

Journal Scientific and Applied Research, Vol. 31, 2026
International Journal

ISSN 1314-6289 (Print)
ISSN 2815-4622 (Online)

Original Contribution

IMPLEMENTATION OF VULNERABILITY WEB SCANNERS IN THE COMPUTER NETWORKS

Petar K. Boyanov

Department of Communication and Computer Engineering and Security Technologies, Faculty of Technical Sciences, Konstantin Preslavsky University of Shumen, Shumen 9712, 115 Universitetska St., e-mail: petar.boyanov@shu.bg

Abstract: *In this scientific article implementation of vulnerability web scanners in the computer networks is presented. The article focuses on web vulnerability scanning and how it can find weaknesses in locally tested web servers. Two scans on the web servers running on the victim and attacking computers were performed.*

Key words: *Apache, PHP, IPv4, IPv6, Linux, Nikto, Ports, Python3, Scanning, Security Policy, Service, Vulnerability, Web server, Windows.*

1. Introduction

Companies, schools, and other organizations use networks to exchange information and provide services. Websites and web applications are also used for many everyday activities [1,3,12,15,19]. The security problems in these systems can put important data at risk. Attackers can use these problems to access systems or steal data. This makes network and web security important. The vulnerability web scanners are able to check the security of these systems. These tools scan web servers and applications for possible weaknesses. They can check server settings, open ports, software versions, and other security problems. One popular tool for this purpose is Nikto [4,5,10,11,13,16]. It is an open-source tool used to scan web servers and find possible security issues. Another useful tool is OWASP ZAP, which can help find problems in web applications. Each web scanner uses its own methods and can detect different security problems. [8,9]. System administrators can use these tools to check their network security. They can find security problems before malicious cyber-attackers do. [13,14]. This scientific research examines web vulnerability scanning. The research focuses mainly on using Nikto in a controlled computer network. [14,17,18]. A web server is tested to find possible security problems. The achieved scan results are examined in order to identify problems and possible ways to improve the tested server security. This research shows how web vulnerability scanners can find security problems in computer networks [1,2,3,4,5,6,12].

The content of this academic article is intended for research, instructional applications and educational purposes only. Any unauthorized or unethical use of the presented material falls outside the author's scope of responsibility.

2. Related work

Several studies have examined web security and vulnerability scanning tools. Agrawal et al. [1] presented a framework for testing web application security while Annaufal et al. [2] used black-box testing to find website problems. Doke et al. [3] studied automatic tools for collecting system information. Fediushyn et al. [4] developed an automated platform for vulnerability scanning, and Goyal et al. [5] studied the use of artificial intelligence in penetration testing. Hansel et al. [6] worked on improving Nikto, while Hussain et al. [7] discussed better methods for web security testing. Karangle et al. [8] compared Nikto and Uniscan, and Kondraciuk et al. [9] tested four different web security tools. Lavhate et al. [10] studied automatic asset discovery, while Lukanta et al. [11] focused on session security problems. Luthfi and Irsan [12] used Nikto and OWASP ZAP to test a website and Naing [13] discussed improvements in web application security. Mulcahy et al. [14] presented a framework for hardware security, while Nrk [15] compared Nessus, Acunetix, and Nikto. Rahman et al. [16] studied vulnerabilities in e-commerce websites, and Riskhan et al. [17] discussed vulnerability testing and protection using artificial intelligence. Sllame et al. [18] examined several security scanning tools while Varma et al. [19] presented ReconMaster, a tool for web reconnaissance. These studies are related to the use of web vulnerability scanners in computer networks.

3. Problem description and results

The experiments in this study in a controlled personal lab test environment were carried out. The used operating system is Kali linux x64 with the following system information: Linux Petar 7.1.5+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 7.1.5-1kali1 (2026-07-29) x86_64 GNU/Linux.

