Received 1 September 2025
Revised 22 September 2025
Accepted 23 September 2025

Design and implementation of a computer endpoint security baseline verification system based on MLPS 2.0

Fan Miao, YaQiong Xu, ZiYang Wan, YingJie Zhuang, Yang Li and JiaPeng Ren

China Academy of Railway Sciences Corporation Limited, Institute of Computing Technologies, Beijing, China

Abstract

Purpose – Amidst an increasingly severe cybersecurity landscape, the widespread adoption of Xinchuang endpoints has become a strategic imperative. Governments and enterprises have established terminal localization as a critical objective, aiming for comprehensive indigenous replacement through rapid technological iteration. Consequently, Xinchuang systems and Windows platforms are expected to coexist over an extended period. This study seeks to establish an automated verification framework for multi-version operating systems and validate the efficacy of baseline hardening in mitigating security risks.

Design/methodology/approach – Based on the Classified Protection 2.0 framework and relevant national standards for endpoint security, this study proposes an endpoint security baseline verification scheme applicable to multiple operating systems. The scheme addresses divergent security policies and implementation methodologies across heterogeneous environments. It automates the inspection of core baselines, including account password complexity, default shared service status and patch installation status. Furthermore, a comprehensive scoring model is established by incorporating differentiated weights for account security, patch management and log auditing, ultimately generating visualized risk reports to facilitate remediation prioritization.

Findings – This study reveals that baseline configuration serves as the fundamental prerequisite in endpoint security practices. Through a scalable detection engine and quantitative scoring model, the system can promptly identify and remediate potential risks, thereby reducing the attack surface and mitigating intrusion risks. However, on certain domestic chip architectures, compatibility issues persist in detecting specific configuration items. Further improvement in hardware–software co-adaptation for domestic platforms is required to advance the development of localized security protection systems.

Originality/value – Through in-depth research on security baseline configurations across multiple operating systems, this study implements an automated and visualized baseline verification methodology. This approach significantly strengthens the security posture of domestic operating systems and supports the establishment of a more robust, national-level cybersecurity defense framework.

Keywords Windows security baseline, Endpoint security, Baseline inspection, Xinchuang system security baseline

Paper type Research article

1. Introduction

With the accelerated digital transformation of enterprises, endpoint devices have emerged as the central cyber warfare battleground. Based on IBM's "2023 Cost of a Data Breach Report," the average remediation cost for ransomware attacks has surged to $4.86 million (IBM Security, 2023), while lateral movement attacks achieve a success rate as high as 67% within enterprise networks. These threats often originate from "vulnerable entry points" caused by

Funding: This work is supported by scientific research projects of China Academy of Railway Sciences Co., Ltd. (grant no. 2024YJ117).

Railway Sciences
Vol. 5 No. 1, 2026
pp. 136-152
Emerald Publishing Limited
e-ISSN: 2755-0915
p-ISSN: 2755-0907
DOI 10.1108/RS-08-2025-0031

flawed endpoint security configurations. Beyond direct financial losses, such vulnerabilities frequently trigger systemic risks including data breaches and operational disruptions. Against this backdrop, security baselines — serving as a core measure to minimize attack surfaces — have been integrated into key practices of mainstream global cybersecurity frameworks. By implementing predetermined system configuration standards, security baselines ensure terminals in a secure and controllable state, thereby effectively disrupting initial infiltration and lateral movement pathways in attack chains (Joint Task Force, 2020). Concurrently, compliance frameworks such as Classified Protection 2.0 Level 3 Requirements and ISO 27001 explicitly mandate enterprises to establish and continuously validate endpoint security baselines. Related configuration items have become mandatory audit metrics in security audits (National Information Security Standardization Technical Committee, 2019, 2024a, b). Consequently, security baseline management is not merely a technical task but a strategic imperative for regulatory compliance and legal risk mitigation.

