain names, host names, IP addresses, and mail servers with preferences. The tool also has a method for searching for company files according to the format chosen by the user from the command line. Addition-ally, the tool can extract results from WHOIS search pages us-ing regular expression matching, which provides useful infor-mation about the organization in a consolidated manner.

3. Proposed System

A. Framework of Proposed System

The proposed scanner is designed to automatically identify potential vulnerabilities of a target web application in a thorough manner. The flow of the scanner is illustrated in Figure
1. Users must first specify a specific target and gather relevant information, such as subwebsites or subdomains. The collected information is then deduplicated and stored in a database to support subsequent scanning and detection. Additionally, other ports and services discovered during the information collection process are automatically added to the scanning list, expanding the scanning scope and increasing the effectiveness of vulner-ability detection. The scanner ultimately reports the collected information and detailed detection results on a web page.

B. Modules Design

The proposed scanner for web vulnerabilities is built using the Python programming language and Django [14] as the back-end framework, with a Browser/Server (B/S) architecture. The server-side hosts the database, Reconnaissance, and vulnera-bility detection modules, while users can manage running tasks through their browser. The main function modules are the Infor-mation Collection module, Vulnerability Detection module, and Tasks and Targets Management module. Each of these modules will be discussed in detail below:

1. Reconnaissance Module: The Reconnaissance module serves two main functions: asset collection and port scanning/service identification. Asset collection expands the target asset scope by exploring subdomains, utilizing tech-nologies such as DNS enumeration, online interfaces, DNS querying, and search engines. During penetration testing, information about related ports and inside services is required to identify potential vulnerabilities. Tar-get ports are scanned, and fingerprint recognition is used to access open services. This allows for a series of test-ing activities to be carried out to verify whether the open services.
2. Vulnerability Detection Module: Once the related infor-mation of the target is collected, a wide range of testing objects is available for vulnerability detection. The pro-posed scanner provides two detection modules: Nuclei and OWASP ZAP. The combination of both of these scanners helps in finding bugs and misconfigurations ranging from informational bugs to critical OWASP top 10 listed bugs. The modular structure of the system also allows us to add more scanners for increasing findings of the system.

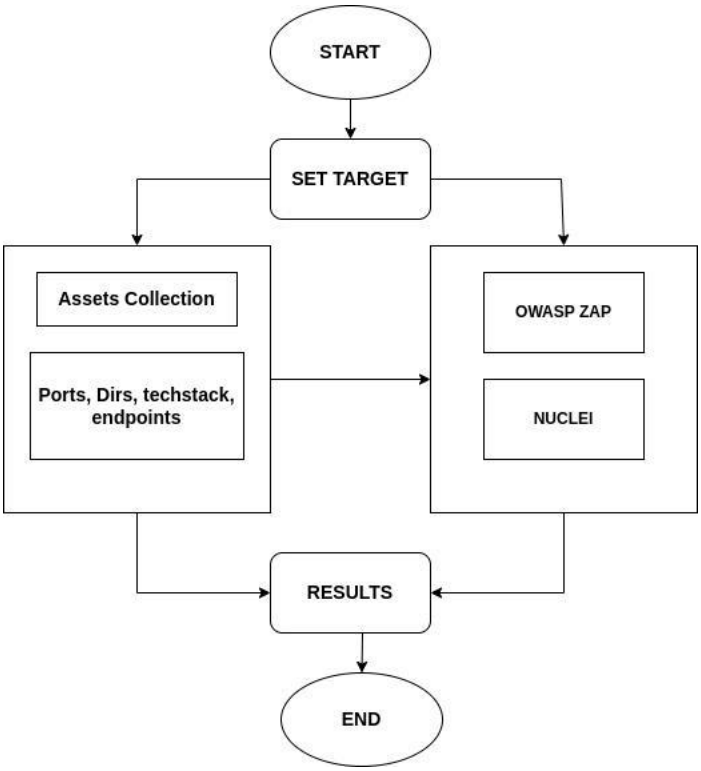

Figure 1: flow of system

C. System Acrhitecture:

1. The client-side of the proposed system is implemented us-ing React JS, which is a popular front-end JavaScript li-brary for building user interfaces. React JS allows for the creation of reusable UI components that can be used to build complex user interfaces. The client-side of the pro-posed system will be responsible for displaying the user interface, handling user inputs, and making requests to the server-side of the system for data and other resources.
2. The server-side of the proposed system is implemented using Django, which is a high-level Python web frame-work for building web applications. Specifically speak-ing, the server-side is based on Django REST framework

(DRF)[14]. DRF follows the principles of Representa-tional State Transfer (REST) which makes it easy to de-velop scalable and maintainable APIs.

