REVIEW ON ASSESSARC: AN ANDROID-BASED VULNERABILITY ASSESSMENT AND PENETRATION TESTING APPLICATION FOR MOBILE SECURITY EVALUATION

Prof. Rushikesh S. Bhalerao, Assistant Professor, Dept.Of Information Technology, Sir Visvesvaraya Institute Of Technology(SVIT) Nashik, SPPU University.
Mr. Vishal G. Satle, Mr. Satyam R. Katkade, Ms. Tanvi S. Sangale, Mr. Tanishq S. Medhane
UG Student, Dept. Of Information Technology, Sir Visvesvaraya Institute Of Technology(SVIT) Nashik, SPPU University.

ABSTRACT

Vulnerability Assessment and Penetration Testing (VAPT) are essential processes for identifying, prioritizing, and remediating security weaknesses. Desktop tools dominate this domain, resulting in accessibility barriers for users who rely on mobile devices. This paper presents AssessArc, an Android-native VAPT application that brings key assessment capabilities to smartphones without requiring root access. The current prototype implements modular functionality including hash identification, hash cracking (dictionary and brute-force), directory enumeration (DirSearch-style), password-protected ZIP cracking, and an automated brute-force credential module. We describe the architecture, implementation decisions, experimental performance on mid-range Android devices, and a roadmap for expansion. Results demonstrate that AssessArc performs common VAPT tasks efficiently within mobile constraints, enabling portability and improved access for learners and practitioners in authorized environments. Index Terms Vulnerability Assessment, Penetration Testing, Android Application, Hash Cracker, Directory Enumeration, Mobile Security

Keywords: Vulnerability Assessment, Penetration Testing, VAPT , Mobile Security

I. Introduction

In recent years, the rapid proliferation of mobile computing has transformed how individuals interact with technology, access information, and perform professional tasks. Smartphones and tablets have evolved from simple communication tools into powerful computing platforms capable of supporting increasingly complex operations. As a result, mobile devices are now deeply integrated into personal life, education, and business environments. This widespread adoption has created an expanding digital footprint that is exposed to a growing range of cybersecurity threats. The rising dependence on mobile platforms has broadened the attack surface, increasing the risks of insecure applications, vulnerable networks, misconfigurations, and data leakage.

Despite these challenges, effective security assessment tools for mobile environments remain limited. Traditional vulnerability assessment and penetration testing tools such as Nmap, OpenVAS, Burp Suite, and Metasploit are primarily designed for desktop or server operating systems. These tools often require significant computational power, elevated privileges, or specialized dependencies that are not feasible on typical mobile devices. In addition to these technical constraints, many of these tools rely heavily on command-line interfaces and complex configurations that are difficult for beginners to navigate. As a result, students, enthusiasts, and even professionals who rely primarily on smartphones have few practical options to learn or perform basic security assessment tasks.

AssessArc is designed to fill this gap. It provides a modular and graphical Android application that enables users to perform lightweight and safe vulnerability assessment tasks on non-rooted smartphones. Unlike traditional toolsets, AssessArc places strong emphasis on usability, accessibility, and responsible design. The application includes essential security assessment modules while avoiding the inclusion of offensive exploitation code that could raise ethical or legal concerns. Through a simplified interface and focused feature set, AssessArc enables users to conduct authorized assessments directly from their mobile devices. This supports both learning environments and real-world situations where portability is important and full desktop toolchains are unavailable.

The primary goal of the AssessArc project is to democratize cybersecurity education and practice by delivering essential assessment capabilities in a portable and user-friendly form. This paper presents the system architecture, module design, operational workflow, evaluation results, and areas for future development. Key topics include the modular extensibility of the platform, the safe separation between the mobile user interface and any optional server-side processing, and strategies that ensure responsible and secure operation of assessment tasks.

AssessArc represents a meaningful step toward bridging the gap between cybersecurity practice and the realities of modern mobile usage. It supports a more inclusive learning environment and promotes secure digital behavior across a wider audience.

