SSRG International Journal of Computer Science and Engineering
ISSN: 2348–8387 / https://doi.org/10.14445/23488387/IJCSE-V11I12P101

Volume 11 Issue 12, 1-6, December 2024

Original Article

A Comprehensive NIDS-Based Strategy for Web Application Penetration Testing

Srujana Manjunath[1], Shreya Malshetty[2], D. Jayalakshmi[3], Chaithra Banger[4], Y. Sharmasth Vali[5]

[1,2,3,4]*Computer Science, Cyber Security, Presidency University, Karnataka, India.*
[5]*Presidency University, Karnataka, India.*

Corresponding Author : srujana.20211ccs0116@presidencyuniversity.in

Received: 10 October 2024 Revised: 15 November 2024 Accepted: 09 December 2024 Published: 29 December 2024

Abstract - *Imagine receiving an email from cybercriminals stating that all your personal information has been compromised—name, date of birth, home address, and finances. They are demanding money from you in exchange for not leaking your sensitive information. It's a terrifying situation to be in, isn't it? We adopt an NIDS-based approach for web application penetration testing in order to resolve this issue. Web application penetration testing is an ongoing security evaluation that mimics actual attacks to evaluate how secure web applications are. The main objective is to find any possible flaws, configuration errors, or vulnerabilities that malicious users can use to jeopardize a web application's availability, confidentiality, or integrity. An NIDS is deployed to detect fraudulent activities at the network level, which can complement conventional penetration testing techniques that concentrate on flaws in the software. The goal of this research is to improve the identification of security vulnerabilities at the network and application levels by combining traditional web application penetration testing with Network Intrusion Detection Systems (NIDS).*

Keywords - *Intrusion Detection, Network Intrusion Detection, Penetration Testing, Vulnerability Assessment,Web Application Security.*

1. Introduction

In today's internet era, a network is essential to every company that conducts virtual operations. Nevertheless, networks' interconnectedness also makes them susceptible to cyberattacks. Using network penetration testing is one method of guaranteeing network security. It includes mimicking actual network assaults to identify and address any weaknesses. Cybercriminals find web applications (such as social networking, cloud storage, e-commerce, and financial transactions) to be appealing because of their growing capability and complexity. Web application flaws can result in serious security lapses, data theft, and harm to one's reputation. However, regular network testing is very important to keep the network safe from constantly evolving cyber threats. As a result, it is crucial to make sure that these applications are secure, and penetration testing has emerged as a key tool for locating and fixing possible flaws before malicious users can take advantage of them.

Pen testing or penetration testing in networks mimics the attacks carried out by actual hackers or malicious users. This aids in identifying risks and vulnerabilities in the network's security measures. Wireless connections are intricate and compromise a mix of WAN, LAN, and Wi-Fi. Countless endpoints and devices, including PCs, servers, and the Internet of Things (IOT), are incorporated into the network as well. Additionally, they encompass Intrusion Prevention Systems (IPS) and firewalls. Few of these, or in most cases, all of these components might have a flaw that hackers could benefit from to gain access. The goal of network pen testing is to ensure the security of the network system and identify the loopholes before any attack arises. Therefore, a holistic approach to securing the web application ought to examine the network system alongside testing the software tool itself.

Blending Network Intrusion Detection Systems (NIDS) into penetration testing or pen testing techniques is an appealing approach to tackle this challenge. NIDS are tools that keep an eye on network movements and identify questionable activities. Based on the volume of traffic flowing through all of the network devices, multiple NIDS will be required. If any unusual behavior is discovered, the NIDS will notify the system administrators or the IT team to examine. By offering a more thorough, multi-layered study of web application security, this method seeks to uncover possible attack routes that could otherwise go unnoticed.

This research involves a distinctive technique that melds NIDS strategy with web application pen testing to achieve and discover the loopholes at network system levels and

application levels. By integrating NIDS into this technique, ethical hackers or security experts can effectively understand how assaults can replicate across the network. In this study, we analyze how to use the NIDS strategy in web application penetration testing utilizing standard tools like Nmap, OWASP ZAP, and Nessus. We assess the efficacy of this infused technique by testing a prototype web application, monitoring network traffic, evaluate NIDS alarms. The conclusions drawn reveal how to enhance web applications' security performance by disclosing loopholes of the attacks or vulnerabilities.