3. The proposed system also makes use of several external tools and services such as Zap and Nuclei for vulnerability scanning, as well as various libraries and packages for handling data, authentication, and other aspects of the system.
4. Overall, the proposed system architecture is designed to be scalable, modular, and easy to maintain. The use of modern web technologies and frameworks allows for rapid development and deployment of the system, while also ensuring that it is secure, efficient, and reliable.

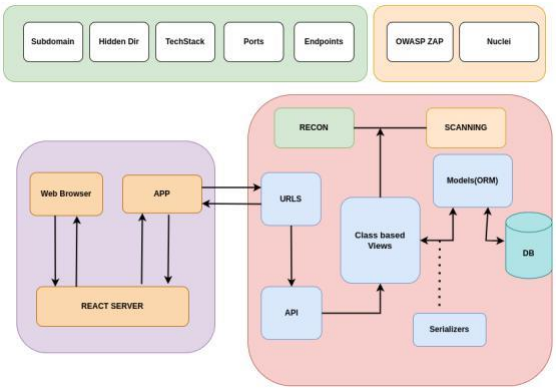

Figure 2: System Architecture

4. IMPLEMENTATION

Test and Evaluation

To determine the effectiveness of the proposed scanner, a sequence of tests were conducted on a specific website called the Test site for Acunetix WVS (http://test.vulnweb.com), which is widely utilized for testing vulnerability scanners.

A. Running Environment

The proposed scanner was run on a machine serving Debian Linux and equipped with 16 GB of RAM and an AMD Ryzen 5 5500U processor running at 3.6 GHz. Python version 3.9.2 was adopted as the programming language.

B. Target Input

First, the user inputs the URL of the web application of the target organization. The input only contains the base URL of the target, with no subdomain information given. (Refer to Fig 3)

C. Subdomain Enumeration

As soon as the target URL is given, the subdomain collec-tion phase automatically starts. The subdomains are collected from various sources, which includes brute-forcing the URL with wordlists [16]. The user is then presented with a list of active subdomains in the output (Refer to Fig 3).

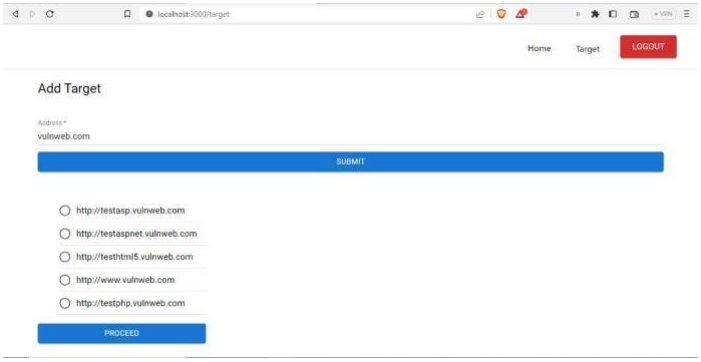

Figure 3: Adding Target.

D. Information Collection

After selecting the required subdomain, further enumeration on that particular subdomain is started. Various information like directories, ports, technology stack, and endpoints are collected (Refer to Fig 4).

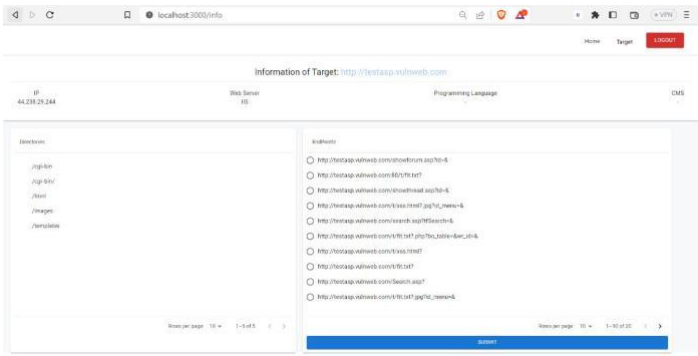

Figure 4: Information Collection

E. Vulnerability Detection

Once the data is collected, vulnerability assessment (Refer to Fig 5) is performed on one of the endpoints found. This endpoint is an ideal point where the user can give untrusted input to the application.