The IPv4 address of the attacking host in the personal controlled laboratory environment is 192.168.1.18/24 (fig. 2) and the IPv4 address of the victim host is 192.168.1.19/24 (fig. 1). The attacking host uses the Kali Linux operating system and the victim host uses Microsoft Windows 10. The local test network also contains several other computers. Several servers are installed on the victim host which are respectively:

- Apache server with ports 80 and 443;
- MySQL server with port 3306;
- FileZilla server with ports 21 and 14147;
- Mercury server with ports 25, 79, 105, 106, 110, 143 and 2224;
- TomCat server with ports 8005, 8080, 49674 and 49675.

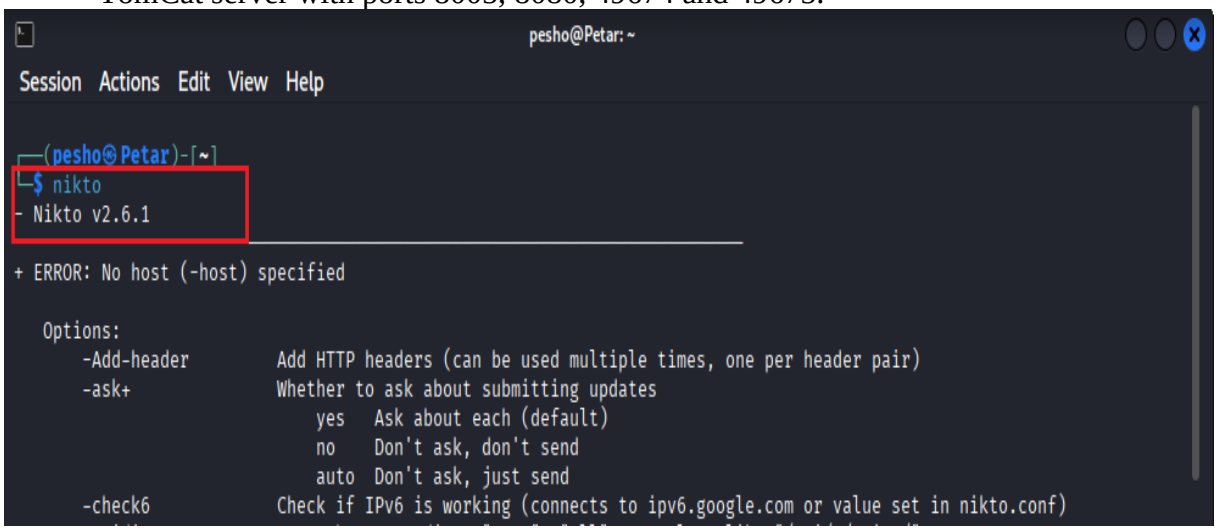


Fig. 1. The version of the web scanner

Fig. 1 shows the version of the program Nikto which is used during the security reteach test. The vulnerability scan with Nikto version 2.6.1 was performed. Nikto is a free tool that scans web servers for security problems and incorrect server parameters [1,4,14,15,19]. It checks HTTP methods, server information, security headers, and accessible directories. The number of the version is important because there are different versions of this tool and each one may contain different tests and security checks [12,13,18]. This information shows which version of Nikto was used in the scientific experiment.

The line "[000287]" shows that the header contains the value PHP/8.2.12 which reveals that the website uses PHP version 8.2.12 [1,4,5,7,9,11]. This information helps identify the technologies used by the web application. Showing the software versions in HTTP headers can provide to malicious attackers useful information about the server. The attackers can use this information to look for known vulnerabilities in the PHP version [13,14,15,16]. The network administrators and security specialists should remove unnecessary technology information from HTTP response headers (fig. 2).

The CGI result shows that the scanner did not find any CGI directories. CGI is a technology that allows the web servers to execute external scripts or programs [8,11,13]. The older websites may use CGI scripts to process user requests and run server-side operations. The scripts can cause serious security problems when they are incorrectly configured or contain vulnerable code [15,16]. This result does not prove that CGI is completely disabled on the tested web server. The web scanner may not have found all CGI directories (fig. 2).

