Vulnerability Assessment Automation Using Cyber Security

Vaishnavi Padhiyar[1], Rangani Foram Mukeshbhai[2], Dr. Kaku Riya Dharmendra[3]
[1]*M.E. Computer Engineering (Software Engineering), B H Gardi College of Engineering and Technology (GTU), Rajkot, Gujarat, India*
[2,3]*Department of Computer Engineering, B H Gardi College of Engineering and Technology, Rajkot, Gujarat, India*

***Abstract*—Vulnerability Assessment (VA) Automation is a modern cybersecurity approach that streamlines the process of identifying and analyzing security weaknesses in digital systems. Traditional vulnerability assessments are often manual, time-consuming, and prone to human error. This project aims to automate the VA process by integrating powerful open-source tools such as Nmap, OWASP ZAP, Burp Suite, Nikto, OpenVAS, and MobSF to scan and assess networks, web applications, mobile apps, and infrastructure components like routers and firewalls. The automated system is developed using a Flask-based GUI, allowing users to launch scans, monitor progress, and generate detailed reports in various formats (CSV, JSON, database). It supports multi-threaded scanning for performance, authenticated scans using SSH / SNMP / HTTP, and connects to vulnerability databases like CVE and Exploit DB to ensure up-to-date analysis. By automating the VA process, this tool significantly reduces manual effort, increases accuracy, and accelerates vulnerability detection and reporting, helping organizations to strengthen their security posture and respond to threats more efficiently.**

***Index Terms*—Vulnerability Assessment, Cybersecurity, Security Automation, Nmap, OWASP ZAP, Burp Suite, Nikto, OpenVAS, MobSF, Flask, Multi-threaded Scanning**

I. INTRODUCTION

The rapid growth of digital infrastructure, web applications, mobile applications, cloud services, and network-connected devices has significantly increased the number of potential cybersecurity vulnerabilities. Vulnerabilities such as outdated software, misconfigured services, weak authentication, insecure web interfaces, exposed ports, and improper security configurations can be exploited by attackers to gain unauthorized access or compromise sensitive information.

Vulnerability Assessment (VA) is an important cybersecurity process used to identify, analyze, and prioritize security weaknesses in information systems. Traditional VA processes generally require security professionals to configure and execute multiple tools separately, analyze their outputs, remove duplicate findings, validate vulnerabilities, and prepare reports manually. This process can become time-consuming and may introduce human errors, particularly when multiple systems and different types of targets need to be assessed.

Several security tools are available for specific assessment activities. For example, Nmap is commonly used for network discovery and port scanning, Nikto for web-server assessment, OWASP ZAP and Burp Suite for web application security testing, OpenVAS for vulnerability scanning, and MobSF for mobile application security analysis. However, using these tools individually can result in a fragmented assessment process. Therefore, this research focuses on developing an integrated VA Automation platform that brings multiple vulnerability assessment tools together through a centralized Flask-based Graphical User Interface (GUI). The proposed system aims to simplify scan execution, monitor scanning activities, collect results, perform vulnerability analysis, and generate structured reports from a single platform.

II. RESEARCH MOTIVATION AND PROBLEM

A. Motivation

- Reduction of repetitive manual effort during scan execution, result collection, and reporting.

- Integration of multiple security tools into a centralized assessment workflow.
- Reduction of fragmentation caused by separate tool interfaces and output formats.
- Support for different target environments, including networks, web applications, mobile applications, servers, routers, and firewalls.
- Use of multi-threaded execution where appropriate to improve workflow efficiency.
- Centralized vulnerability information and structured reporting using CVE/ExploitDB references and CSV, JSON, or database outputs.

B. Problem Statement

Traditional vulnerability assessment can become time-consuming and fragmented because different security tools are commonly operated separately. Manual collection, analysis, validation, and reporting of findings can increase repetitive work and may introduce configuration mistakes, missed findings, or inconsistent reporting. The research problem addressed in this work is therefore the design of a centralized automation platform capable of coordinating multiple vulnerability assessment activities and presenting their outputs through a common workflow.