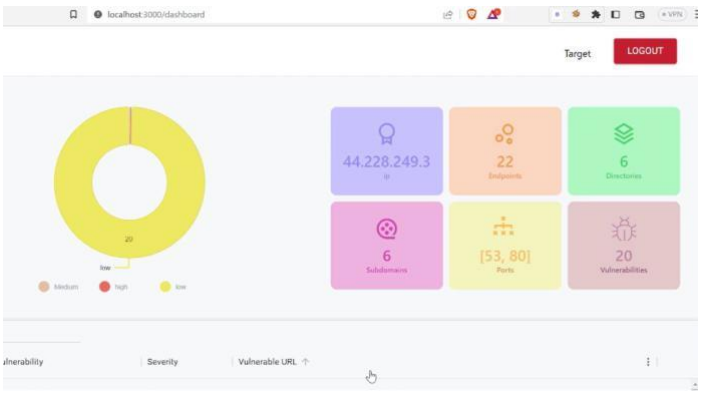

Figure 5: Vulnerability Detection

5. CONCLUSION

The vulnerability scanner is a significant tool for web se-curity assurance, aims to detect possible vulnerabilities in ad-vance. At present, most of the scanners are only focuses on sin-gle target without utilizing other useful information. In this pa-per, we have proposed an automatic web vulnerability scanner which integrates information collection with vulnerability de-tection. Moreover, the vulnerability detection in proposed scan-ner is guided by the collected useful information. Thus, once the proposed scanner obtains a specific target, the deeper and comprehensive detection will be constructed, which may lead to an ideal performance. In addition, the further experimental testing results prove that proposed scanner achieves remarkable effectiveness and can be feasible implemented in practice.

6. REFERENCES

1. Haibo Chen, Junzuo Chen, Jinfu Chen, Shang Yin, Yiming Wu, Jiaping Xu. ”An Automatic Vulnerability Scanner for Web Applications”, 2019, Institute of Electrical and Elec-tronics Engineers: www.ieee.org
2. Trapti Jain, Nakul Jain. ”Framework for Web Applica-tion Vulnerability Discovery and Mitigation by Customiz-ing Rules through ModSecurity”, 2019, Institute of Elec-trical and Electronics Engineers: www.ieee.org
3. Wang Ze. ”Design and implementation of Core Mod-ules of WEB Application Vulnerability Detection Model”, 2019, Institute of Electrical and Electronics Engineers: www.ieee.org
4. Ahana Roy, Louis Mejia, Paul Helling, Aspen Olmsted. ”Automation of Cyber-Reconnaissance”, 2017, Institute of Electrical and Electronics Engineers: www.ieee.org
5. Prof. Sangeeta Nagpure, Sonal Kurkure. ”Vulnerabil-ity Assessment and Penetration Testing of Web Applica-tion”, 2017, Institute of Electrical and Electronics Engi-neers: www.ieee.org
6. Rohan Vibhandik, and Arijit Kumar Bose. ”Vulnerability Assessment of Web Applications - A Testing Approach”, 2015, Institute of Electrical and Electronics Engineers: www.ieee.org
7. Holik, S. Neradova. ”Vulnerabilities of Modern Web Ap-plications”, 2017, Institute of Electrical and Electronics Engineers: www.ieee.org
8. Kali — By Offensive Security: https://www.kali.org/
9. OWASP Foundation. ”OWASP Top 10 - 2021 rc1”: https://owasp.org/Top10/
10. Python Multi Processing programming: https://machinelearningmastery.com/ multiprocessing-in-python/
11. OWASP ZAP API Documentation: https://www.zaproxy.org/docs/api/introduction
12. Django SSE Responses: https://andrewbrookins.com/django/how-does -djangos-streaminghttpresponse-work-exactly/
13. Nuclei template repository: https://github.com/projectdiscovery/nuclei-templates
14. Django REST Framework tutorial: https://www.youtube.com/watch?v=c708Nf0cHrs
15. Json Haddix Recon Methodology: https://www.youtube.com/watch?v=FqnSAa2KmBI
16. Seclist A wordlist repository for Bruteforcing: https://github.com/danielmiessler/SecLists