Comparative Analysis of Penetration Testing Tools in Assessing Web Application Vulnerabilities

***Abstract*— This research analyses and compares the effectiveness of penetration testing tools to detect web application vulnerabilities. The penetration testing tools evaluated were Burp Suite Professional, Burp Suite Community, OWASP ZAP, Wfuzz, Hydra, and SQL Map. These tools have been assessed in a controlled environment. Regardless, DVWA (Damn Vulnerable Web Application a real-world web application, which is equipped with different levels of security option was used for this research. The key performance indicators that were analysed are the detection accuracy, ease of use, capability to integrate with another tool, and capability to generate a report. This research identifies security vulnerabilities which can be exploited and will help to strengthen the organisation's defence system. This initiative-taking approach supports compliance, reduces risks and improves overall security posture.**

Keyword – Web application security, Pen Testing tools, Vulnerability Assessment, Brute Force Attack, Tool Comparison, Ethical Hacking

I. Introduction

Cybersecurity is gaining more attention as a result of all industries' increasing dependence on digital infrastructure. Protecting sensitive data and maintaining the integrity of systems are major concerns for enterprises as cyber threats continue to increase. Penetration testing has become a crucial preventative step among the many tactics used to improve information security. Penetration testing allows security experts to find and fix flaws before malicious actors can take advantage of them by mimicking actual attack situations.

This research investigates the effectiveness of several penetration testing tools when applied to a vulnerable web application configured with varying levels of security. The study includes both open-source and commercial tools, with the aim of evaluating their performance, reporting capabilities, and suitability for different testing environments. The primary objective is to provide a comparative analysis that highlights each tool's strengths and limitations in detecting common web application vulnerabilities.

Through a structured comparative analysis using performance metrics such as detection accuracy, speed, usability, and reporting capabilities, the project aims to determine which tools offer the most reliable and efficient testing. A simulated real-world testing environment, leveraging the Damn Vulnerable Web Application (DVWA) at varying security levels, provides a controlled setting to ensure consistent and objective results of web-based systems.

The significance of this research lies in its practical implications for cybersecurity practitioners and organisations. With a wide range of tools available, selecting the most appropriate one can be challenging. This study offers insights into how different tools perform under realistic conditions, contributing to informed decision-making in the selection and implementation of penetration testing methodologies. Ultimately, the findings aim to support improved vulnerability management and enhanced organisational security posture [1]

II. LITERATURE REVIEW

Comparing penetration testing tools is crucial because web applications are often the most targeted attack surface in organizations. The increasing sophistication of cyber threats demands tools that are not only accurate but also practical, scalable, and dependable. Recent studies highlight the diverse landscape of penetration testing tools, which vary widely in their approaches and effectiveness. Research paper [1] emphasizes this need by showing that no single tool is universally effective, misalignment between tools and use cases can lead to undetected flaws or inefficient testing. Selecting the right tool depends in the specific goals and context of the assessment. In [2] Shelbi and Beheshti's research offer a structured penetration testing methodology – consists of planning, reconnaissance, scanning, exploitation, and reporting that plays as a base foundation for evaluation tools across the different stages. According to Researchers [3] and [4] reveal that tools such as Burb Suite, OWASP ZAP, SQL Map and Wfuzz differ significantly in detection accuracy, speed, usability, and reporting capabilities. Burb Suite excels in manual and detailed analysis, whereas OWASP ZAP provides stronger automation and easier integration into development pipelines, as noted by Kim and Lee [5]. SQL Map is specialized for SQL injection attacks, and Wfuzz effectively supports brute-force enumeration. With many studies focus on controlled test environments like DVWA [6] which may not fully replicate real world application complexity, limiting the applicability of results. [7]Khan and Bangash underline the importance of standardized and clear reporting to translate technical findings into actionable insights for stakeholders, a feature more developed in some tools than others. Moreover, understanding when and how to deploy these tools within the testing lifecycle improves efficiency and effectiveness, reducing redundancy and maximizing coverage.

By applying the 5Ws framework – What, When, Why, Where and How, this research addresses critical questions that facilitate comparative studies. It clarifies the **Why** by highting the need for better – informed tool selection amid rising web threats. Defines the **What** as a comparison of tools based on performance, usability and integration specifies then the **Whe**n through the lens of controlled versus real world testing environments. Focuses on the **How** by targeting both technical professionals and decision makers who benefit from these insights.

Despite the solid foundation provided by existing literature, [8] gaps remain in benchmarking newer tools, enhancing reporting standardization and simulation realistic environments. This study aims to bridge these gaps, providing a comprehensive and practical comparative analysis that supports informed decision-making in securing web applications.