III. RELATED WORK

The literature survey covers studies involving web application assessment, automated reporting, CCTV/DVR VAPT, Raspberry-Pi-based assessment, open-source vulnerability detection, WordPress assessment, large-scale vulnerability research, automated severity assessment, penetration-testing workflows, and application/API security. The reviewed studies demonstrate the value of automation but also show variation in target type, tool selection, automation level, and reporting method.

A. Research Gap

A practical gap can be stated as the need for a centralized workflow that coordinates multiple types of vulnerability assessment rather than focusing on only one target or one assessment function. The proposed design addresses this gap by combining network, web, infrastructure, and mobile assessment capabilities in one platform.

This should be described as an integration and workflow contribution; it should not be claimed as universally superior to every existing tool or paper without controlled experimental evidence.

IV. LITERATURE REVIEW

The following table summarizes the 16 studies reviewed for the proposed Vulnerability Assessment Automation research. The information is retained from the DP-1 review material and formatted for academic presentation.

TABLE I. SUMMARY OF REVIEWED STUDIES

Ref.	Year	Journal / Publication	Method / Model Used	Target / Sample	Major Findings / Performance	Limitations
[1]	2026	Priviet Social Sciences Journal	OWASP WSTG + NIST SP 800-115; Nuclei, Nmap, Subfinder, Acunetix	1 HRMS website	Identified Low and Informational findings related to security headers, cookies and client-side security; no High /Critical findings within scope.	No exploitation, brute-force, privilege escalation or DoS testing; internal infrastructure outside scope.
[2]	2026	IEEE conference paper	Slack-based automated scanner, payload testing, Google Gemini API, ngrok, AI-assisted reporting	Web applications / URLs	Detects SQL injection, command injection, subdomain takeover and insecure APIs; provides AI-assisted explanations and remediation information.	Existing scanners may miss complex vulnerabilities; prioritization remains a challenge.

[3]	2026	Proceedings of ICICIT 2026	AI-driven framework: ARIE, STEM, DRPA, CRDF; adaptive reconnaissance and behavioral payload analysis	DVWA and OWASP Juice Shop	Reported 95.6% detection accuracy compared with traditional 76.4%, ML 84.9% and deep learning 89.7%.	Limited to web applications; zero-day, mobile and cloud-native vulnerabilities not explicitly addressed.
[4]	2025	Journal of Applied Informatics and Computing	OWASP Top Ten, Nmap, Acunetix, ZAP	1 website	Nmap found 9 open ports; Acunetix identified 1 High, 5 Medium and 8 Low vulnerabilities; four vulnerabilities were successfully exploited.	Assessment focused on one website; some findings could not be externally exploited due to limited access.
[5]	2025	IEEE Transactions on Intelligent Transportation Systems	SODTA, Attacker-Defender game, MILP, Bender's decomposition	Electrified road network in North Carolina; EV/fuel traffic and charging stations	Identified critical charging stations and quantified network vulnerability; load redistribution can mitigate losses.	Computational efficiency can be improved; power-grid topology and traffic dynamics not fully integrated.
[6]	2025	IEEE International Conference on Knowledge Engineering and Communication Systems	OWASP ZAP + Isolation Forest + live traffic monitoring + anomaly detection	Simulated web environments; multiple test scenarios	Detects OWASP vulnerabilities and abnormal traffic; provides severity classification and automated PDF reports.	High computational load during simultaneous scanning/monitoring; limited zero-day adaptability.
[7]	2025	IJSRCSEIT	Python + Nmap + Nikto + Metasploit + credential/firmware testing + Flask/React dashboard	Multiple CCTV cameras and DVRs	Reported 87% efficacy in identifying critical risks and about 65% reduction in manual testing time.	Focused mainly on CCTV/DVR; proprietary firmware/protocol support may require further work.
[8]	2025	IJERT	Raspberry Pi 4 + Kali Linux + Nmap + OpenVAS + Nikto + Metasploit + Python/Bash	Raspberry-Pi-based simulated /remote environment	Low-cost, portable, remote scanning and automation; suitable for basic-to-intermediate assessment.	Processing/storage limitations; scanning depth lower than enterprise platforms.