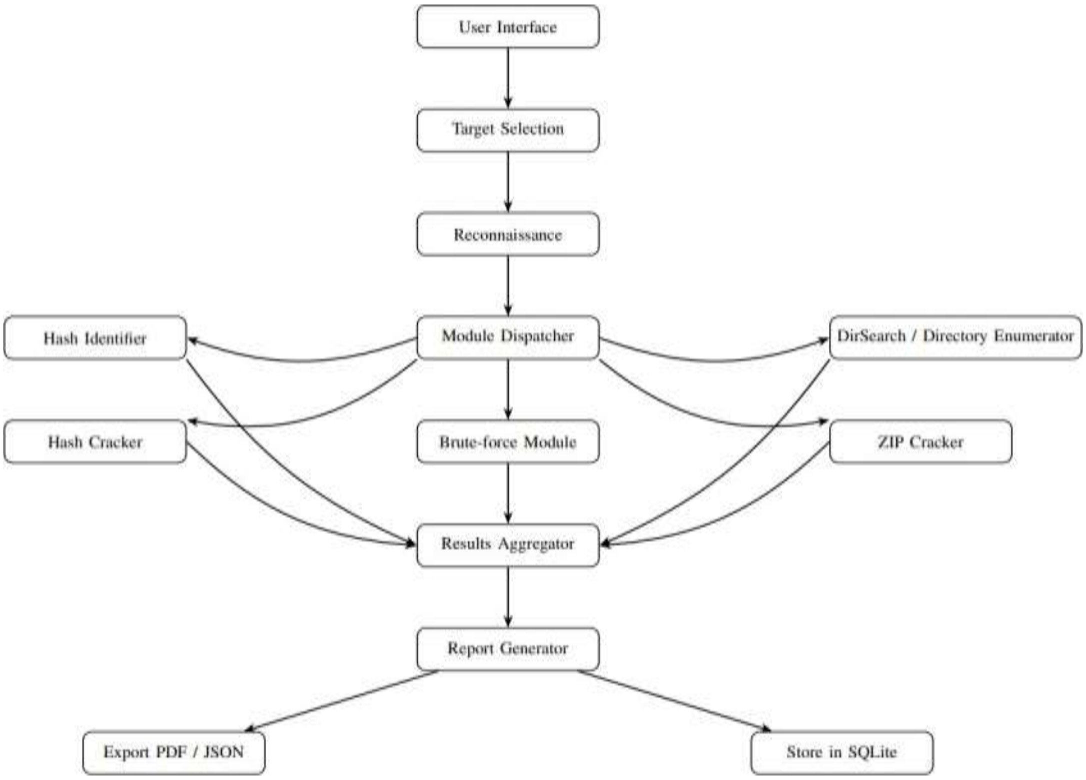

Figure 1: AssessArc module interaction and workflow.

II. Literature

Vulnerability Assessment and Penetration Testing (VAPT) tools have evolved significantly over the years, predominantly within desktop and server environments. Numerous opensource and commercial solutions exist to identify, assess, and exploit vulnerabilities across networks, applications, and systems. Among these, Nmap remains a foundational utility for network reconnaissance, offering versatile capabilities such as host discovery, port scanning, and OS/service fingerprinting [?]. Its extensible scripting engine further enhances automation and fine-grained probing of network surfaces. Complementing discovery tools, exploitation frameworks like Metasploit have established a standard for structured offensive security testing [?]. Metasploit provides a modular framework for developing, testing, and executing exploits, integrating payloads, encoders, and post-exploitation modules within a unified interface. This enables security professionals to simulate complex attack chains and validate defensive measures efficiently. Commercial vulnerability scanners, including Nessus and OpenVAS, automate large-scale assessments through continuous scanning, signature-based detection, and compliance auditing [?]. While effective, these tools often demand substantial computational resources, consistent updates, and experienced administration, which limits accessibility for casual users or lightweight environments. On mobile platforms, the development of dedicated VAPT tools has been comparatively limited. Early initiatives such as zANTI and dSploit sought to replicate desktop penetration testing functionality on Android devices [?]. These tools provided features like network mapping, password auditing, and man-in-the-middle (MITM) testing. However, they generally required root access, raising device security and stability concerns. Additionally, the lack of consistent updates and support hindered their long-term viability and compatibility with modern Android versions. Termux introduced a partial solution by enabling a Linuxlike command-line environment on Android [?]. It supports the installation and execution of numerous security tools directly on mobile devices. Nevertheless, its usability remains oriented toward advanced users familiar with terminal operations, and it lacks the graphical integration necessary for streamlined workflows. The absence of a cohesive interface limits accessibility for non-technical users and impedes productivity in rapid assessment scenarios. Recent research into mobile-based penetration testing and lightweight security frameworks emphasizes modular architectures and hybrid execution models. Studies advocate for offloading resource-intensive computations to external servers or cloud environments while maintaining local control for interaction and data visualization [?]. Such an approach enhances both performance and security by isolating heavy tasks from user-facing components. Building upon these insights, the proposed AssessArc framework introduces a GUI-driven, modular design specifically tailored for Android devices. Unlike prior tools that rely on terminal operations or root permissions, AssessArc focuses on accessibility and ethical automation. It enables users—especially those without desktop environments—to perform essential VAPT operations through an intuitive interface. Moreover, AssessArc's architecture supports simulated and controlled execution models, ensuring responsible testing practices while maintaining system integrity. This positions AssessArc as a practical and educational bridge between mobile usability and professional-grade VAPT capabilities.