III. METHODOLOGY

Hybrid Methodology

This research employs a hybrid project management approach as we utilize Gantt Charts to monitor and ensure each phase is complete.

Also, we prioritise SecSDLC as our foundational framework and utilise the OWASP Pen Testing workflow to guide our PT testing activities, emphasising the phases below. [9]

- Information GatheringPurpose: This stage sets the foundation for the penetration test, which helps identify potential entry points.
- Review Possible Test MethodPurpose: This is the critical stage where different testing tools can be compared based on their ability to detect various vulnerabilities.
- Have the Attack Methods been Analysed and Investigated?Purpose: Executing an attack, the Tester evaluates whether potential methods have been explored.
- Did the Attack succeed?Purpose: To confirm that the vulnerability is real and exploitable rather than a false positive.
- Evaluate the Tool /Risk Assessment of the AttackPurpose: This stage can be used to compare how different tools are prioritised, which is essential for practical evaluation.

With the Agile structure, we utilised the Kanban Jira Platform as an ideal methodology required in our implementation testing phase, with continuous testing, research, and evaluation as essential when dealing with evolving threats and vulnerabilities.

IV. IMPLEMENTATION & DESIGN

Figure 1: Test Bed for Penetration Testing

Figure 1 shows the design used in implementing this research. Virtual Machines from various locations with different penetration testing tools to assess the target web application, which our stakeholders host; our IP address has been whitelisted. As such, we are the only ones to assess the system and provide reliable and effective penetration testing. Testing penetration testing tools in a controlled environment is the most ethical way of analysing their results.

Figure 2: Performance Criteria Template

Figure 2 shows selection criteria we used to assess each tool in this research.

Assessment criteria are one of the crucial factors to choose when it comes to evaluating the efficiency of the penetration testing tools. Based on the testing, the assessment criteria are categorised into varied factors for each tool are:

Tool Type: It defines open-source or paid version.

Version No: It defines the version of the tool.

Latest Update: It defines the date of the newest update.

Cost Effective: It defines if the tool is a paid version and how much it costs.

Integration: It defines whether this tool can be implemented with other web application penetration testing tools.

Ease of Use: Comprehensive and needs tutorials or a manual to use the application.

Accuracy: This gauges how consistently a tool detects real vulnerabilities without producing false positives.

Performance/speed: It defines how quickly it can perform its operation.

Usability: It defines how the tool is used in terms of installation.

Scalability: It refers to the adaptability of the tool in different operating systems.

Reporting: It defines the reporting format and understandability of the outcome. Out of these criteria, accuracy, integration, performance, and scalability are most important. In addition to these, usability is also important because when it comes to complex testing, usability becomes the prime factor. Performance criteria could be very effective in terms of identifying vulnerabilities in a customised environment, which explains how it can adjust according to the needs.

V. Implementation and Testing

A. Burp Suite Professional

Burp Suite Professional is a paid tool by PortSwigger that was installed on Kali Linux. The Proxy settings were already configured to intercept and capture requests from the target web applications. The tool has a configured browser feature that eliminates manual adjustment for third-party browsers. Test credentials were used to log in to the target web application and test the three security levels, which are low, medium, and high. The login attempt was intercepted using the Burp Suite Professional tool to launch brute-force attacks. Parameters that can be brute forced are usernames and passwords. The captured request needs to be transferred to the Intruder section to set up attack positions and payloads. Under the Intruder section, highlight the parameters to attack and add payloads like password lists or external sources. Attack execution will display the result depending on the security level set. Burp Suite has automated scanning, customizable reporting features, and vulnerability classification. After getting the result, this project will provide recommendations for remediation.

B. Burp Suite Community

Burp Suite Community Edition is free and available from the PortSwigger website. After installing Kali Linux, the proxy settings must be manually configured before capturing the request for the web application. The Burp CA certificate was installed on the browser to facilitate the capture of secure traffic. After this setup, the tool is ready to capture and analyse requests.

During the login and request capture phase, an attempt to log in to the DVWA is intercepted by the Burp Suite tool when the intercept feature is on, which is a similar step to the Professional edition. However, the difference is that since this free version lacks automation, payloads need to be manually configured before the execution of the attack. Under payload configuration, extra manual steps are needed to test combinations of parameters.

Finally, a test was conducted on the various levels of DVWA security. The primary focus was to brute force the system to check the vulnerabilities of each level. Testing duration varied significantly with each higher level of security, which required more time and complexity.