Over the years, both academia and industry have conducted extensive research on automated security baseline verification. Microsoft's SCAP (Security Content Automation Protocol) enables rapid security configuration and scanning for Windows systems through standardized templates. However, its coverage is primarily limited to official generic policies, making it difficult to flexibly adapt to enterprises' customized security requirements (Gan, Cybersecurity and Informatization). Within the open-source community, lightweight solutions based on Ansible can validate specific configuration items via registry queries and service status checks. Nevertheless, their fragmented implementation leads to inconsistent verification standards and universally lacks compliance quantification capabilities (GitHub Community, 2022). Domestic research primarily focuses on Classified Protection 2.0 compliance. In 2021, Zhang *et al.* developed a detection tool integrating password policy checks mandated by Classified Protection 2.0, yet failed to automate verification of critical security requirements such as log auditing (Zhang & Chen, 2021). In 2022, Li *et al.* proposed a Group Policy Object (GPO)-based baseline hardening framework, but it still demonstrated cross-version compatibility limitations across Windows systems (Li & Wang, 2022). In 2025, Gan *et al.* proposed a security policy and configuration scheme for the Xinchuang operating system, but the operation is complex and lacks systematic management and control measures (Gan, Sun, Ling *et al.*, 2025). Overall, existing methods still demonstrate significant deficiencies in detection efficiency, coverage completeness, and quantitative assessment capabilities.

This study proposes an automated framework for verifying and testing endpoint security baselines across diverse versions and platforms. The framework systematically integrates compliance requirements from China's Multi-Level Protection Scheme 2.0 (MLPS 2.0) with practical attack surface reduction objectives. By designing an extensible detection engine and a quantitative scoring model, it systematically addresses issues inherent in traditional approaches—such as high reliance on manual efforts, inconsistent standards, and insufficient coverage—providing enterprises with an integrated endpoint security management solution that offers both compliance audit value and practical protection efficacy.

2. Theoretical foundations

2.1 Security baseline definition and classification

A security baseline constitutes a standardized set of system configurations designed to enhance cybersecurity resilience. Its core objective is to eliminate attack surface exposure caused by configuration oversights by normalizing dimensions such as account policies, access control, and log management. According to NIST SP 800-171 and Microsoft security practices, terminal baseline configurations can be classified into three categories: Account Policies, Access Control, and Log Management. These classifications directly map to requirements specified in Baseline for Classified Protection of Cybersecurity as illustrated in Table 1.

Table 1. Mapping table: MLPS 2.0 level 3 general endpoint requirements

Item	Requirement	Alignment
Secure computing environment	Tamper-proofing and disabling for devices or storage media	Host peripherals
	Identity authentication	Account policies
	Access control	Account policies, Host files, Security configurations
	Security auditing	Audit policies
	Intrusion prevention	Host processes, Host services, Network ports, System patches, Anti-malware, Security configurations
	Trusted verification	Security configurations, System version
Security communication network	Data integrity	Security configurations, Log storage
	Data confidentiality	Security configurations, Log storage
	Vulnerability and risk management	System patches, System version, Host software

Source(s): Baseline for classified protection of cybersecurity

As a cornerstone of China's cybersecurity classified protection framework, the Multi-Level Protection Scheme 2.0 (MLPS 2.0) mandates comprehensive security protection measures for information systems. For example, the Identity Authentication clause requires strict adherence to password complexity baseline standards, while the Security Auditing clause imposes a minimum log retention period of 180 days. These requirements are operationally implemented through standardized security baseline configurations. This mapping relationship provides organizations with a dual-perspective theoretical foundation, enabling simultaneous achievement of technical protection and regulatory compliance objectives.

2.2 Key metrics for security baseline verification

Security baseline verification shall encompass three core components: account security, access control configurations, and log management, Quantitative metrics are designed to facilitate a precise assessment. At the account security level, verification must validate password complexity, account lockout thresholds, and password expiration periods. Detection of blank passwords or weak passwords shall be classified as a critical vulnerability. Regarding service configuration, the audit focuses on high-risk service states and default shared resources; failure to disable such services significantly increases lateral movement risks. For log management, it ensures audit policies cover critical operations, with Event Viewer verifying log retention periods is no less than 180 days to prevent traceability failures due to log truncation.