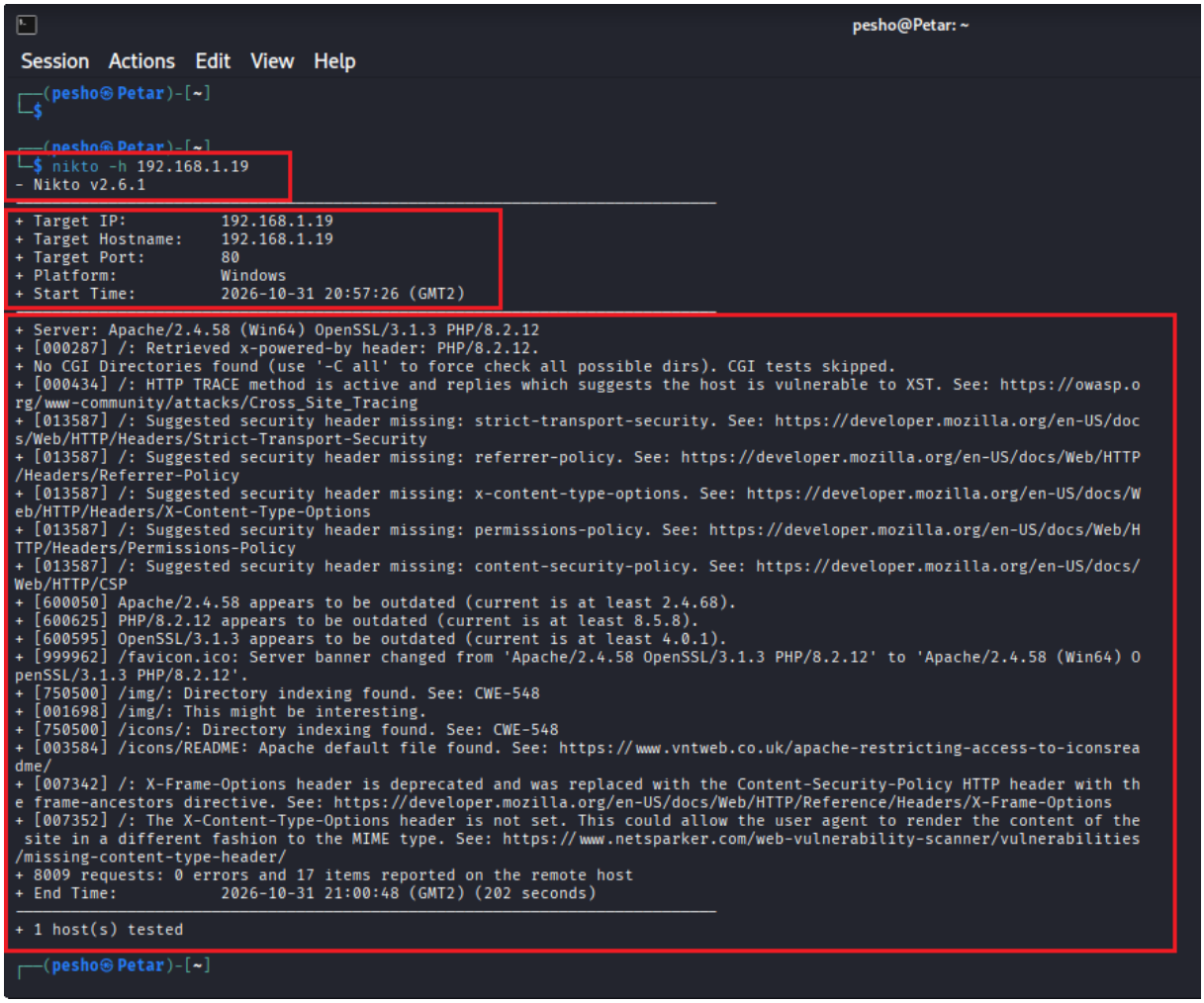


Fig. 2. The execution of the command "nikto -h 192.168.1.19"

The line “[000434]” shows that the HTTP TRACE method is enabled on the server. This method allows the server to return information about a received request [4,5,6]. I It can be used to test and troubleshoot communication problems. The HTTP TRACE method is enabled in this test environment. This may create security risks related to Cross-Site Tracing (XST) attacks. This attack may allow attackers to obtain information from HTTP requests when other security weaknesses are present [2,6,9,12,15,19]. The web administrator should check the server settings and disable the TRACE method (fig. 2).

