International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Agentless Windows Security Scanner: A Modular Framework for Deep Vulnerability Assessment

Shwetha R
School of Computer Science and Engineering
Reva University
Bengaluru, India

Chithranjan S
School of Computer Science and Engineering
Reva University
Bengaluru, India

Prajwal S Bharadwaj
School of Computer Science and Engineering
Reva University
Bengaluru, India

Sujan Rathod
School of Computer Science and Engineering
Reva University
Bengaluru, India

Harsha Shriya
School of Computer Science and Engineering
Reva University
Bengaluru, India

***Abstract*—**Windows-based systems can encounter the problem of agent fatigue: the processes of managing many security agents and auditing dozens of hosts manually are tiring and can result in analyst burnout and vulnerabilities being missed. Hand Vis-Audit patching and configuration is sluggish and prone to error in addition, generates security risk. We in turn suggest Agentless framework: a credentialed, no-agent vulnerability evaluation system. It scans with the Windows standard protocols (WinRM/WMI) to inspect systems with authenticated access and finds systems with low-noise network probes (TCP SYN scans) avoiding intensive version detection or complete handshakes.

The scanner analyses 13 key areas of security; patch (hotfix) status, AMSI settings, UAC settings, PowerShell logging and RDP-related registry entries and provides a comprehensive report in the form of a series of charts in a JSON file without the need to install any software on the target systems. When investigating the network, it will use SYN probes exclusively, thus yielding the low-noise process of network discovery and reducing the chances of generating alerts in SIEM or IDS space. The system is designed with a multi-phase workflow which incorporates network scanning, verified evaluation and central information gathering. A standard Windows 11 subnet diagnosis would be able to reveal such problems as packages not updated and poorly configured but would go unnoticed by monitoring devices because of its friendly appearance to SOC. In general, Agentless 2.0 is beneficial to boost the security of an organization by eliminating endpoint agents and minimizing the manual audit. Concurrently it ensures the reliability of data by checking with the help of SHA-256 integrity check and sustains compliance needs with diverse windows environments.

Keywords*—Agentless Vulnerability Scanning, Windows Security Auditing, Network Exposure Assessment, TCP SYN Scanning, Credentialed Security Inspection, WinRM/WMI Remote Management, Enterprise Vulnerability Management, Security Misconfiguration Detection, PostgreSQL JSONB Security Data Storage.*

I. INTRODUCTION

Defensive (Blue Team) operations are constantly challenged in enterprise Windows environments, especially those with vast and dynamic or legacy resources. The established vulnerability assessment tools that include Nessus and OpenVAS are now industry standards with extensive vulnerability database, well-developed plugins and with robust authenticated scanning. These tools are normally based on a layered architecture which is network discovery, service enumeration, vulnerability detection and reporting. Although successful, they require aggressive network probing and in certain instances endpoint agents, that pose operational challenges.

In agent-based solutions, there is a need to deploy and maintain all endpoints which in turn incurs a high administrative cost and resource usage. This is especially an issue in enterprise environments with legacy applications or limited environments since installing new agents might not be possible, and coverage will not be complete. Agents even when effectively used might put a visible performance burden on CPU, memory and network bandwidth particularly at scale.

On the other hand, agentless scanning approach to traditional tools can be widely based on intensive network interrogation procedures to identify services and versions. These can create much noise on a network, which can lead to an alert in Security Information and Event Management (SIEM) or Intrusion Detection Systems (IDS) and in other instances may cause disruption of the normal business processes.

To eliminate these shortcomings, this paper presents a credential-based, modular, agentless vulnerability scanning model specifically designed to work in a Windows setting. Rather than engaging in intrusive probing or endpoint defining Agents, the proposed system utilizes native windows protocols namely Windows Management Instrumentation (WMI) and windows remote management (WinRM) as a means to inspect the system, authentically and thoroughly, with the help of windows management instrumentation and windows remote management. The approach eliminates the need to install extra software on target hosts in order to support a specific set of ports to transmit data and employ inbuilt system features.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Authentication is done by using Win32 LogonUserW API, which uses generated security token of an authenticated user, to get a query to be run by a scanner in an authorized security context. Such a design will ensure protection and less influence on the activity of the network and systems.