To comprehensively evaluate endpoint compliance, this study designs a weighted scoring model wherein differentiated weights are assigned to account security, service configuration, and log management. Endpoints scoring below the established threshold of 80 points require immediate remediation. This model transforms abstract configuration rules into quantifiable metrics, thereby providing a theoretical foundation for the development of automated tools and risk-tiered governance.

3. Scheme design

3.1 Functional design

The security baseline verification management platform supports multiple operating systems and seamlessly integrates with self-developed agents to execute endpoint security baseline assessments. By employing parallelized data collection and resource-efficient analytical methods, the platform deploys baseline compliance commands to individual monitoring

endpoints, then interprets their submitted baseline evaluation reports. It dynamically generates an OS-adaptive scoring model to produce holistic verification results and actionable remediation guidance. The system adopts a modular three-layer architecture (user interface (UI) layer, business logic layer, and data storage layer), as shown in Figure 1.

Self-developed Agent is primarily responsible for the protocol-based classification and parsing of baseline tasks dispatched by the management platform and the subsequent execution of directive-driven scans. It caches collection parameters, packages and organizes the scan results based on data type, and transmits the consolidated data package to the management platform.

The UI presentation layer, implemented via a web interface, provides visual representation of system data and human-machine interaction. Users leverage this interface to configure baseline templates, with the system supporting adaptation of differentiated templates based on machine models and operating system versions. Integration with terminal management systems enables assignment of appropriate baseline template types to terminals historically flagged as high-risk or medium-risk levels. The verification statistics module displays scan results in multiple dimensions according to scoring models, assisting users in holistically assessing overall network security posture and vulnerabilities. The user management page provides cascaded configuration capabilities for accounts and probe terminals, supporting hierarchical access to device information and baseline results based on account privileges. The audit logging module records user operations to establish audit trails for future investigations.

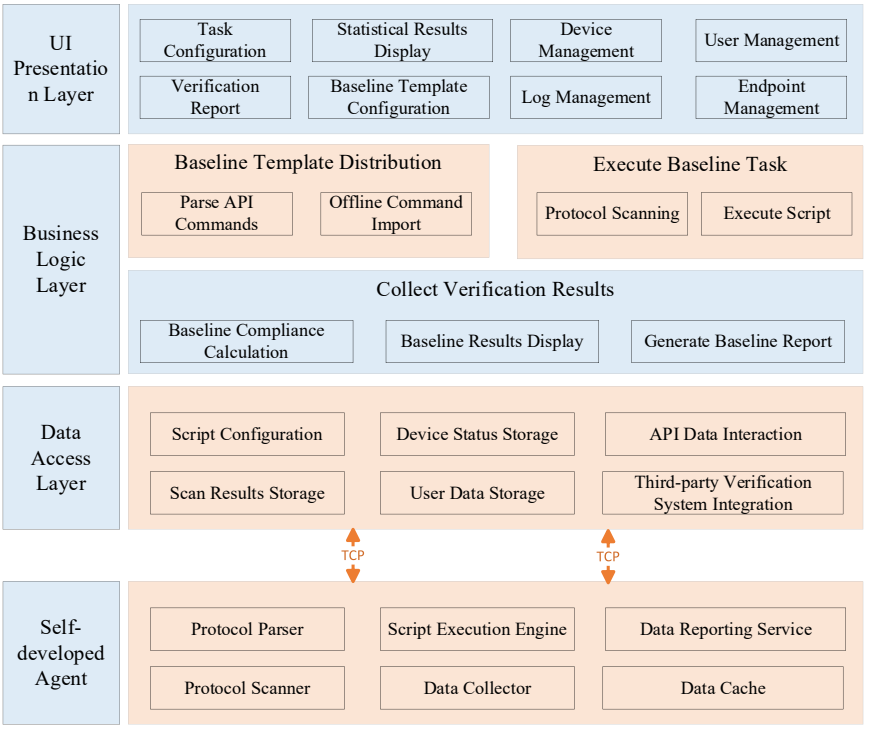

Figure 1. Functional architecture of the security baseline verification management platform. Source: Authors' own work