C. SQL Map

To evaluate the SQL injection vulnerability, we need to run a basic command in Kali Linux for SQL Map:

Figure 3: SQL Map injection command for enumeration.

a) Database Enumeration Performance

If the detection of SQL Injection exists, the enumeration of database names will appear next. It will list all the available database names, which confirms the vulnerability.

b) Extraction of Table names from the database

Once the name of the database is available, the extraction of table names can be gathered with the following command:

Figure 4: SQL Command for extraction of tables from the database.

c) Retrieve information from a specific table

To retrieve data from a specific table, SQL Map will display data from the user table with the following command:

Figure 5: Command payload to retrieve data in SQL Map

As the retrieved information will be used for a brute force attack, the value for each user will be in hash values, which will be decrypted by the dictionary attack on that specific output. It will display the user and other details from the database table, and the passwords will be displayed in hash values. SQL Map will be able to crack the hash values during the testing phase.

d) Export the result and save the output file

To save the output file, we need to run the command in the terminal during the same session. The command for saving the output file:

Figure 6: SQL Map command to export and save result output file.

D. OWASP Zed Attack Proxy (ZAP)

Using OWASP ZAP's Fuzzer attack allows us to test input fields for vulnerabilities such as brute force, injection, and other malicious payload scenarios.

This assessment involved using the **Fuzzer tool** within **OWASP ZAP** to identify brute-force vulnerabilities on the **Damn Vulnerable Web Application (DVWA),** hosted on a private web server. DVWA's security levels were set to *low*, *medium*, and *high* to analyse how the application responds to brute-force attacks under different configurations.

OWASP was set up as a proxy for intercepts, and a test login attempt on DVWA's login page was performed to capture the HTTP POST request, which exposed the username and password parameters

E. Hydra Tool

The objective is to perform a brute-force attack on a web login form using Hydra with the http-get-form module, targeting the DVWA Brute Force vulnerability page. Hydra, often referred to as THC-Hydra, is a popular open-source tool used for password cracking through brute force or dictionary attacks. This is also needed a password list such as rockyou.txt wordlist.

Figure 7: Brute Force command given to assessed Hydra tool.

In command breakdown:

- -l admin sets the login username as admin.
- -p assigns the path to the password wordlist.
- Ip address identifies the target address.
- http-get-form module used to brute-force an http get form.
- "/DVWA/vulnerabilities/brute/index.php form parameter & failure condition.
- -s 80 identifies port number.

F. Wfuzz Tool

Like Hydra, Wfuzz is a versatile command line tool primarily used in web application penetration testing for brute forcing and discovering hidden files, such as directories and Get/Post parameter. To implementing for testing was easily installed on a Kali Linux environment.

VI. OVERALL RESULTS

A. Burp Suite Professional

After implementing Burp Suite and testing various levels of security, the low-security level was completed in just eleven seconds, medium-security required over thirty-three minutes, while high security required an hour and a half to successfully brute force.

B. Burp Suite Community Edition

The free edition revealed a notable difference in each level of security in the web application. For low-security configurations, tests were completed in approximately two hours, and medium-security tests required more time, finishing around three hours and thirty minutes. However, high-security attempts were unsuccessful after being tried multiple times. These results show the limitations of the Community edition in handling advanced security measures effectively.

a) Insights between Burp Suite Professional and Community Edition

Burp Suite Professional offers advanced automation, detailed reporting, and seamless integration with other tools, making it ideal for large-scale, professional penetration testing. Its pre-configured features streamline tasks like brute-force testing, enhancing efficiency and accuracy. In contrast, the Community Edition is more manual and time-consuming but serves as a valuable learning platform for beginners. While both versions are effective in identifying web vulnerabilities, the Professional edition provides greater scalability and actionable insights. Overall, the Professional version is best suited for experienced users and organizations, while the Community edition is a cost-effective option for basic testing and skill development

C. SQL Map

After running a successful command, SQL Map can retrieve information from the database. The username and password were retrieved where the password is in the hash value. Based on the dictionary attack, SQL Map can crack the hash value in low security mode.

D. ZAP Test

Before the full scan was completed, high -level results in 69ms RTT success attempts faster than low- security results however, it does not fully tell whether a password has been successfully found. The scan completion was long unfortunately the resulting output does not appear any starting or finished time.

During the test phase, a valid password was identified, evidenced by a 200 OK HTTP status code and redirection to the dashboard. This highlighted a lack of rate limiting and account lockout mechanisms in DVWA, making it vulnerable to brute force attacks.

a) Benefits of using ZAP tool

Zap is completely free and open source, making it accessible to individuals, small businesses and organisations without the need for expensive licenses. Community Driven: Regular updates and enhancements are contributed by an active community of security professionals, ensuring the tool remains current.

