E-ISSN:2583-2468

Research Article

Vulnerability Scanner

Applied Science and Engineering Journal for Advanced Research

2026 Volume 5 Number 2 March

Publisher

Singh Publication

www.singhpublication.com

Advanced Operating System Vulnerability Scanner – DefenX

Sakthivadivel M[1*], Sarathy D[2], Gayathri V[3], Naveen Kumar K[4]

DOI:10.54741/ASEJAR/5.2.2026.185

[1*] M. Sakthivadivel, Assistant Professor Ss, Department of CSE (Cyber Security), Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu, India.

[2] Sarathy D, UG Student, Department of CSE (Cyber Security), Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu, India.

[3] Gayathri V, UG Student, Department of CSE (Cyber Security), Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu, India.

[4] Naveen Kumar K, UG Student, Department of CSE (Cyber Security), Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu, India.

A vulnerability in an operating system refers to a weakness in software, drivers, or system configurations that can be exploited by attackers to compromise confidentiality, integrity, or availability. With the widespread adoption of Windows systems in personal and enterprise environments, operating system–level vulnerabilities have become a major attack vector for privilege escalation, malware execution, and ransomware. Traditional vulnerability scanners often focus on network-level detection and generate high false positives while neglecting driver-level risks and real-time host visibility. This paper presents DefenX, a lightweight host-based vulnerability scanner designed specifically for the Windows ecosystem. DefenX performs system profiling, asset extraction, CVE correlation, risk scoring, and actionable reporting. The proposed system emphasizes driver-aware scanning and low performance overhead. Experimental evaluation shows improved prioritization accuracy and reduced system impact compared to conventional scanners.

Keywords: vulnerability scanner, CVE correlation, host security, driver vulnerabilities, risk scoring

Corresponding Author

M. Sakthivadivel, Assistant Professor Ss, Department of CSE (Cyber Security), Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu, India.

Email: sakthivadivelm@gmail.com

How to Cite this Article

Sakthivadivel M, Sarathy D, Gayathri V, Naveen Kumar K, Advanced Operating System Vulnerability Scanner – DefenX. Appl Sci Eng J Adv Res. 2026;5(2):12-15.

Available From https://asejar.singhpublication.com/index.php/ojs/article/view/185

To Browse

Manuscript Received	Review Round 1	Review Round 2	Review Round 3	Accepted
2026-02-05	2026-02-21			2026-03-10

Conflict of Interest	Funding	Ethical Approval	Plagiarism X-checker	Note
None	Nil	Yes	5.26	

1. Introduction

The rapid growth of digital systems has increased reliance on operating systems for daily computing and enterprise operations. Windows operating systems are widely used but frequently targeted due to their large attack surface and software diversity. Operating system vulnerabilities often arise from outdated software, insecure drivers, and unpatched system components. Attackers exploit these weaknesses to gain unauthorized access or escalate privileges. Although vulnerability management tools exist, many focus on network scanning rather than host-level visibility. They may also introduce performance overhead and produce excessive alerts. Therefore, there is a need for a lightweight and host-centric vulnerability scanner that provides accurate and actionable intelligence.

1.1 Vulnerability Management

Vulnerability management is a continuous process involving detection, analysis, prioritization, and remediation. Many existing solutions emphasize detection but fail to consider exploitability and system impact. This leads to inefficient remediation and alert fatigue. DefenX aims to improve this process by focusing on host-level accuracy and prioritization.

2. Literature Review

Several vulnerability assessment tools are widely used in cybersecurity. Existing scanners use signature-based detection and CVE matching to identify vulnerabilities. Many tools rely on network scanning and predefined templates. While effective for network visibility, they often lack deep host-level inspection and driver-specific auditing. Some solutions also require complex configuration and significant system resources. Recent research highlights the importance of risk-based vulnerability management and host-level security monitoring. However, limited focus has been placed on lightweight, driver-aware host scanners. DefenX addresses this gap by combining host profiling, driver auditing, and CVE correlation.

3. Problem Statement

Existing vulnerability management tools suffer from:

- High false positive rates
- Limited driver-level inspection
- Performance overhead and system lag
- Complex configuration requirements
- Delayed or non-actionable remediation guidance

These limitations reduce the effectiveness of vulnerability management and leave systems exposed.

4. Proposed System

4.1 System Overview

DefenX is a host-centric operating system vulnerability scanner designed to automate vulnerability detection, analysis, and remediation guidance for Windows systems. Unlike network-focused scanners, DefenX operates at the host level to provide deep visibility into installed software, drivers, and system components. The system continuously evaluates local assets and identifies vulnerabilities by correlating them with CVE databases. It emphasizes lightweight operation, driver-aware scanning, and actionable reporting to minimize system impact while improving security visibility.