Business logic layer processes core business including parsing scan instructions, executing baseline check scripts, and collating/returning scan results. The system invokes corresponding scanning scripts based on instructions and generates baseline verification reports against built-in scoring criteria. The verification statistics functionality supports multi-dimensional filtering and analysis of scan results to identify critical vulnerabilities within the security posture. Statistical metrics encompass compliance rates, total inspected devices, required verification quantities, and verification item counts.

Data access layer manages data exchange and storage. The storage module handles persistent retention of scripts, device information, user data, and scan results. The API interface module processes data communication tasks including script distribution, scan result reporting from probe nodes, and dissemination of non-compliant item notifications. Standardized interface templates enable interoperability with third-party systems from different vendors, establishing a unified security protection chain.

3.2 Terminal verification process design

Endpoint security baseline verification encompasses three sequential phases: Preparation, Assessment, and Report Generation. Windows and Linux exhibit significant implementation divergences due to fundamental differences in operating system architectures, command frameworks, and security mechanisms. Windows verification depends on proprietary closed-system interfaces and graphical configuration utilities, requiring adaptation to its unique binary storage mechanisms and permission structures. Conversely, Linux verification relies on open-source text-based configurations and command-line tools, with its implementation predominantly leveraging file parsing logic and script-driven automation processes.

3.2.1 Windows. Native Windows system commands serve as the core detection tools in this phase, enabling direct invocation of system APIs to retrieve critical configurations such as registry entries and service states, while ensuring cross-version compatibility. For example, invoking the Get-Service command via scripts facilitates rapid scanning of high-risk services such as Telnet and SNMP (Microsoft, 2025). Additionally, the management endpoint must preconfigure inspection templates that integrate detection rules for account policies, service configurations, and access controls, defining assessment thresholds for varying risk levels. For instance, systems with enabled Telnet services or blank-password accounts are flagged as critical risks, while log retention periods of longer than 90 days that are still insufficient are classified as low-risk.

A phased assessment strategy sequentially scans account policies, access controls, and log audits. Account policy detection employs commands to read the registry key HKEY_LOCAL_MACHINE\SECURITY\Policy\PolAdtEv, extracting parameters such as password complexity and account lockout thresholds, supplemented by parsing security policy templates (via secedit/export/cfg command) to validate password expiration settings (Microsoft, 2023). Access control assessment focuses on high-risk services and shared resources, using net share to verify closure of default shares. Log auditing analyzes local security policy files to extract audit policies and log retention configurations, enabling comprehensive operational monitoring of event logs (Li & Xia, 2018).

Risk items are categorized per predefined rules to generate remediation guidance. Critical risks (e.g. enabled Telnet services, blank-password accounts) require immediate remediation. Medium risks (e.g. weak passwords, open default shares) permit resolution within 72 h. Low risks warrant configuration enhancements. Reports detail risk descriptions with actionable remediation commands, displaying verification results and remediation status via the management endpoint. To validate remediation effectiveness, the endpoint initiates re-verification tasks, establishing a closed-loop “detection-remediation-validation” cycle.

3.2.2 Xinchuang systems. Baseline verification for Xinchuang terminals requires developing OS-specific compatibility tools that invoke security interfaces (e.g. parsing/etc/security/policy.conf) to automate inspection of critical configurations (National Information

Security Standardization Technical Committee, 2017). Policy templates strictly comply with China's Classified Protection 2.0 requirements, encompassing Chinese cryptographic algorithms, three-powers-separation privilege models, and domestic middleware controls. This ensures compliance with national information security standards from initial system deployment. Detection tools must support heterogeneous hardware architectures (LongArch, Kunpeng, and Phytium architectures), guaranteeing comprehensive coverage.