The line “[013587]” shows that the Strict-Transport-Security security header from the HTTP response is missing [6,7,8,16,17]. This header is known as HSTS. It tells web browsers to access the website only through HTTPS connection. Network administrators and security specialists should first configure HTTPS and then enable the header for strict transport security.

The line “[013587]” shows that the tested website does not have the header for referrer policy [1,4,7,10,12]. Web administrators should configure this policy to limit the sharing of unnecessary information (Figs. 2 and 4).

The line “[013587]” shows that the server response does not contain the header of “x content type options”. The administrator should add the “x content type options” with the value “nosniff” to improve browser security (fig. 2 and 4).

The line “[013587]” shows that the “permissions policy” header is not configured on the tested web server [17,18,19]. The line “[013587]” that the “content security policy” header is missing from the tested website [2,3,4]. The web administrators should configure a policy that allows the resources needed by the website (fig. 2 and 4).

The following lines “[600050]”,”[600625]” and “[600595]” show that the installed versions of Apache, PHP, and OpenSSL servers are outdated. Web administrators should update these programs to newer versions.

The line “[750500]” shows that directory indexing in the /img/ directory is enabled. A directory listing can help attackers understand the website structure and find interesting files [5,8,10,11]. This security issue is associated with CWE-548 vulnerability. The web administrator should disable directory indexing.

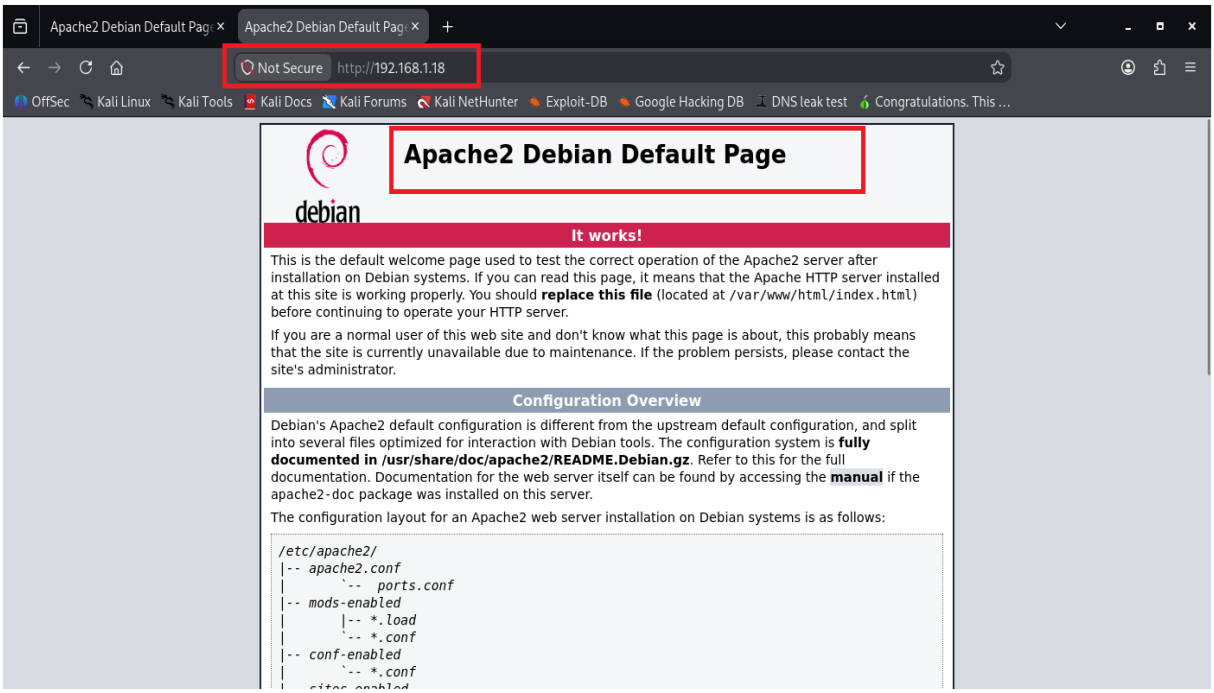


Fig. 3. The default Apache web server page in Debian systems

The line “[003584]” shows that the scanner found a README file in the icons directory [1,2,3,6,7,12]. Web administrators should check the file and remove it or block access to it if the website does not need it (Fig. 2).