[9]	2024	Electronics	Agent-server + Nmap + Python + automated reporting	Laboratory and external organizations /testers	Automated discovery, scanning, vulnerability identification and reporting with positive external-user evaluation.	Some backported vulnerabilities may not be detected.
[10]	2023	Journal of Computer Networks, Architecture and High-Performance Computing	Nmap + WPScan	1 WordPress website	Identified security vulnerabilities in the tested WordPress website and demonstrated usefulness of network-based scanning.	Single website; limited generalization; no standard accuracy/precision/recall metrics reported.
[11]	2023	Computer Science Review	SLR + Python data mining + author/community graphs + Louvain algorithm	23,459 final articles from 1975–2022	Identified 15 major research communities and 126 subcommunities in cybersecurity vulnerability research.	Scopus was the only data source; relevant publications outside Scopus may be missed.
[12]	2022	Atmósfera	Systematic Literature Review; vulnerability assessment methods and indicators	57 studies	Identified 21 vulnerability-assessment methods and significant methodological differences among studies.	Different assessment methods, indicator weights and normalization approaches make comparison difficult.
[13]	2022	Array, Vol. 15, 100209	Machine Learning, Logistic Regression, text mining, majority voting, CVSS, NVD/SecurityFocus /CERT/vendor data	156,040 NVD vulnerability records; CPS vulnerabilities	Attack Vector accuracy of 90.36% using NVD and 93.68% using NVD + SecurityFocus.	Inconsistent data sources, class imbalance and dependence on vulnerability descriptions.
[14]	2022	Journal of Web Engineering	Raspberry Pi + Kali Linux + Nmap + Nessus/OpenVAS + Metasploit; seven-phase penetration testing	Internal LAN and vulnerable/test systems	Demonstrated reconnaissance, scanning, vulnerability identification, exploitation, post-exploitation, reporting and re-testing.	Controlled test environment; Raspberry Pi resource limitations.
[15]	2021	Annals of Operations Research	Delphi + AHP based supply-chain vulnerability model	12 managers from 6 manufacturing	Identified and prioritized 26 vulnerability factors; Supply Chain Complexity was a major driver.	Small expert sample; depends on expert judgment.

				companies		
[16]	2021	International Electronics Symposium	VAIS + OWASP API Security Risks + containers + gamification + DAST	Vulnerable API/web application learning environment	Provides a legal environment for API-security learning, exploitation and enumeration using DAST-based challenges.	Primarily educational rather than a general enterprise VAPT platform.

V. COMPARISON STUDY

The comparison below uses the comparison-study information provided in the Phase-3 presentation. It has been reformatted into a continuous academic table rather than reproducing the PPT slide layout.

TABLE II. COMPARISON STUDY

Sr. No.	Publication & Year	Author	Model / Approach	Outcome
1	2026	Idalia Francisca Viegas et al.	OWASP WSTG, NIST SP 800-115, Nuclei, Nmap, Subfinder, Acunetix	Identified security issues related to headers, cookies and client-side security; no High/Critical vulnerabilities within the assessment scope.
2	IEEE paper, 2026	Raja Varshini	Slack + automated scanner + Google Gemini + payload testing	Enables developers to submit URLs through Slack, automatically scan for complex vulnerabilities and receive AI-assisted explanations and remediation information.
3	ICICIT, 2026	Arun K S et al.	ARIE + STEM + DRPA + CRDF	Integrated reconnaissance, vulnerability scanning, behavioral exploit validation and automated reporting; achieved 95.6% detection accuracy in the reported evaluation.
4	2025	Diana Rohmaniah, Wahid Miftahul Ashari, Lukman & Andriyan Dwi Putra	OWASP Top 10 + Nmap, Acunetix, ZAP	Nmap identified 9 open ports; Acunetix detected 1 High, 5 Medium and 8 Low vulnerabilities, with four vulnerabilities successfully exploited.
5	IEEE TITS, 2025	Zhang et al.	SODTA + Attacker-Defender game + MILP + Bender's decomposition	Identified vulnerable EV fast-charging stations and quantified network performance loss under attacks; charging-load redistribution can reduce impact.
6	ICKECS, 2025	Gupta et al.	OWASP ZAP + Isolation Forest + live monitoring	Combined vulnerability scanning with real-time anomaly detection and automated reporting for web applications.
7	IJSRCSEIT, 2025	Auti et al.	Python + Nmap + Nikto + Metasploit + automated CCTV/DVR VAPT	Automated CCTV/DVR security testing, vulnerability identification and reporting while reducing manual assessment time by about 65%.