A tiered scanning strategy sequentially examines account policies, service configurations, and log audits. Account policy verification parses/etc/pam.d/system-auth to validate password complexity rules, minimum length, and expiration policies (Bai, Tian, & Li, 2024). Service configuration assessment employs systemctl to detect unauthorized database services or exposed high-risk daemons. Log auditing analyzes rsyslog.conf to ensure compliance with audit trail standards (GB/T 22239-2019), enhancing system traceability.

Critical risks, such as disabled cryptographic transmission protocols compliant with Chinese national standards and accounts with passwordless authentication configurations, trigger immediate blocking alerts and require urgent remediation within 24 h. Medium risks, including the absence of off-site log backups and non-compliant password policies, necessitate resolution within 72 h. Low risks, exemplified by insufficient log retention periods and misconfigured service parameters, warrant systematic optimization within defined maintenance windows. The reporting framework integrates with the China National Vulnerability Database (CNNVD) to generate tailored remediation plans aligned with national standards (NSFOCUS, 2023). Post-remediation verification procedures ensure closure of the security "detection-remediation-validation" lifecycle, achieving full compliance with MLPS 2.0 requirements.

3.3 Scoring model and compliance index

To evaluate endpoint security posture, this study introduces a mathematical assessment framework. The initial phase requires delineating Key Performance Indicators (KPIs) for target information system components, structured across six tier-1 dimensions: system configuration, vulnerabilities and patching, access control, logging and auditing, malware countermeasures, and data security, as detailed in Table 2.

Each tier-1 dimension is decomposed into 3–5 concrete quantifiable metrics, with explicitly defined baseline requirements and standardized assessment methodologies. Through quantitative scoring of individual metrics and hierarchical weighting aggregation, a comprehensive endpoint security posture score is generated.

3.3.1 System security configuration. System hardening is achieved through the implementation of critical security configurations, including password policy compliance, account lockout mechanisms, strict adherence to the principle of least privilege, and the disabling of unnecessary services. The corresponding baseline values and respective evaluation methodologies for each configuration item are detailed in Table 3.

The quantitative assessment methodology for each sub-metric is structured as follows:

Table 2. Tier-1 metrics and corresponding weights

Tier-1 metrics	Weight	Description
System configuration	0.20	Security of fundamental operating system configurations
Vulnerabilities and patching	0.20	Timeliness of vulnerability remediation
Access control	0.15	Identity authentication and privilege management
Logging and auditing	0.15	Traceability of security events
Malware countermeasures	0.15	Defense capabilities against malware/threats
Data security	0.15	Storage and transmission of sensitive data

Source(s): Authors' own work

Table 3. Sub-metrics for system security configuration

Sub-metrics	Baseline value	Assessment methodology
Password policy compliance	Minimum password length of 8 characters; complexity must include uppercase letters, lowercase letters, digits, and special characters	Inspect system password policy configuration
Account lockout policy	Account locked after 5 consecutive failed login attempts	Verify account lockout threshold and unlock time settings
Principle of least privilege	Regular user accounts lack administrative privileges	Audit user group privilege assignments
Unnecessary services	High-risk services (e.g. Telnet, SMB sharing) are disabled	Scan active port and service list

Source(s): Authors' own work

(1) Password Policy Compliance

(2) The score is directly proportional to the compliance rate, calculated by subtracting 25 points for each unmet requirement from a baseline of 100 points. For example, one unmet requirement yields a score of 75 points.

(3) Account Lockout Policy Enforcement

(4) Scores are assigned based on compliance with the following security levels:

- 100 points for properly configured lockout mechanisms
- 50 points for configurations permitting excessive access attempts
- 0 points for systems lacking lockout protections

(5) Principle of Least Privilege Adherence

(6) The score is determined by deducting 50 points for each overprivileged account identified, starting from a perfect score of 100 points. For instance, two overprivileged accounts result in a score of 0 points.

(7) Superfluous Service Deactivation Rate

(8) The score is linearly scaled to the deactivation rate, where a 90% or higher rate yields 100 points. The score decreases by 10 points for every 5-percentage-point reduction below this threshold. For example, an 85% deactivation rate results in a score of 90 points.