2. Methodologies

2.1. Techniques for Penetration Testing with OWSAP ZAP, Nmap, Nessus:

2.1.1. The Zed Attack Proxy (OWASP ZAP)

A dynamic application security testing (DAST) tool called OWASP ZAP is intended to identify weaknesses in online applications. It may be used to perform vulnerability assessments and penetration tests both manually and automatically.

- Reconnaissance & Spidering: The target web application may be automatically crawled using the ZAP Spider tool. For additional testing, ZAP generates a thorough map of the target web application's surface area.
- Active Scanning: ZAP's Active Scanner checks the web application for replies that point to possible vulnerabilities by sending different payloads. These tests involve looking for typical web application vulnerabilities like as SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF).
- Passive Scanning: During the reconnaissance stage, ZAP also uses passive scanning to find possible problems like incorrect HTTP headers, cookie settings, and poor SSL/TLS setups without having to engage with the application actively.
- Automated Reports: Following testing, provide thorough reports that include all vulnerabilities found, their risk levels, and possible fixes that can be applied to repair.

ZAP offers a thorough overview of the attack surface and weaknesses. Unusual traffic patterns or payloads that mimic attack vectors are examples of how ZAP warnings may be tracked by a Network Intrusion Detection System (NIDS) when it is integrated with one.

2.1.2. Nmap

A strong tool for network discovery and vulnerability analysis is Nmap (Network Mapper). It is used to find open ports, services, and any security flaws in the network of interest.

- Network Discovery: Find active hosts inside the target architecture to carry out network discovery. To find devices inside the desired range, use Nmap's ICMP ping sweep (`nmap -sn`).
- Port Scanning: Perform a thorough port scan (`nmap -sS`) to find all open ports on the target web application server. This will help you identify services that are using common ports and other service ports that could be possible points of attack.
- Service Detection: Nmap's service detection feature (`nmap -sV`) may be used to determine which service versions are operating on open ports. This can assist in locating known vulnerabilities linked to software versions.
- Vulnerability Scanning: To find known vulnerabilities in services, use Nmap scripts for vulnerability scanning (`nmap --script=vuln`). These scripts may automate vulnerability tests against the target system and are a component of Nmap's NSE (Nmap Scripting Engine).

The NIDS may be set up to watch for certain attack patterns that make use of flaws discovered during the scanning phase using the data that Nmap collects.

2.1.3. Nessus

A thorough vulnerability scanner, Nessus, is especially helpful for evaluating network security, finding flaws, and offering practical remedial advice. It may check for known security holes, configuration errors, and compliance problems in both systems and online apps.

- Initial Network Scan: To find hosts and evaluate the infrastructure's overall security posture, do a first network scan. Like Nmap, but with a focus on security, Nessus will find open ports and services.
- Vulnerability Assessment: Conduct thorough vulnerability scans using Nessus on the web application as well as network infrastructure, such as firewalls, routers, and switches. Nessus scans systems and apps for more than 130,000 known vulnerabilities and configuration flaws.
- Web Application Scanning: Find vulnerabilities in web applications such as SQL injection, cross-site scripting, unsafe HTTP methods, and more by using Nessus. Critical vulnerabilities like out-of-date plugins or servers without fixes can be detected by it.
- Report Generation: Following the scan, Nessus produces comprehensive reports detailing the vulnerabilities found, along with information on their severity, risk assessment, and suggested corrective actions.

Nessus discoveries may be loaded into the NIDS to create custom signatures and monitoring rules for vulnerabilities found. It is possible to set up the NIDS to indicate any efforts at exploiting a vulnerability that Nessus finds, such as an Apache server that is not patched.

2.2. Integration of OWASP ZAP, Nmap, and Nessus with Network Intrusion Detection Systems (NIDS)

2.2.1. Pre-Test Configuration

Initially, we used Nmap to discover live hosts, open ports, and services on the target network. This information feeds into ZAP and Nessus to target specific attack vectors during the testing phase.

2.2.2. Penetration Testing

The results obtained are fed to OWASP ZAP for active web application testing and vulnerability scanning. At the same time, Nessus provides an in-depth vulnerability assessment of the target network and application.