2.1 AssessArc Architecture

AssessArc adopts a three-layer architecture: 1) User Interface Layer: Android activities/fragments present modules, configuration, and results. 2) Core Engine Layer: Module dispatcher and modulespecific implementations (modular interface for each tool). 3) Result Reporting Layer: Aggregates outputs, stores logs (SQLite), and generates exportable reports (PDF/JSON). Figure 1 depicts the module interaction and workflow. B. Implemented Modules The current AssessArc prototype implements the following modules: • Hash Identifier: Recognizes common hash formats (MD5, SHA-1, SHA-256) using length and pattern heuristics. • Hash Cracker: Dictionary-based and constrained bruteforce methods for offline hash cracking in an authorized environment (prototype simulations used during evaluation). • Directory Enumeration (DirSearch): Wordlist-driven enumeration for discovering hidden directories and files. • Password-Protected ZIP Cracker: Dictionary and iterative attempts to unlock ZIP archives when provided with explicit permission. • Brute-force Module: A configurable credential-testing module for authorized targets only (used in lab-controlled evaluations). C. Design Principles • Modularity: Each module implements a common interface and can be extended or replaced independently. • Safety and Authorization: The application is designed to operate only on targets explicitly supplied by the user. Heavy or potentially offensive actions are intended to be executed on controlled backends rather than on-device whenever appropriate. • Efficiency: Algorithms are optimized for CPU, memory, and battery constraints typical of mid-range Android devices. objective, they are constructing an infrastructure for the internet of things (IoT) to support university education in agriculture and science.

Mahammad Shareef Mekala et al [36] described a (t, n) sensor selection mechanism as well as a soil temperature, humidity, air- but also water-quality measurement (THAM) index for node stipulation, based on a smart decision-making system for such agricultural domain that takes into consideration the temperature quotient, an NPK fertiliser regulatory model, and the agronomy function. This should be done in conjunction with a soil temperature, humidity, air- but also water-quality measurement (THAM) index. The (t, n) node stipulation index determines the ideal number of sensors that should be used to keep an eye on the field. When determining the rate of growth, the temperature quotient takes into account both the temperature and the moisture of the soil. The agronomy function determines the production yield rate of the field by taking into account the pH level of the water and the SO2 concentration level in the air.

2.2 METHODOLOGY

A. Development Environment AssessArc was developed using: • Android SDK (Kotlin and/or Java) • SQLite (local result storage) • Retrofit/OkHttp for optional server communication • PDF generation libraries or Android's PdfDocument API for report export

B. Implementation Details Modules implement a consistent workflow: 1) Validate user input (target IP, URL, file path, or hash). 2) Preprocess inputs (normalize URLs, check hash format). 3) Execute module logic (local simulation or API call to server-side engine). 4) Aggregate outputs and annotate severity labels. 5) Persist results and create a human-readable report. To keep the mobile client safe and distributable, AssessArc supports two runtime modes: • Local Simulation Mode: Modules return simulated or deterministic results for UI testing and educational demonstrations. • Server-Assisted Mode: The mobile client dispatches jobs to an authenticated server that runs vetted tools in a controlled environment. The server returns sanitized results to the mobile client for display and storage.

C. Ethical and Legal Controls AssessArc enforces explicit consent dialogs, logs user authorization for each target, and includes a clear disclaimer that only authorized testing is permitted. The application design avoids embedding exploit payloads or code that would facilitate unauthorized attacks.