3.3.2 Vulnerability and patch management. The assessment framework for vulnerabilities and patch management is operationalized through three pivotal sub-metrics: the critical vulnerability remediation rate, the timeliness of patch deployment, and vulnerability scan coverage. The corresponding baseline values and standardized evaluation methodologies governing each sub-metric are exhaustively delineated in Table 4.

The quantitative assessment methodology for each sub-metric is structured as follows:

(1) Critical Vulnerability Remediation Rate

(2) The score is directly proportional to the remediation rate, calculated by multiplying the rate by 100. For example, a remediation rate of 95% yields a score of 95 points.

(3) Patch Deployment Timeliness

Table 4. Sub-metrics for vulnerability and patch management

Sub-metrics	Baseline value	Assessment methodology
Critical vulnerability remediation rate	>95% critical-level vulnerabilities patched	Patched critical vulnerabilities/total critical vulnerabilities (monthly)
Patch deployment timeliness	Critical patches installed within 48 h of release	Elapsed time between patch release and installation timestamps
Vulnerability scan coverage	100% endpoint coverage	Scanned endpoints/total managed endpoints

Source(s): Authors' own work

(4) Scores are assigned based on compliance with the following deployment time intervals:

- 100 points for deployments completed within 48 h.
- 80 points for deployments completed between 48 and 72 h.
- 50 points for deployments completed between 72 and 120 h.
- 0 points for deployments exceeding 120 h.

(5) Vulnerability Scan Coverage

(6) The score is linearly scaled to the coverage rate, derived by multiplying the rate by 100. For instance, a coverage rate of 90% results in a score of 90 points.

3.3.3 Access control. The access control assessment framework is structured around three pivotal sub-metrics: multi-factor authentication (MFA) enforcement rigor, adherence to the principle of least privilege, and comprehensiveness of access log recording. The corresponding baseline security thresholds and standardized evaluation protocols for each sub-metric are exhaustively delineated in Table 5.

The quantitative assessment methodology for each sub-metric is structured as follows:

(1) Multi-Factor Authentication (MFA) Enforcement Rigor

(2) The score is directly proportional to the deployment adoption rate, calculated by multiplying the rate by 100. For example, an adoption rate of 80% yields a score of 80 points.

(3) Adherence to Least Privilege

(4) Scores are assigned based on compliance with the following criteria:

Table 5. Sub-metrics for access control

Sub-metrics	Baseline value	Assessment methodology
Multi-factor authentication (MFA) enforcement rigor	Enabled for critical systems	Verify MFA enforcement for access to critical systems (e.g. VPN, email, servers)
Adherence to least privilege	Zero redundant permissions	Conduct sample-based audits (e.g. 10% of endpoints) for excessive user privileges
Access log comprehensiveness	Logs for login/logoff/ permission changes	Verify enablement of system audit policies for relevant event types

Source(s): Authors' own work

- 100 points if no over-privileged accounts exist.
- A deduction of 30 points per over-privileged account identified.

(5) Access Log Recording Comprehensiveness

(6) The score is linearly scaled to the deployment adoption rate, derived by multiplying the rate by 100. For instance, an adoption rate of 80% results in a score of 80 points. A score of 0 is assigned if no logging mechanism is implemented.

3.3.4 Logging and auditing. The logging and auditing assessment framework is operationalized through three core sub-metrics: log integrity, log retention period compliance, and anomalous event alerting efficacy. Corresponding baseline values and standardized evaluation methodologies for each sub-metric are systematically delineated in Table 6.

The quantitative assessment methodology for each sub-metric is structured as follows:

(1) Log Integrity

(2) Scores are assigned based on the following criteria:

- 100 points if no evidence of log tampering is detected.
- 0 points if any tampering is identified.

(3) Log Retention Period Compliance

(4) Scores are assigned based on compliance with the retention threshold:

- 100 points for retention periods ≥90 days.
- A deduction of 50 points per 15-day reduction below the 90-day threshold.