Overall, the solution would provide a rethought approach to test vulnerability of enterprise Windows networks by incorporating the more thorough investigations of authenticated scanning and efficacy and discreteness of agentless model. It can eliminate network noise and eliminate the deployment of agents; reducing agent count scales effortlessly and meets the needs of modern Security Operations Center (SOC); it retains end-to-end visibility and compliance.

II. Related Work

There are a lot of vulnerability scanners, but they vary greatly in their solution. Such tools as OpenVAS or Nessus are at the network level and scan for open ports, and known vulnerabilities on the hosts. Such techniques typically consist of complete TCP handshakes, OS fingerprinting, which has the potential to create detectable network noise and elevates the chances of detection [1]. On the other hand, Microsoft Defender for Endpoint uses an agent model, which uses telemetry to gather continuous data about the endpoints [2]. Although it is effective in monitoring and auditing, it leaves a footprint on the system, which can be easily tampered with.

This work adopts a different approach in emphasizing low noise discovery and less impact of the system. It takes advantage of methods like TCP SYN (half-open) scanning to make it less detectable [3], with authenticated, agentless queries to do a detailed analysis. The proposed framework is Windows-specific, unlike cloud-oriented agentless implementation, which tends to have a passive checks-based implementation, allowing an overview of the registry and policy audit to be thorough without any software being deployed to the target systems. Such stealth-depth equilibrium fills in one of the critical undergaps to the available solutions to the operation of the Blue Team.

III. Proposed System Architecture

The general architecture of the proposed Agentless Windows Security Scanner (Agentless 2.0) is shown in figure 1.

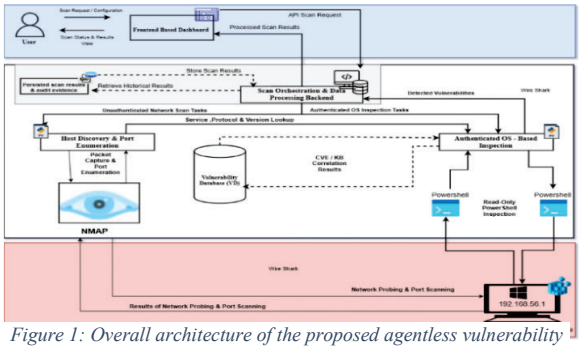

Figure 1: Overall architecture of the proposed agentless vulnerability scanning framework.

This system is structured as a modular vulnerability assessment system that conducts network discovery, authenticated system inspection as well as centralizing security information data aggregation without the need of using software agents on the target end points.
On the upper tier, the system offers a frontend dashboard interface where security analysts can set scan parameters and make assessment requests. These requests are sent to the Scan Orchestration and Data Processing Backend through the REST API endpoints. Backend is the control layer which coordinates scanning activities, aggregates findings, and keeps historical audit logs.

The scanning engine can be used in two main modes namely unauthenticated network scan and authenticated OS based scan. In the initial phase, the system does a host discovery and port enumeration with the network scanning engine based on the SYN-based scanning techniques in the form of Nmap framework. This is done to detect the presence of exposed services and network entry points and reduce network intrusion footprint.
Network reconnaissance results are matched against a centralized database of Vulnerability Database (VD) where the vulnerability and configuration weakness information is stored [2]. Once valid credentials are presented the system goes into the authenticated inspection phase, where more penetrating host-level auditing occurs through PowerShell-based interrogation via Windows management interfaces [3], [4]. The backend processes all results collected and storing them in a persistent database to be used in additional analysis and visualization. Moreover, the packet-level check and analysis can be done with the help of Wireshark tools [5], which enables analysts to authorize network communications when scanning. The end of the processing products is fed into the dashboard, which allows the analysts to examine the vulnerabilities and misconfigurations in the systems.