2.2.3. NIDS Monitoring

Implement a comprehensive monitoring system to analyze traffic patterns for indicators of potential exploitation attempts. This involves establishing baseline traffic behaviors and continuously observing network activity for anomalies that may suggest malicious behaviors, such as unusual spikes in traffic, unexpected access to sensitive resources, or patterns typically associated with known attack vectors. Employing advanced analytics and automated alerts will enhance the ability to quickly identify and respond to potential threats, ensuring a proactive defense against intrusion.

2.2.4. Real-Time Detection

As penetration testing progresses, the Network Intrusion Detection System (NIDS) continuously monitors network traffic for unauthorized access attempts or exploitation activities. It utilizes vulnerability signatures and attack vectors that have been identified through tools such as ZAP, Nmap, and Nessus.

By analyzing this data in real-time, the NIDS can promptly detect suspicious behavior, allowing for immediate response and mitigation efforts to protect the system from potential threats. This proactive approach ensures a robust defense against various attack strategies throughout the testing phase.

2.2.5. Post-Test Analysis and Reporting

After completing the penetration testing phase, the NIDS provides logs of any detected exploits, while Nmap and Nessus provide comprehensive assessments and remediation steps for the discovered weaknesses.

3. Implementation

Here, we outline the detailed steps for putting our all-inclusive Network Intrusion Detection System (NIDS)-based approach to web application penetration testing into practice. These tools—NMAP, NESSUS, and OWSAP ZAP—were chosen because they can perform a variety of security testing tasks on web applications, including vulnerability scanning, network traffic analysis, and real-time monitoring.

3.1. Nessus Vulnerability Report

Nessus has found numerous vulnerabilities, including a major buffer overflow vulnerability that requires urgent attention.

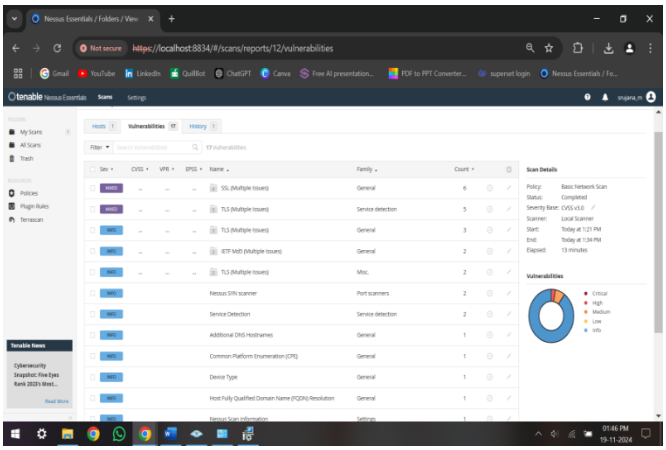


Fig. 1 Presents the Nessus vulnerability assessment report, highlighting critical findings

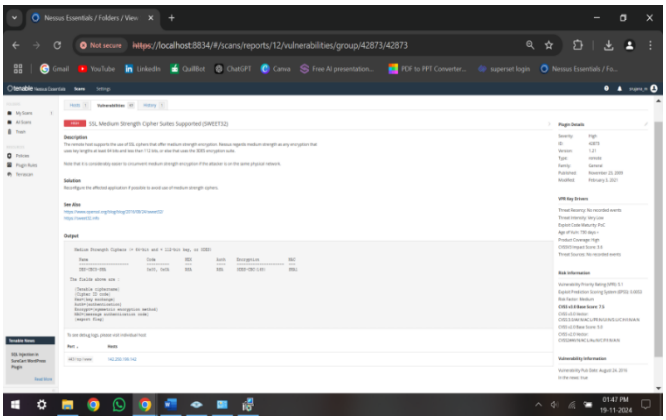


Fig. 2 Illustrates description of critical finding of a target host

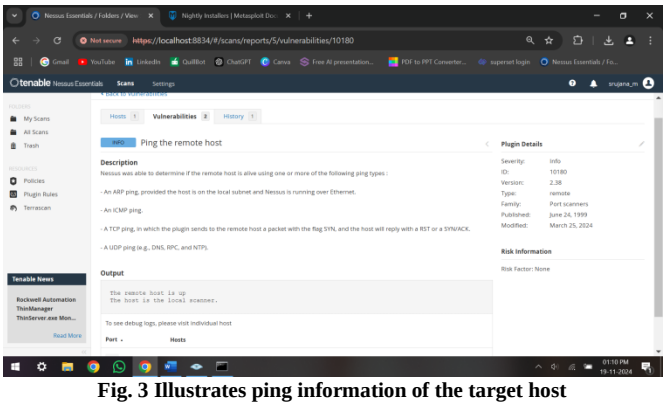


Fig. 3 Illustrates ping information of the target host

3.2. Nmap Scanning Result

The first target web application was reconnaissance using Nmap. It provides vital information to further exploit vulnerabilities by assisting in the identification of open ports and services.