Sr. No.	Publication & Year	Author	Model / Approach	Outcome
8	IJERT, 2025	Shaik Sanheera	Raspberry Pi 4 + Kali Linux + Nmap + OpenVAS + Nikto + Metasploit	Demonstrated a low-cost, portable and remotely accessible vulnerability assessment platform suitable for basic-to-intermediate assessments.
9	Electronics, 2024	Seara & Serrão	Agent-server + Nmap + Python + automated reporting	Developed an automated vulnerability scanner that performs discovery, scanning, vulnerability identification and reporting, with positive external-user evaluation.
10	2023	Dewi Laksmiati	Nmap + WPScan	Conducted network-based vulnerability assessment of a WordPress website and identified security weaknesses requiring remediation.
11	Computer Science Review, 2023	Heiding, Katsikeas & Lagerström	SLR + Python data mining + Louvain algorithm	Large-scale analysis of cybersecurity vulnerability research. 23,459 papers were synthesized and 15 research communities with 126 subcommunities were identified.
12	2022	Areli Nájera-González & Fátima Maciel Carrillo-González	Systematic Literature Review	Reviewed 57 studies and compared vulnerability concepts, assessment methods, indicators and tools. Identified significant methodological differences between studies.
13	Array, 2022	Jiang & Atif	Logistic Regression + text mining + CVSS + majority voting	Improved automated vulnerability severity assessment and achieved 90.36% AV accuracy with NVD and 93.68% with NVD + SecurityFocus.
14	Journal of Web Engineering, 2022	Lachkov, Tawalbeh & Bhatt	Raspberry Pi + Kali Linux + Nmap + Nessus/OpenVAS + Metasploit	Demonstrated a complete seven-phase penetration-testing process in an internal LAN, including reconnaissance, vulnerability identification, exploitation, reporting and re-testing.
15	2021	Satyendra Kumar Sharma et al.	Delphi + AHP	Developed a hierarchical supply-chain vulnerability model and prioritized 26 vulnerability factors. Supply Chain Complexity was identified as the major vulnerability driver.
16	IES, 2021	Idris, Syarif & Winarno	VAIS + OWASP API Security + containers + gamification	Created a safe API-security learning environment where users can practice API exploitation and enumeration using DAST-based challenges.

VI. PROPOSED VULNERABILITY ASSESSMENT AUTOMATION SYSTEM

The proposed system is a Flask-based graphical interface that acts as a centralized control and reporting layer for vulnerability assessment tools. The user supplies or selects the authorized target and assessment activity. The platform manages supported scanning tasks, collects outputs, normalizes the results, associates' findings with vulnerability information, and generates structured reports.

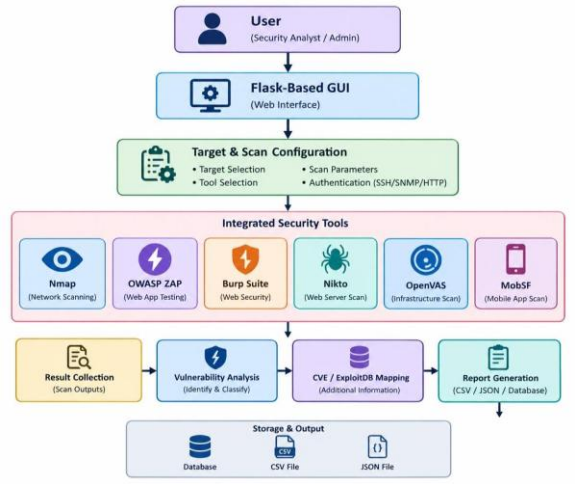

Fig. 1. Integrated Vulnerability Assessment Automation Architecture

A. Integrated Security Tools

TABLE III. INTEGRATED SECURITY TOOLS

Tool	Primary role	Target area
Nmap	Host discovery, port scanning, service and OS detection	Network
OWASP ZAP	Web application security testing	Web
Burp Suite	HTTP/HTTPS inspection and web security testing	Web
Nikto	Web-server checks and configuration/known-issue detection	Web
OpenVAS	Vulnerability scanning and infrastructure assessment	Network / Infrastructure
MobSF	Mobile application security analysis	Mobile

B. Multi-threaded Scanning

Multi-threading to allow supported scanning activities to execute concurrently instead of waiting for every task to finish sequentially. For example, separate targets or assessment jobs can be assigned to different execution threads where technically and ethically appropriate. The final paper should report measured time values only after repeated controlled tests.