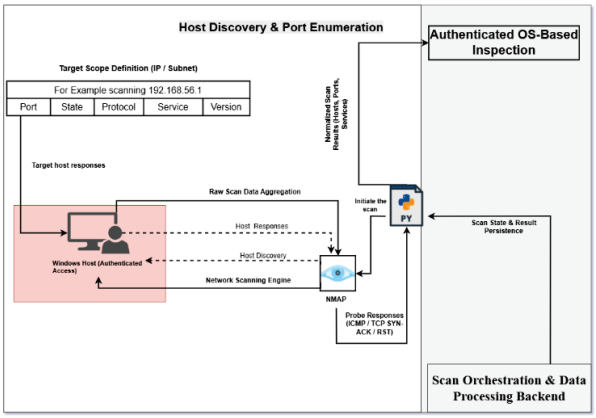

Figure 2: Host discovery and port enumeration process

Figure 2 shows the architecture of Host Discovery and Port Enumeration that are concerned with the discovery of the exposed assets in the network environment.
The layered architecture allows low noise age network discovery along with deep system auditing, which offers a scalable and enterprise-friendly vulnerability assessment framework. This starts with target scope definition, which the analyst defines an IP address range or subnet which will be scanned. The system facilitates individual host scanning as

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

well as subnets discovery operations. After defining the target scope, the orchestration module is a Python-based program that initiates the scanning process.

Host discovery is done by the Network Scanning Engine through ICMP and TCP-based probing systems. In particular, the scanner is based on TCP SYN probe responses (SYN-ACK or RST) to test the availability of the host and open ports. The method permits the scanner to scan services that do not use the entire TCP handshake, minimizing network footprint and reducing the likelihood of causing intrusion detection systems.

The scanner collects raw probe responses of the target Windows hosts and transforms them into organized scan data as shown in Figure 2. Such information as port state, protocol type, service name, and detected version metadata are obtained and stored in structured form.

Results are then sent to the Scan Orchestration and Data Processing Backend and scan states and results are stored. This step is an effective approach to creating a network exposure inventory, which involves the discovery of reachable hosts and external visibility services in the environment.

The result of this step can be used as input to the next step of authenticated inspection so that the system can only conduct a more in-depth vulnerability inspection on the identified and reachable hosts.

A. *Authenticated OS-Based Inspection*

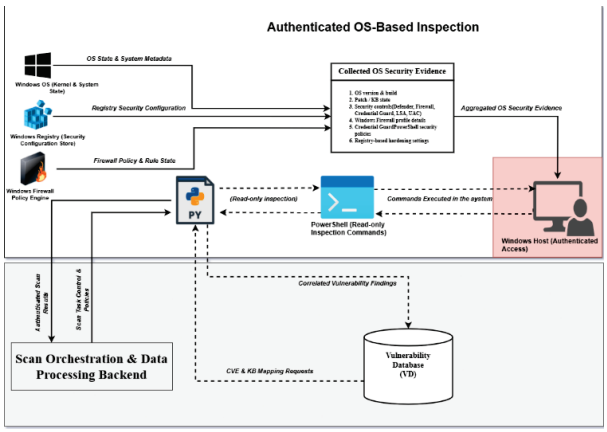

Figure 3: Authenticated OS-based inspection workflow

Figure 3: The architecture of Authenticated OS -Based Inspection, which does a thorough system-level audit against Windows hosts found during Phase 2.

This in contrast to the old-fashioned vulnerability scanners, which depend on network-based probing as their principal tool, the stage uses credentialed inspection methods to obtain security-relevant configuration information directly out of the operating system [6] [7] .
After success in the authentication, the scanning engine gathers data of various important system components such as:

- Windows system and kernel documentation.
- The security configuration of windows registry.
- States of firewall policy and rule.
- Patches and hotfix data are installed.
- Security software status
- System hardening configurations of registry.

The interrogation process is performed by use of PowerShell read-only inspection commands, hence ensuring the system does not alter the target host on analysis itself. These instructions are remotely executed by using authenticated Windows management protocol including WinRM or WMI.