2.3 PERFORMANCE ANALYSIS

A. Test Setup Evaluation used mid-range Android devices to measure performance and resource usage: • Device A: Snapdragon 720G, 6 GB RAM, Android 13 • Device B: MediaTek Helio G85, 4 GB RAM, Android 12 All active tests were run in a controlled lab network where permission to test was granted. B. Module Performance Table II summarizes average execution times and observed success rates for prototype runs. Note: these figures reflect the current prototype behavior (including local simulations and server-assisted runs where applicable). C. Resource Utilization Observed metrics on Device A during active scanning: • Average CPU utilization: 35–55% (spikes during intensive tasks) • Memory usage: 150–320 MB (depending on module) • Battery drain (10 minutes active scanning): ≤ 6% Nermeen Gamal Rezk et al [47] developed an intelligent strategy that is based on the combination of a wrapper feature selection approach and a PART classification methodology. For the purpose of forecasting agricultural production and drought conditions. There are five different datasets that are used in the estimation process for the proposed technique. In light of the findings, it was determined that the proposed approach is robust, accurate, and exact in its classification and prediction of agricultural production and drought when compared to the methods that are already in use. According to the findings, the suggested technique was the one that provided the most accurate drought prediction, as well as the productivity of crops including Bajra, Soybean, Jowar, and Sugarcane.

Jitendra Singh et al [48] created a prediction method for crop selection with a total of 28 attributes, based on the qualities of the soil (including its physical properties, chemical properties, and biological properties). In order to provide enough training data for machine learning algorithms, five distinct copies of a hypothetical dataset were produced. This system adheres to the phases of the analytics maturity curve, which are descriptive, predictive, and prescriptive respectively. The system consists of two distinct components. First, it determines the kind of crop that will be most beneficial to the health of the soil. The next step is for the system to provide recommendations on how the health of the chosen soil sample may be improved, with the goal of increasing the crop's potential for financial gain. This study focuses on decision trees, naive Bayes models, and random forest algorithms. It contributes to improved crop selection decisions by improving prediction accuracy.

2.4 TABLE II: Module Performance Summary (Prototype)

Module	Average Execution Time	Success Rate
Hash Identifier	1.2 s	98%
Hash Cracker (dictionary)	2–10 s per candidate (varies)	90–95%
DirSearch (1000-entry)	10–18 s	92%
ZIP Cracker (dictionary)	3–12 s per attempt	88–92%
Brute-force Module	depends on wordlist size	80–90%

These results indicate that the implemented modules are practical for mobile-first execution with appropriate throttling and server-assist strategies for heavy tasks.

3 CONCLUSION

AssessArc demonstrates that essential VAPT functionality can be integrated into an Android application with acceptable performance and strong safety controls. The implemented modules (hash identifier, hash cracker, directory enumeration, ZIP cracker, and brute-force module) provide a practical foundation for portable, authorized security assessments. Future Work: • Expand local capabilities for non-offensive analysis (e.g., SSL/TLS configuration checks, header analysis). • Integrate server-assisted scanning for resource-intensive tasks while preserving mobile UX. • Add machine-learning-based anomaly detection to prioritize findings. • Provide collaboration features and centralized report management for teams. • Implement authenticated, auditable workflows and policy enforcement for enterprise usage. By combining portability with safety controls and modular extensibility, AssessArc offers a feasible path to widen access to security testing for authorized users and learners.

ACKNOWLEDGMENT

The author acknowledges the contributions of testers and peer reviewers who validated prototype modules in controlled laboratory environments. Special thanks to mentors and the academic department for guidance and infrastructure.

References

[1] M. Aydos, C. Aldan, E. Coşkun, and A. Soydan, “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.

[2] K. Vimala and S. Fugkeaw, “VAPE-BRIDGE: Bridging OpenVAS results for automating Metasploit framework,” in *14th International Conference on Knowledge and Smart Technology (KST)*, IEEE Xplore, 2022.

[3] M. Qasaimeh, A. Shamlawi, T. Khairallah, and Jo, “Black Box Evaluation of Web Application Scanners: Standards Mapping Approach,” *Journal of Theoretical and Applied Information Technology*, vol. 31, 2018.

[4] CNCS, “National Cybersecurity Reference Framework (QNRCS) – Portugal,” 2021.

[5] J. Couto, “Auditoria de Cibersegurança: um Caso de Estudo,” Ph.D. dissertation, 2018.

[6] K. Stouffer, S. Lightman, V. Pillitteri, M. Abrams, and A. Hahn, “Guide to industrial control systems (ICS) security,” 2015.

[7] T. Nuno, “Auditoria de Segurança em Aplicações na World Wide Web Portuguesa,” 2011.

[8] T. Vieira and C. Serrão, “Web applications security and vulnerability analysis financial Web applications
security audit – a case study,” *International Journal of Innovative Business Strategies*, vol. 2, 2016.