The line “[007342]” describes how attackers may trick users into clicking buttons or performing actions without knowing what they are doing. This type of attack is called clickjacking [2,5,8,9,11,13,14,19]. Web administrators should configure a suitable “x-frame options” header or a “content security policy” (Figs. 2 and 4).

The line “[007352]” shows that the tested web server does not send the “x-content type options” header in its HTTP response [1,5,7,9,12]. The web administrators and security specialists should set the header value to “nosniff” and check that the server sends it correctly (fig. 2 and 4).

The second test involved starting the Apache web server on the attacking computer (fig. 3). The attacking host has got a IPv4 address 192.168.1.18. The web server was then scanned with Nikto (fig. 4). The purpose of this test was to find vulnerabilities in the web server. The final test was performed on the victim's web server using the SSL option (Fig. 5).



Fig. 4. The execution of the command “nikto -h 192.168.1.18”

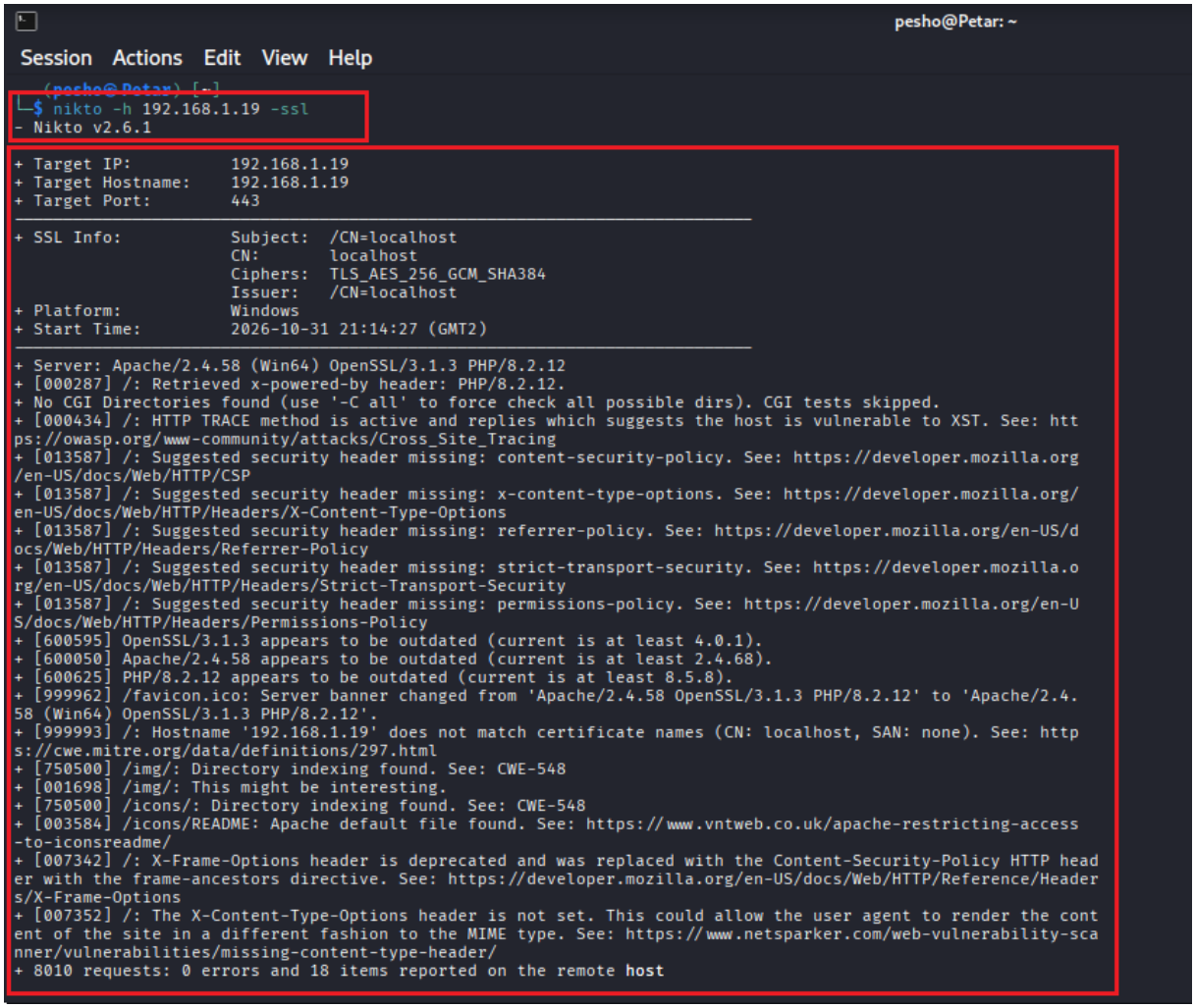


Fig. 5. The execution of the command "nikto -h 192.168.1.19 -ssl"

5. Conclusion

The results show that web administrators should improve server security. They should disable unnecessary HTTP methods, remove directory indexing, restrict access to default files, configure missing security headers, and update outdated software. These changes can reduce security risks and help protect the web application from common cyber-attacks.

References

[1] Agrawal, A. M. K., Bolli, D. B., Teja, C., Budhwani, T. P., Sehgal, L., Dharia, J. N., & Aggarwal, A. (2021). Offensive Web Application Security Framework. Design Engineering (London), ISSN: 0011-9342, Year 2021, Issue: 9, pp. 17334 – 17342.