The evidences obtained are combined in the form of a structured data indicating various types of security posture such as operating system configuration, patch management status, credential policies, firewall configurations and registry hardening settings.After data collection the results are passed on to the Scan Orchestration and Data Processing Backend where the results are matched against records in the Vulnerability Database (VD). In this step of correlation, configuration states that are found are mapped to known vulnerabilities and security misconfigurations through CVE and knowledge base mappings.

The end result of this stage is a list of correlated vulnerabilities results, which give us a close-up of the security stature of every scanned host. These are then stored in the database where they are availed on the security dashboard where they can be analyzed.

B. *Secure Data Ingestion*

The results of the scans, as parsed as JSON are injected into a PostgreSQL database using dedicated ingestion scripts. The semi-structured scan data will be stored in a PostgreSQL with the type JSONB [8]. According to PostgreSQL, JSONB is stored in a decomposed binary form to enable rapid processing and can index on JSON columns so that it can easily query and index a JSON data. The ingestion process uses an atomic transaction of hash results acquired by each scan with an SHA-256 hash [9],[10]. In case a matching hash is already found, the script will avoid an insertion (replay protection). The local JSON files are safely erased after insertion to block leakages of data. This combined data is loaded on the frontend dashboard on demand thanks to the backend API.
In summary, the architecture ensures no software agent is ever installed on scanned hosts. The scanner only leverages built-in Windows mechanisms (WinRM/WMI) and controlled network probes, preserving system state. The modular design (separate phases and ingestion) also allows a SOC or administrator to run or skip components as needed.

IV. Methodology & Implementation

The table presents a structured overview of the 13 audited Windows security categories, where each row corresponds to a specific control area, its function, and its security relevance. It can be read by first identifying the category (e.g., patching, services, or policies), then understanding what aspect of the system is being examined and finally interpreting why it matters from a security perspective. Together, the categories provide a layered view of the system's posture—covering updates, configurations, access controls, and potential attacker persistence points. By reviewing the table holistically, analysts can quickly identify misconfigurations,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

missing protections, or suspicious indicators, enabling efficient prioritization of risks and remediation actions.

Table 1: Overview of Endpoint Security Audit Parameters

Category	Description	Security Relevance
Hotfix Audit	Lists installed and missing Windows updates	Unpatched systems are highly vulnerable to known exploits
Software Inventory	Identifies installed applications and versions	Outdated or unknown software introduces vulnerabilities
Firewall Rules	Reviews inbound/outbound firewall configurations	Detects risky rules (e.g., open RDP ports, "allow any")
Service Status Check	Monitors critical services (e.g., Remote Registry, WinRM)	Disabled or unauthorized services may indicate compromise
EDR/AV Health	Verifies presence and status of security tools via Windows Security Center and AMSI	Ensures endpoint protection is active and not tampered with
Audit Policies	Examines Windows audit policies (logon , registry, etc.)	Proper logging is essential for forensic analysis
Persistence Mechanisms	Inspects Run keys, Scheduled Tasks, an d startup entries	Detects unauthorized persistence used by attackers
AMSI / Defender Settings	Checks AMSI providers and Windows Defender configurations	Identifies evasion techniques like disabling security features
UAC & System Policies	Validates UAC and local security policies	Weak policies can reduce system protection
PowerShell History	Extracts command history from PSReadline logs	Helps detect malicious or suspicious command execution
RDP Registry Keys	Checks registry values related to RDP (e.g., DenyTSConnection s)	Detects unauthorized enabling of remote access
Local Accounts & Shares	Gathers information on local users, shares, and permissions	Misconfigurations can lead to privilege escalation
Domain Trust Information	Collects domain trust and network relationship data	Helps identify lateral movement risks

V. Results and Discussion

As a proof-of-concept, we applied the tool to a small Windows 11 subnet. The Network Exposure Scan quickly identified all live hosts and their open ports. In one test (a IP address and ports), we discovered 12 Windows hosts. Using provided admin credentials, the Authenticated Inspection found multiple issues: e.g., several machines missing recent hotfixes, one host had UAC disabled, two had weak firewall rules allowing RDP to "Any" network, and assorted suspicious run-keys (including an orphaned PowerShell startup script). Crucially, because our scan used only SYN probes and no full TCP connects, the existing IDS did not flag any of the probes. Modern IDS can detect SYN scans, but by spacing requests and avoiding aggressive fingerprinting, our scan remained below most alert thresholds. In contrast, a typical full-network Nessus scan on this network would likely have triggered numerous alerts (due to SMB version checking and login attempts). In fact, Nmap's designers note that SYN scans generate TCP packets "with the SYN flag set" and leave the handshake incomplete, making them stealthier. Our case study confirms this: despite scanning dozens of ports, we observed no SIEM or firewall alarms.