(5) Anomalous Event Alerting Efficacy

(6) Scores are assigned based on the following criteria:

- 100 points for 100% alerting accuracy (no missed detections).
- A deduction of 30 points per missed detection.

3.3.5 Malicious code protection. The malicious code protection framework is operationalized through three pivotal sub-metrics: anti-virus coverage, real-time monitoring adoption rate, and signature update frequency compliance. Corresponding baseline security thresholds and standardized evaluation methodologies for each sub-metric are exhaustively delineated in Table 7.

Table 6. Sub-metrics for logging and auditing

Sub-metrics	Baseline value	Assessment methodology
Log integrity	No evidence of tampering	Verify log integrity by validating cryptographic hash values
Log retention period	Minimum 90-day retention	Audit log server configuration to validate retention policy settings
Anomaly alert triggers	Alerts for failed logins/ permission changes	Perform functional tests to validate alert generation for specified security events

Source(s): Authors' own work

Table 7. Sub-metrics for malicious code protection

Sub-metrics	Baseline value	Assessment methodology
Anti-virus software coverage	Installation on all endpoints	Full-system scan verification
Real-time monitoring adoption rate	Enablement of file and registry monitoring	Inspection of real-time protection mode in anti-virus software
Signature update frequency compliance	Updates within 24 h	Check of last signature update timestamp

Source(s): Authors' own work

The quantitative assessment methodology for each sub-metric is structured as follows:

(1) Anti-Virus Software Coverage

(2) Scores are calculated through direct linear scaling of the deployment coverage rate. The final score corresponds to the percentage of endpoints with installed anti-virus software, multiplied by 100. For instance, a verified coverage rate of 80% yields a score of 80 points.

(3) Real-Time Monitoring Adoption Rate

(4) Scoring follows a proportional allocation method based on the implementation rate of real-time monitoring capabilities. The score is derived by multiplying the percentage of endpoints with enabled monitoring features by 100. As an example, an 80% adoption rate results in a score of 80 points.

(5) Signature Update Frequency Compliance

(6) Score assignment employs a time-based tiered framework:

- 100 points: Signature updates occurring within 24-h intervals
- 80 points: Updates completed within 24 to 48-h timeframes
- 50 points: Updates performed within 48 to 72-h windows
- 0 points: Updates exceeding 72-h thresholds

3.3.6 Data security. The data security assessment framework is operationalized through three core sub-metrics: encryption-at-rest for sensitive data, encryption-in-transit, and backup effectiveness verification. Corresponding baseline security thresholds and standardized evaluation methodologies for each sub-metric are systematically delineated in Table 8.

The quantitative assessment methodology for each sub-metric is structured as follows:

Table 8. Sub-metrics for data security

Sub-metrics	Baseline value	Assessment methodology
Sensitive data encryption at rest	Encryption applied to databases and documents	Inspection of filesystem and database encryption configurations
Data transmission encryption	Implementation of transport encryption protocols	Network packet capture analysis to verify encryption protocols
Data backup effectiveness	Demonstrated backup recoverability	Simulation of data loss scenarios with backup restoration validation

Source(s): Authors' own work

(1) Encryption-at-Rest Coverage

(2) Scoring follows a graduated deduction model based on encryption implementation rates:

- Maximum score (100 points) requires at least 90% encryption coverage across all sensitive data repositories
- A deduction of 10 points is applied for every 5-percentage-point deficiency below the 90% threshold

(3) Encryption-in-Transit Compliance

(4) The security score is determined through proportional scaling of observed encryption rates during packet analysis. The final score corresponds directly to the percentage of encrypted packets, where an 80% encryption rate yields a standardized score of 80 points.