[2] Annaufal, A. N., Ardhani, M. N., Bintang, S., Simangunsong, G. A., Faylasuf, H. F., Wicaksana, A. D., ... & Mindara, G. P. (2025). Analisis Kerentanan pada Situs Buatinkamu. id dengan Pengujian Penetrasi Metode Black Box. Explore, 15(1), 11-17., p-ISSN : 2087-894X, e-ISSN : 2656-615X.

[3] Doke, A., Shripnnavar, V., Baviskar, D., Naik, V., & Kumavat, K. S. (2023, September). Survey of Auto Recon SQL Mapping, 2023 6th International Conference

on Contemporary Computing and Informatics (IC3I), Gautam Buddha Nagar, India, 2023, pp. 710-714, doi: 10.1109/IC3I59117.2023.10397857.

[4] Fediushyn, O., Khivrenko, H., Popova, N., & Stadnik, V. (2026). Design and implementation of an integrated automated vulnerability scanning platform for web application security assessment, https://doi.org/10.20998/3083-6298.2026.02.11.

[5] Goyal, D., Subramanian, S., Peela, A., & Shetty, N. P. (2024). Hacking, the lazy way: LLM augmented pentesting. arXiv preprint arXiv:2409.09493, https://doi.org/10.48550/arXiv.2409.09493.

[6] Hansel, S., Tanuwijaya, K., Laksono, T. J., & Kurniawan, A. (2026, April). Enhanced Nikto: Integrating Application Layer Vulnerability Detection into Web Vulnerability Scanner, 2026 International Conference on Electrical/Electronics, Robotics, Artificial Intelligence, and Informatics (ICERAI), Ras Al Khaimah, United Arab Emirates, 2026, pp. 1-6, doi: 10.1109/ICERAI69511.2026.11494579.

[7] Hussain, K., Sheikh, M. A. U., Naing, T., Hussain, M., & Amin, N. U. (2025). Reshaping Cybersecurity Practices by Optimizing Web Application Penetration Testing. In Reshaping CyberSecurity With Generative AI Techniques (pp. 395-410). IGI Global Scientific Publishing, DOI: 10.4018/979-8-3693-5415-5.ch011.