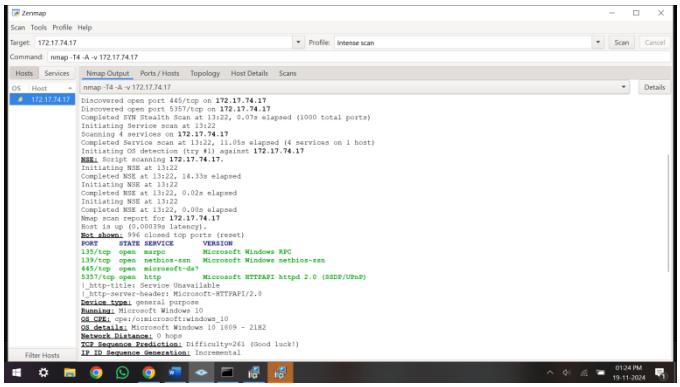


Fig. 4 Represents target host (172.17.74.17) description

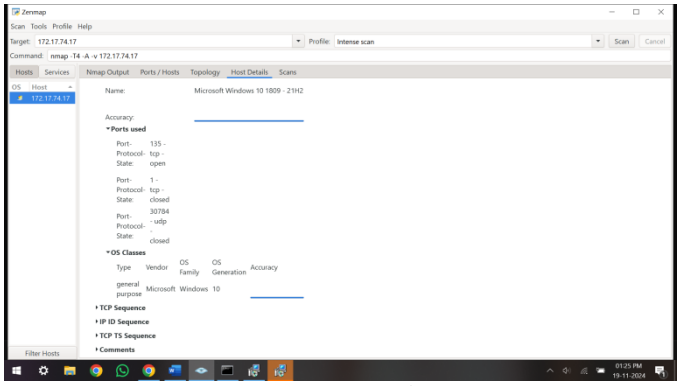


Fig. 5 Illustrates target host details after intense scanning

3.3. OWASP ZAP (Zed Attack Proxy)

Employed for active and passive security testing. The automated spider and passive scanning functionalities were used in the penetration testing to find weaknesses.

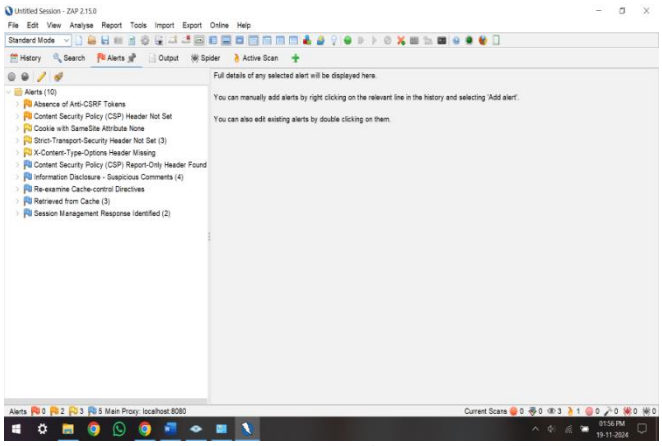
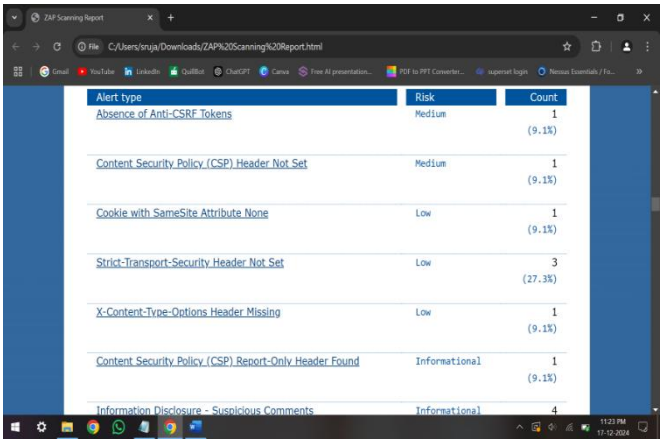


Fig. 6 Represents possible alerts of given remote target host.