C. Authenticated Scanning

The proposed platform considers authenticated assessment through SSH, SNMP, and HTTP where supported by the target and tool. Authenticated assessment can provide deeper visibility into configurations, installed software, services, and security settings than an unauthenticated assessment. Credentials should only be used with explicit authorization.

D. Vulnerability Information and Reporting

The platform can associate detected findings with CVE information and ExploitDB references to provide additional context. The DP-1 report specifies structured outputs in CSV, JSON, and database formats. This centralization is important because different tools may produce different output formats and levels of detail.

VII. PROPOSED WORKFLOW

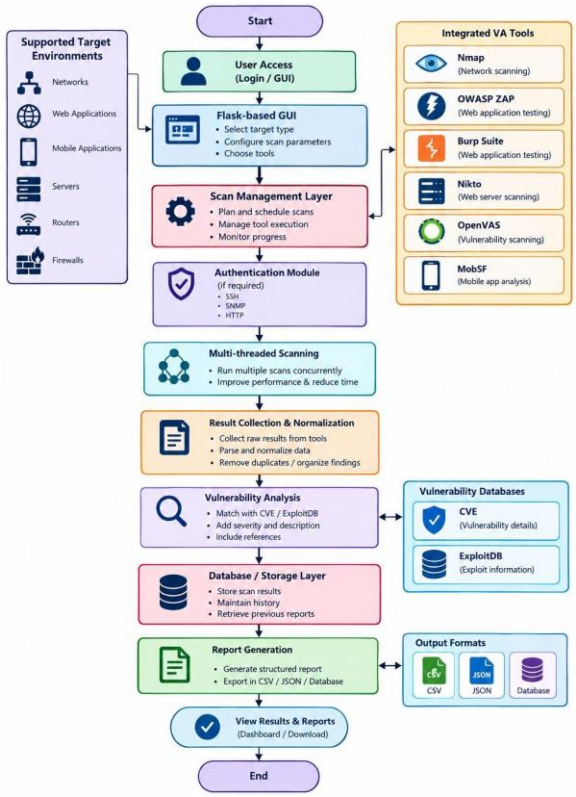

Fig. 2. Proposed automated vulnerability assessment workflow

A. Key Learning from Other Papers

The literature survey suggests that the research contribution should be positioned around centralized integration and workflow management rather than simply counting tools. A paper using fewer tools can still solve a different or deeper problem. Therefore, the

meaningful comparison is whether the proposed platform combines complementary assessment functions, target types, result handling, and reporting in one workflow.

The documented six named tools provide complementary coverage: network discovery, web application assessment, web-server checking, infrastructure scanning, and mobile application analysis. This breadth can reduce the need to move between separate platforms during a multi-domain assessment, provided the final implementation successfully integrates and executes each tool.

VIII. EVALUATION METHODOLOGY

The final implementation should be evaluated using repeatable authorized test cases. The following metrics are recommended because they can demonstrate the claimed automation benefits without fabricating results.

TABLE IV. EVALUATION METRICS

Metric	How to measure	Purpose
Tool integration coverage	Number of successfully integrated and executable tools	Verify integration breadth
Target coverage	Number/ type of authorized target categories successfully tested	Verify scope
Scan execution time	Measure identical test cases with separate-tool workflow and centralized workflow	Assess time effect
Manual steps	Count configuration, execution, result collection, and reporting steps	Assess workflow reduction
Finding consistency	Compare normalized outputs with original tool reports	Check result preservation
Reporting time	Measure time from completed scans to final report	Assess reporting automation
Severity distribution	Count Critical/ High/Medium/ Low/Informational findings on the same test set	Characterize findings

IX. DISCUSSION

The proposed platform is designed around a practical workflow problem: vulnerability assessment may involve several specialized tools, but the analyst still needs a common place to initiate scans, monitor progress, consolidate findings, and prepare reports. The architecture therefore separates specialized detection functions from centralized orchestration and reporting.