[8] Karangle, N., Mishra, A. K., & Khan, D. A. (2019, July). Comparison of Nikto and Uniscan for measuring URL vulnerability, 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944463.

[9] Kondraciuk, A., Bartos, A., & Pańczyk, B. (2022). Comparative analysis of the effectiveness of OWASP ZAP, Burp Suite, Nikto and Skipfish in testing the security of web applications. Journal of Computer Sciences Institute, 24, 176-180. https://doi.org/10.35784/jcsi.2929.

[10] Lavhate, S., Shaikh, S., Bangal, S., Borawake, V., Shinde, K., & Patil, K. (2026, April). Optimizing Bug Bounty Recon Workflows using Automated Asset Discovery and Vulnerability Correlation, 2026 9th International Conference on Inventive Computation Technologies (ICICT), Kirtipur, Nepal, 2026, pp. 1050-1056, doi: 10.1109/ICICT68280.2026.11511063.

[11] Lukanta, R., Asnar, Y., & Kistijantoro, A. I. (2014, November). A vulnerability scanning tool for session management vulnerabilities, 2014 International Conference on Data and Software Engineering (ICODSE), Bandung, Indonesia, 2014, pp. 1-6, doi: 10.1109/ICODSE.2014.7062682.

[12] Luthfi, A., & Irsan, M. (2025, October). Identification of Security Vulnerabilities of XYZ Website using Open Web Application Security Project Zed Attack Proxy and Nikto," 2025 IEEE 9th International Conference on Software Engineering & Computer Systems (ICSECS), Pekan, Pahang, Malaysia, 2025, pp. 458-462, doi: 10.1109/ICSECS65227.2025.11279205.

[13] Naing, T. (2024). Reshaping Cybersecurity Practices by Optimizing Web Application. Reshaping CyberSecurity With Generative AI Techniques, 395, DOI: 10.4018/979-8-3693-5415-5.

[14] Mulcahy, P., Nowatkowski, M., Morris, J., & Millan, J. G. (2025, March). Developing a Framework for Hardware Security, SoutheastCon 2025, Concord, NC, USA, 2025, pp. 128-135, doi: 10.1109/SoutheastCon56624.2025.10971450.

[15] Nrk, S. (2024). A comparative analysis of vulnerability management tools: Evaluating nessus, acunetix, and nikto for risk based security solutions. arXiv preprint arXiv:2411.19123, https://doi.org/10.48550/arXiv.2411.19123.

[16] Rahman, M. A., Amjad, M., Ahmed, B., & Siddik, M. S. (2020, January). Analyzing web application vulnerabilities: an empirical study on e-commerce sector in Bangladesh. ICCA '20: Proceedings of the International Conference on Computing Advancements, Article No.: 66, Pages 1 – 6, https://doi.org/10.1145/3377049.3377107.

[17] Riskhan, B., Sheikh, M. A. U., Hossain, M. S., Hussain, K., & Hussain, M. (2025). A Methodical Approach to Exploiting Vulnerabilities and Countermeasures Using AI. In Utilizing Generative AI for Cyber Defense Strategies (pp. 287-308). IGI Global Scientific Publishing, DOI: 10.4018/979-8-3693-8944-7.ch008.

[18] Sllame, A. M., Tomia, T. E., & Rahuma, R. M. (2024, May). A Holistic Approach for Cyber Security Vulnerability Assessment Based on Open Source Tools: Nikto, Acunitx, ZAP, Nessus and Enhanced with AI-Powered Tool ImmuniWeb, 2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), Tripoli, Libya, 2024, pp. 68-75, doi: 10.1109/MI-STA61267.2024.10599685.

[19] Varma, K. R., Sharma, S. S., Kulkarni, Y. Y., & Samdurle, S. S. (2023, December). ReconMaster: A CLI Based Web Reconnaissance Tool. In 2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG) (pp. 1-5). IEEE, https://doi.org/10.1109/ICTBIG59752.2023.10455983.