By applying this comprehensive NIDS-based strategy, the testing effectively simulated real-world attack scenarios on the web application. Through the use of tools like Nmap, Nessus, and OWASP ZAP, we identified a diverse range of vulnerabilities, from network traffic anomalies to significant application flaws. The insights gained from each tool highlighted different aspects of the application's security posture, allowing us to recommend improvements and develop targeted strategies to mitigate risks constructively.



Alert type	Risk	Count
Absence of Anti-CSRF Tokens	Medium	1 (9.1%)
Content Security Policy (CSP) Header Not Set	Medium	1 (9.1%)
Cookie with SameSite Attribute None	Low	1 (9.1%)
Strict-Transport-Security Header Not Set	Low	3 (27.3%)
X-Content-Type-Options Header Missing	Low	1 (9.1%)
Content Security Policy (CSP) Report-Only Header Found	Informational	1 (9.1%)
Information Disclosure - Suspicious Comments	Informational	4

Fig. 7 Shows the number of alerts of each alert type, with the alert type's risk level

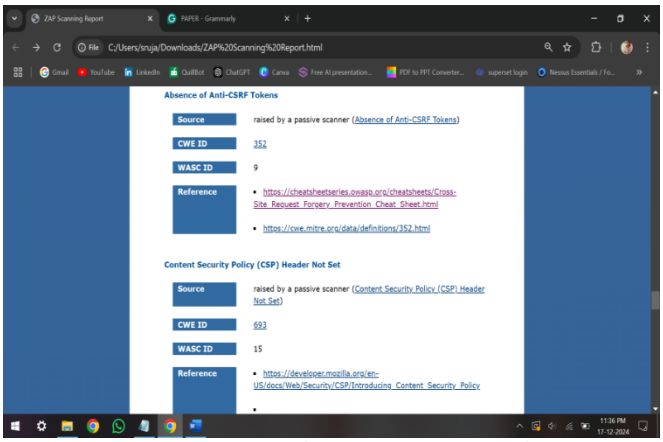


Fig. 8 Represents additional information on the types of alerts in the ZAP report

4. Discussions

The installation of Metasploit for this project presented a number of difficulties, which may prevent it from integrating well in some situations. One major problem was the installation process failing, which was partially caused by antivirus software interference. The system identified some Metasploit components as possible risks, which led to unsuccessful installation attempts or unfinished setups. Furthermore, in settings with strict security procedures, the Metasploit framework's dependence on particular system configurations and security permissions may lead to further issues. Because of this, even when the program is compatible with the operating system, users may still have issues.

It is advised to either temporarily disable antivirus software or set it up to permit Metasploit's components to operate without interruption in order to lessen these problems in subsequent installations. Disabling antivirus software,

however, may put the system in danger; thus, it could be wiser to employ different strategies like using Metasploit on a virtual machine or containerized environment. In addition to separating Metasploit from the host system, this method offers a safer and more regulated setting for resolving installation problems without jeopardizing system security. Furthermore, utilizing cloud-based technologies or creating specialized testing environments can make future Metasploit endeavours more efficient.

6. Conclusion

When performing web application penetration testing, the NIDS strategy provides a more holistic way to locate and identify security flaws. This literature survey has covered popular web application threats, the growth of penetration testing processes, and the role of NIDS in enhancing safety checks. The implementation of NIDS with penetration testing provides robust protection over contemporary threats.

Acknowledgments

I would like to express my sincere gratitude to all those who have supported me throughout this research. First and foremost, I would like to thank my supervisor, Dr. Sharmasth Vali Y, sir, for their invaluable guidance, patience, and expertise. Their continuous encouragement and constructive feedback were crucial to the development of this work.

I would also like to extend my thanks to my co-authors, Shreya Malshetty, Jayalakshmi D, and Chaithra Banger, for their insightful discussions and contributions. Special thanks to my university library for their assistance with plagiarism and for their help in reviewing and improving the manuscript.

Lastly, I would like to thank my family and friends for their unwavering support and understanding throughout this project.