An important lesson from the reviewed papers is that tool quantity alone is not an adequate measure of research contribution. The proposed work should demonstrate that each integrated tool has a defined role, that outputs are collected correctly, and that centralized execution does not remove important findings or introduce inconsistent interpretation. Controlled experiments are therefore essential for supporting claims about time reduction, accuracy, or efficiency. The report also identifies multi-threading, authenticated scanning, CVE/ExploitDB enrichment, and structured reporting as important design elements. These features can strengthen the practical usefulness of the platform when they are implemented and validated.

X. CONCLUSION

This paper presents a centralized approach to Vulnerability Assessment Automation using a Flask-based platform. The documented design integrates Nmap, OWASP ZAP, Burp Suite, Nikto, OpenVAS, and MobSF and is intended to support network, web, infrastructure, and mobile assessment activities. The central contribution is the coordination of multiple specialized tools through a common workflow for scan execution, result collection, vulnerability analysis, and reporting.

The literature review shows that existing studies frequently focus on a specific target, assessment function, or a smaller set of tools. In contrast, the proposed design emphasizes broader integration across several assessment domains. This does not by itself establish superior security performance; instead, it defines a practical research direction that should be validated through controlled experiments measuring execution time, manual effort, reporting time, finding consistency, and target coverage.

XI. FUTURE WORK

- Complete integration and validation of all tools intended for the final implementation.

- Add the remaining tool names to the paper once their implementation is confirmed.
- Perform controlled benchmark tests comparing separate-tool execution with centralized execution.
- Add duplicate-finding detection and result normalization across heterogeneous tool outputs.
- Improve severity/risk prioritization using CVSS and contextual asset information.
- Extend authenticated scanning and credential management with secure handling.
- Add dashboards for scan status, severity distribution, trend analysis, and remediation tracking.
- Evaluate the platform using larger and more diverse authorized test environments.

REFERENCES

[1] "Vulnerabilities assessment of a web-based human resources management system using a penetration testing approach."

[2] "Automated web application security reporting system with Slack integration."

[3] "Web application penetration testing: Attack analysis and documentation."

[4] "Enhancing website security using vulnerability assessment and penetration testing (VAPT) based on OWASP Top Ten."

[5] "Vulnerability assessment of charging stations in the electrified road network."

[6] "Enhancing web application security using penetration testing and vulnerability scanning."

[7] "Automated vulnerability assessment and penetration testing tool for CCTV cameras and DVR."

[8] "Vulnerability assessment & penetration testing for applications security," *International Journal of Engineering Research & Technology (IJERT)*, 2025.

[9] "Automation of system security vulnerabilities detection using open-source software," *Electronics*, 2024.

[10] D. Laksmiati, "Vulnerability assessment with network-based scanner method for improving website security," *Journal of Computer Networks, Architecture and High-Performance Computing*, 2023.

[11] M. Heiding, S. Katsikeas, and L. Lagerström, "Research communities in cybersecurity vulnerability assessments: A comprehensive literature review," *Computer Science Review*, 2023.

[12] N. Nájera-González and M. Carrillo-González, "Vulnerability assessment studies on climate change: A review of the research in Mexico," *Atmósfera*, 2022.

[13] J. Jiang and Y. Atif, "Towards automatic discovery and assessment of vulnerability severity in cyber–physical systems," *Array*, 2022.

[14] M. Lachkov, L. Tawalbeh, and S. Bhatt, "Vulnerability assessment for applications security through penetration simulation and testing," *Journal of Web Engineering*, 2022.

[15] S. K. Sharma *et al.*, "Supply chain vulnerability assessment for manufacturing industry," *Annals of Operations Research*, 2021.

[16] Idris, Syarif, and Winarno, "Development of vulnerable web application based on OWASP API Security Risks," *IES*, 2021.