E. Hydra Test

The Hydra scan result was noted as the fastest compared to other brute force tools to complete; however, it requires multi-testing scans to compare the results, to ensure it is not a false positive. Benefits of using Hydra, does supports numerous protocols and services, making it versatile for penetration testers and security professionals and already installed on Kali Linux.

a) Weaknesses of the Hydra Tool

- Noisy and Detectable – this can be a drawback easily detectable by an intrusion detection/prevention system
- Rate Limiting – Many modern systems implement rate limiting or account lockout mechanisms; Hydra may not be able to perform effectively.
- Wordlist dependency – Effectiveness depends on the quality of the username/ password wordlists; it is unlikely to succeed against systems using strong, unique credentials.

F. Wfuzz tool

During testing, one of the main advantages of using Wfuzz is its high level of customization, however it lacks a graphical user interface, making it less accessible for beginners or those familiar with command line environments. Large brute -force operations can also be time consuming and resource – intensive. Although, these drawbacks, Wfuzz remains one of the useful tools for experience pen tester.

VII. SPECIFICATION OF TOOLS & EVALUATION

Each tool has its niche. WFuzz encountered a few crashes. Hydra excels in brute forcing, SQLmap in SQL injection detection, and ZAP in real-time interception. Selecting an appropriate tool based on the type of vulnerability is essential.

Overall Efficiency

The variability in scanning times highlights the importance of tool selection in security testing workflows, balancing both speed and the depth of scanning based on the resources and specific needs of a security assessment.

	Low-Level Security	**Mid-Level Security**	**High-Level Security**
HYDRA	**Successful attempt**	**Successful attempt**	**Successful attempt**
Complete Full Scan	02 seconds	03 seconds	03 seconds
BURP SUITE PRO	**Successful attempt**	**Successful attempt**	**Unsuccessful attempt**
Complete Full Scan	**11 seconds**	**33 minutes 35 seconds**	**01 hour 25 minutes**
BURP SUITE COMMUNITY	**Successful attempt**	**Successful attempt**	**Unsuccessful attempt**
Complete Full Scan	**02 hours 15 minutes**	**03 hours 33 minutes**	**01 hour 25 minutes 33 seconds**
SQL MAP TOOL	**Successful attempt**	**Successful attempt**	**Successful attempt**
Complete Full Scan	**05 minutes 05 seconds**	**08 minutes 15 seconds**	**12 minutes**
ZAP	**Successful attempt**	**Successful attempt**	**Successful attempt**
Complete Full Scan	**78ms RTT**	**65ms RTT**	**69ms RTT**
WFuzz	**Successful attempt**	**Successful attempt**	**Unsuccessful attempt**

Complete Full Scan	**14 minutes**	**20 minutes**	**20 minutes system crashed**

Note: ms in milliseconds and RTT in Round Trip Time

VIII. CONCLUSION

This penetration testing followed a methodical way to find out the vulnerability of web applications. This project aimed to test Damn Vulnerable Web Applications (DVWA) with the help of well-known penetration testing tools such as Burp Suite, SQL Map, WFuzz, Hydra and OWASP ZAP. Each tool was assessed based on multiple criteria based on different security levels, and the source code of each security level is carefully analysed for future reference. As this research project is properly documented, it is ensured that these testing procedures are followed according to the framework.

IX. REFERENCES

- A. Altulaihan, A. Alismail, and M. Frikha, “A Survey on Web Application Penetration Testing,” Electronics, vol. 12, no. 5, p. 1229, Mar. 2023
- Shebli and A. Beheshti, “Penetration testing process,” Cyber Defense Methods, vol. 10, pp. 71–80, 2018.
- Albahar, “Empirical comparisons of tool performance in penetration testing,” *Journal of Cybersecurity*, vol. 14, pp. 102-115, 2021.
- Kim and J. Lee, “Integration of modern penetration testing tools,” *Information Security Trends*, vol. 18, no. 2, pp. 200-215, 2023.
- Smith, “Controlled environments for testing vulnerabilities,” *Tech Insight Reports*, vol. 8, pp. 55-60, 2022.
- Khan and S. Bangash, “Standardized reporting in penetration testing,” *Cybersecurity Review*, vol. 12, pp. 88-100, 2020.
- OWASP Foundation, “OWASP ZAP documentation and features,” 2023. [Online]. Available: https://owasp.org.

- Nagpure and S. Kurkure, “Vulnerability Assessment and Penetration Testing of Web Application,” in *2017 International Conference on Computing, Communication, Control and Automation
- OWASP Top Ten | OWASP Foundation.” [Online]. Available: https://owasp.org/www-project-top-ten/

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.