4.2 System Components

The proposed DefenX system consists of the following components:

- Local user dashboard (GUI interface)
- Core scanning engine (Python-based)
- Asset & software extraction module
- CVE correlation engine
- Risk scoring & prioritization module
- Reporting & remediation module

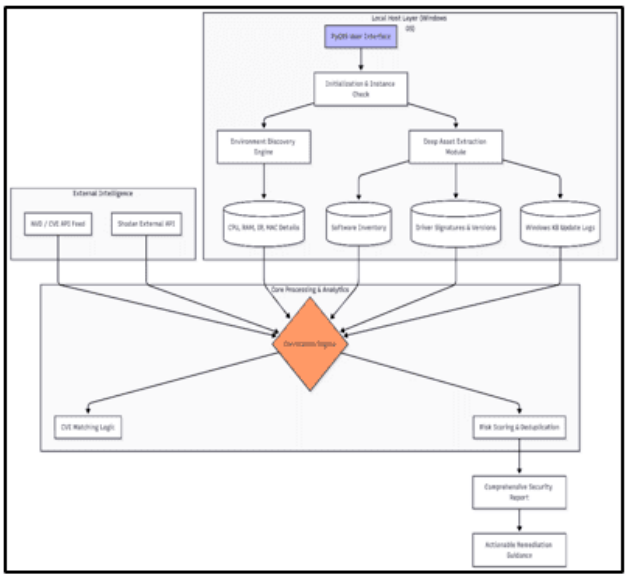

Figure 1: Overall Architecture of DefenX

5. Methodology

The system begins by profiling the host environment to collect OS details, hardware specifications, and network identity.

Next, the asset extraction module enumerates installed applications, system drivers, and Windows updates.

The collected data is correlated with CVE databases to identify known vulnerabilities.

A risk score is then calculated based on severity, exploitability, and system impact.

Finally, the system generates reports with remediation guidance, enabling users to patch or mitigate vulnerabilities efficiently.

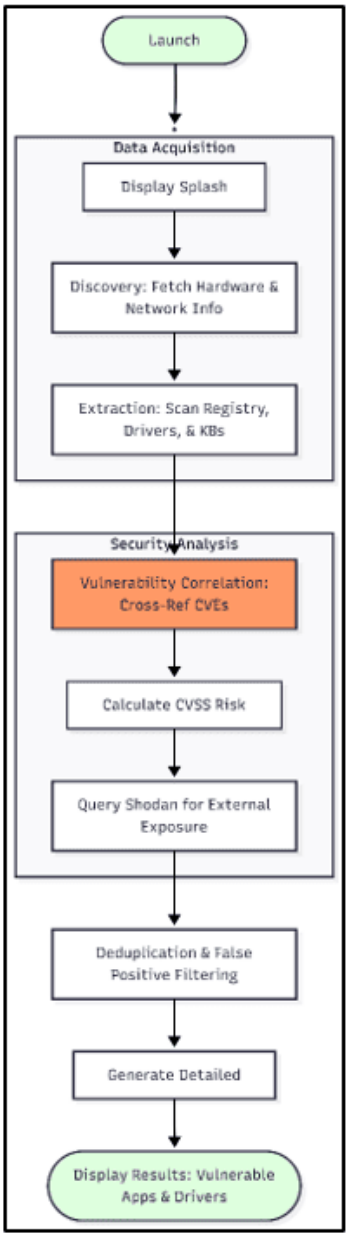

Figure 2: Work Flow of DefenX

6. Implementation

The proposed system is implemented using the following tools:

- Programming Language: Python
- GUI Framework: PyQt6
- System Information Libraries: psutil, WMI, py-cpuinfo
- Vulnerability Intelligence: CVE/NVD APIs
- Operating System: Windows 10/11

The tool runs locally and supports real-time host-based vulnerability assessment with minimal resource usage.

7. Results and Discussion

DefenX was tested on Windows systems containing outdated software and vulnerable drivers. The results indicate that DefenX effectively detects vulnerable applications and insecure drivers often overlooked by general scanners. The lightweight design reduces performance overhead while maintaining detection accuracy. Risk-based prioritization helps reduce alert fatigue by focusing on critical vulnerabilities.

8. Conclusion

This paper presented DefenX, a host-based vulnerability scanner that enhances operating system security through asset discovery, CVE correlation, and risk prioritization. By focusing on driver-level auditing and lightweight scanning, DefenX provides actionable intelligence without causing system lag. The system improves vulnerability visibility and supports efficient remediation, thereby strengthening overall system security.

References

1. https://learn.microsoft.com/en-us/microsoft-365/security/defender-vulnerability-management/

2. https://www.tenable.com/products/nessus

3. https://www.qualys.com/apps/vulnerability-management-detection-response/

4. https://documentation.wazuh.com/current/user-manual/capabilities/vulnerability-detection.html

5. https://www.greenbone.net/en/product-comparison/

6. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf

7. https://www.qualys.com/docs/qualys-vmdr.pdf

8. https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nbacc/ir7747.pdf