From the data ingestion perspective, the scan reports were efficiently stored in PostgreSQL (JSONB fields). This allowed ad-hoc queries via the dashboard: e.g., we could query "Which hosts have any uninstalled security updates?" or "Show any instances of disabled firewall on any host." The ability to index JSON fields made these queries performant . Performance-wise, the agentless scan ran unobtrusively; CPU usage on the server was minimal, and target hosts showed no noticeable load during WMI queries.
To conclude, the tool proved the so-called low noise and high depth trade-off. It discovered a wide range of weaknesses and configuration errors without overloading endpoints or raising alerts. The results demonstrate that agentless credentialed scans are able to provide full auditing (Rather than port checks only) without affecting a stealth profile.

VI. Conclusion

This paper introduced the Agentless Windows Security Scanner, a credential-based framework that is modular and can be used to carry out in-depth vulnerability testing in Windows environments without the use of endpoint agents. The proposed system belongs to the category of systems that can provide both operational stealth and security depth, based on low-noise network discovery based on TCP SYN probing and authenticated host inspection based on WinRM and WMI. This framework does not burden endpoints with probing or persistent agents, which applications of important probes was prevalent in traditional scanners, nor does it generate large numbers of alerts and does not affect the system stability, which is still useful in identifying critical security vulnerabilities.

The applied proof-of-concept illustrated the viability of the approach in detecting missing patches, lax firewall rules, deactivated security controls and suspicious persistence mechanisms across numerous Windows hosts. Structured reporting based on the use of JSON is enhanced with integrity validation via the use of the SHA-256 hash and PostgreSQL storage using a JSONB format, which further increases the reliability and traceability of the system as well as scalable analysis support. Generally, it can be concluded that the results indicated that agentless modular architecture can offer a viable alternative to SOC teams and administrators requiring detailed SP of windows security but not at the expense and complexity of the traditional agent-based systems.

The framework is not a ward of magic. It remains constrained by the quality of credentials, the reachability of a network, and the correctness of the underlying audit checks. Future areas of work will concentrate on building the vulnerability

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

knowledge base, enhancing detection logic over misconfigurations and persistence methods, support of larger enterprise environments, and incorporation of remediation suggestions to transform the system through passive assessment to more automatable networks with defense functions.

REFERENCES

[1] G. Lyon, Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Sunnyvale, CA, USA: Insecure.Com LLC, 2009.
[2] Tenable, "Nessus Vulnerability Scanner," [Online]. Available: https://www.tenable.com/products/nessus
[3] Greenbone Networks, "OpenVAS – Open Vulnerability Assessment System," [Online]. Available: https://www.openvas.org
[4] Microsoft, "Microsoft Defender for Endpoint Overview," [Online]. Available: https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint
[5] Microsoft, "Windows Management Instrumentation (WMI)," [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/wmisdk/wmi-start-page
[6] Microsoft, "Windows Remote Management (WinRM)," [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/winrm/portal
[7] National Institute of Standards and Technology (NIST), "National Vulnerability Database (NVD)," [Online]. Available: https://nvd.nist.gov
[8] Wireshark Foundation, "Wireshark User Guide," [Online]. Available: https://www.wireshark.org/docs
[9] PostgreSQL Global Development Group, "PostgreSQL Documentation: JSON Types," [Online]. Available: https://www.postgresql.org/docs/current/datatype-json.html
[10] National Institute of Standards and Technology (NIST), "Secure Hash Standard (SHA-256)," FIPS PUB 180-4, 2015