References

[1] Mariam Alhamed, and M.M. Hafizur Rahman, "A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions," *Applied Sciences*, vol. 13, no. 12, pp. 1-24, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Esra Abdullatif Altulaihan, Abrar Alismail, and Mounir Frikha, "A Survey on Web Application Penetration Testing," *Electronics*, vol. 12, no. 5, pp. 1-23, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Vladimir Ciric et al., "Modular Deep Learning-Based Network Intrusion Detection Architecture for Real-World Cyber-Attack Simulation," *Simulation Modelling Practice and Theory*, vol. 133, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Ferry Astika Saputra et al., "The Next-Generation NIDS Platform: Cloud-Based Snort NIDS Using Containers and Big Data," *Big Data and Cognitive Computing*, vol. 6, no. 1, pp. 1-15, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Satish Kumar, Sunanda Gupta, and Sakshi Arora, "Research Trends in Network-Based Intrusion Detection Systems: A Review," *IEEE Access*, vol. 9, pp. 157761-157779, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Heather Lawrence et al., "CUPID: A Labelled Dataset with Pentesting for Evaluation of Network Intrusion Detection," *Journal of Systems Architecture*, vol. 129, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Lirim Ashiku, and Cihan Dagli, "Network Intrusion Detection System using Deep Learning," *Procedia Computer Science*, vol. 185, pp. 239-247, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Patrick Vanin et al., "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning," *Applied Sciences*, vol. 12, no. 22, pp. 1-27, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Vinod Varma Vegesna, "Utilising VAPT Technologies (Vulnerability Assessment & Penetration Testing) as a Method for Actively Preventing Cyberattacks," *International Journal of Management, Technology and Engineering*, vol. 12, vol. 7, pp. 81-94, 2022. [Google Scholar] [Publisher Link]

[10] Ke Chen et al., "Research on the Application of Penetration Testing Frameworks in Blockchain Security," *Computational and Experimental Simulations in Engineering*, pp. 307-330, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Fredrik Heiding et al., "Penetration Testing of Connected Households," *Computers & Security*, vol. 126, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Arvind Goutam, and Vijay Tiwari, "Vulnerability Assessment and Penetration Testing to Enhance the Security of Web Application," *4th International Conference on Information Systems and Computer Networks*, Mathura, India, pp. 601-605, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Prashant Vats, Manju Mandot, and Anjana Gosain, "A Comprehensive Literature Review of Penetration Testing & Its Applications," *8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, Noida, India, pp. 674-680, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Khaled Abdulghaffar, Nebrase Elmrabit, and Mehdi Yousefi, "Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners," *Computers*, vol. 12, no. 11, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Soroush M. Sohi, Jean-Pierre Seifert, and Fatemeh Ganji, "RNNIDS: Enhancing Network Intrusion Detection Systems through Deep Learning," *Computers & Security*, vol. 102, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Ms. Khushnaseeb Roshan, and Aasim Zafar, "Boosting Robustness of Network Intrusion Detection Systems: A Novel Two-Phase Defense Strategy Against Untargeted White-Box Optimization Adversarial Attack," *Expert Systems with Applications*, vol. 249, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Murat Aydos et al., "Security Testing of Web Applications: A Systematic Mapping of the Literature," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6775-6792, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Jaydeep R.Tadhani et al., "Securing Web Applications Against XSS and Sqli Attacks Using a Novel Deep Learning Approach," *Scientific Reports*, vol. 14, pp. 1-17, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Amel F. Aljebry, Yasmine M. Alqahtani, and Norrozila Sulaiman, "Analyzing Security Testing Tools for Web Applications," *International Conference on Innovative Computing and Communications*, vol. 1387, pp. 411-419, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Branislav Rajić, Žarko Stanisavljević, and Pavle Vuletić, "Early Web Application Attack Detection Using Network Traffic Analysis," *International Journal of Information Security*, vol. 22, pp. 77-91, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] Maruthi Rohit Ayyagari et al., "Intrusion Detection Techniques in a Network Environment: A Systematic Review," *Wireless Networks*, vol. 27, pp. 1269-1285, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[22] Muhammad Ali et al., "Effective Network Intrusion Detection Using a Stacking-Based Ensemble Approach," *International Journal of Information Security*, vol. 22, pp. 1781-1798, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Ankur Chowdhary, Kritshekhar Jha, and Ming Zhao, "Generative Adversarial Network (GAN)-Based Autonomous Penetration Testing for Web Applications," *Sensors*, vol. 23, no. 18, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]