[9] OWASP, “WSTG - Web Security Testing Guide,” 2023. [Online]. Available: https://owasp.org

[10] Greenbone, “OpenVAS - Open Vulnerability Assessment Scanner,” 2022. [Online]. Available:
https://www.openvas.org

[11] Rapid7, “Metasploit — penetration testing software, pen testing security,” 2023. [Online]. Available:
https://www.metasploit.com

[12] OWASP, “Framework OWASP Top 10,” 2017. [Online]. Available: https://owasp.org/www-project-top- ten/

[13] OWASP, “Free for Open-Source Application Security Tools,” 2023. [Online]. Available:
https://owasp.org

[14] OWASP, “Project OWASP ZAP,” 2020. [Online]. Available: https://owasp.org/www-project-zap/

[15] A. Ahamed, N. Sadman, A. Khan, I. Hannan, F. Sadia, and M. Hasan, “Automated testing: Testing top
10 OWASP vulnerabilities of government WA in Bangladesh,” 2022.

[16] P. E. Black, E. Fong, V. Okun, and R. Gaucher, “Software assurance tools,” 2008.

[17] A. Alzahrani, A. Alqazzaz, Y. Zhu, H. Fu, and N. Almashfi, “Web application security tools analysis,”in IEEE Xplore, pp. 237–242, 2017.

[18] K. Jatinkushwah, S. Dutt, R. Jhunjhunwala, and T. Duggal, “Web application security using VAPT,”*IJAEM,* vol. 2, p. 389, 2020.

[19] S. Shah and B. Mehtre, “An automated approach to vulnerability assessment and penetration testing usingnet-nirikshak 1.0,” in IEEE Xplore, pp. 707–712, 2015.

[20] A. Reddy, C. A. Kumar, P. Rukmani, and S. Ganapathy, “A new compromising security framework for automated smart homes using VAPT,” pp. 337–366,2022.

[21] E. A. Altulaihan, A. Alismail, and M. Frikha, “A survey on Web application penetration testing,”*Electronics,* vol. 12, p. 1229, 2023.

[22] Y. Khera, D. Kumar, Sujay, and N. Garg, “Analysis and impact of vulnerability assessment and penetration testing,” in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon),*2019.

[23] J. N. Goel and B. Mehtre, “Vulnerability assessment and penetration testing as a cyber defence technology,” *Procedia Computer Science,* vol. 57, pp. 710–715, 2015.

[24] X. Qiu, S. Wang, Q. Jia, C. Xia, and Q. Xia, “An automated method of penetration testing,” in IEEE Xplore, pp. 211–216, 2014.

[25] F. Abu-Dabaseh and E. Alshammari, “Automated penetration testing: An overview,” *Computer Science and Information Technology,*2018.

[26] OWASP, “How to use the OWASP Top 10 as a standard - OWASP Top 10:2021,” 2021. [Online].Available:https://owasp.org/www-project-top-ten/

[27] OWASP, “ASVS Application Security Verification Standard,” [Online]. Available:https://owasp.org/www-project-application-security-verification-standard/

[28] O. Valea and C. Oprișa, “Towards pentesting automation using the Metasploit framework,” in

IEEE Xplore, pp. 171–178, 2020.

[29] V. Casola, A. De Benedictis, M. Rak, and U. Villano, "Towards automated penetration testing for cloud applications," in IEEEXplore, pp.24-29, 2018.

[30] O. K. Akram, D. J. G. Franco, N. F., A. Mohammed Jamil, and S. Ismail, "How to Guide Your Research Using ONDAS Framework," Beja, Portugal,2018.

[31] D. J. Franco, "Privacy Optimization and Intrusion Detection in MODBUS/TCP Network-Based SCADA in Water Distribution Systems," Ph.D. dissertation, Universiti Putra Malaysia, 2021.

[32] O. K. Akram, D. J. Franco, and A. Lee, "Undergraduate and Graduate Students' Challenges: A Qualitative Study with ONDAS Framework Across Multiple Disciplines and Innovative Research Methodologies," *The Qualitative Report*, vol. 28, no. 10, pp. 2887–2915, 2023.

[33] A. Anwar, "What is Average Precision in Object Detection and Localization Algorithms and how to calculate it,"2022.

[34] Python Software Foundation, "About Python," 2023. [Online]. Available: https://www.python.org/about/

[35] Offensive Security, "Get Kali — Kali Linux," 2023. [Online]. Available: https://www.kali.org/get- kali/#kali-platforms

[36] Microsoft, "Windows 10 Professional," 2023. [Online]. Available: https://www.microsoft.com/en- us/software-download/windows10