(5) Backup Integrity Verification

(6) Evaluation utilizes a sliding-scale scoring system based on demonstrated restoration efficacy:

- Optimal score (100 points) requires at least 95% successful restoration rate in validation testing
- A linear deduction of 5 points occurs for every 2-percentage-point degradation below the 95% benchmark
- Representative example: A 93% restoration success rate corresponds to an evaluated score of 90 points

To standardize and unify security assessments, the endpoint security baseline transforms discrete configuration items into a unified compliance score through layered weighted summation.

$$S_i = \sum_{j=1}^{n_i} \left(S_{ij} \times w_{ij}\right) \tag{1}$$

n_i:Number of sub-metrics under the i-th primary metric; *wij*:Weight of the j-th sub-metric under the i-th primary metric, satisfying $\sum wij = 1$

$$S = \sum_{i=1}^{6} \left(S_I \times w_I\right) \tag{2}$$

Where w_i denotes the weight of the i-th primary metric (e.g. $W_1 = 0.25$, $W_2 = 0.20$, etc.),

Each security configuration item on endpoint devices (such as account password strength, patch update status, shared service configurations) is assigned a compliance score based on its security posture. These scores reflect whether configuration items meet predetermined security standards, while incorporating their priority levels and impact on system security. Subsequently, these scores undergo weighted computation through a comprehensive scoring model to derive a unified compliance index. This compliance index effectively characterizes the overall security posture of systems while enabling administrators to dynamically evaluate the effectiveness of baseline hardening measures by tracking its evolution. Particularly during long-term security monitoring and improvement initiatives, the index provides system administrators with a concise tool to assess whether security protection measures are effective, facilitating timely adjustments and optimization. Through this methodology, security assessments gain enhanced transparency and operational feasibility, simultaneously establishing quantitative evidence for dynamically tracking baseline hardening efficacy.

4. Experimental process and experimental result analysis

To validate the effectiveness and cross-platform applicability of the proposed terminal security baseline verification system, this study conducted empirical experiments on enterprise-grade operating environments. Quantitative metrics—including scan duration, coverage breadth, and false-positive rates—were rigorously evaluated through comparative analysis against widely adopted open-source tools. The assessment leveraged established benchmarking frameworks, including OpenSCAP, Lynis, InSpec, and Wazuh, to ensure methodological robustness.

4.1 Experimental environment

The test configuration consists of 5 terminal devices, including: 2 Windows platforms (1 Windows 10 and 1 Windows 11, both equipped with Intel Core i5-12400K CPUs) 3 domestic OS platforms (1 ARM-based KylinOS, 1 ARM-based UnionTech OS, both with Phytium D3000 CPUs; and 1 x86-based UnionTech OS with a Hygon 3350 CPU) All terminals feature 16GB RAM and 1TB SSD storage. Due to inherent security architecture differences between Windows and Linux systems, the detection item count varies: 69 items for Windows platforms and 58 items for domestic OS platforms (refer to Table 9).

4.2 Experimental process

(1) Security Tool Deployment and Configuration

Open-source security scanning tools and autonomously developed agents were deployed on the test endpoints. Specifically, OpenSCAP, Wazuh, and the self-developed agent provided compatibility with Windows, Kylin, and UOS operating systems. In contrast, Lynis was evaluated exclusively on Kylin and UOS platforms due to the absence of official native Windows support. For InSpec, Ruby runtime environment preinstallation was required on Kylin and UOS systems, whereas Windows deployments necessitated installation via the Chocolatey package manager prior to scan execution and results verification.

(2) Baseline Configuration

OpenSCAP and InSpec establish detection benchmarks by integrating officially maintained MLPS 2.0 extension packs but require explicit Profile specification within their command-line interfaces. Wazuh necessitates client-side configuration modifications, adding the <rule_file>etc/rules/gb22239-2019.xml</rule_file> directive with the <ruleset> tag, to enable support for MLPS 2.0 compliance items. As Lynis inherently lacks native MLPS 2.0 compliance modules, it requires post-processing of its lynis-report.dat output: detection items are extracted via regex pattern matching and subsequently mapped to the control items enumerated in Appendix A of (GB/T 22239-2019) MLPS 2.0 to align the assessment results with the baseline requirements.

(3) Verification of False Positive Rate

Conducted rigorous verification of the critical control item “Prohibit direct SSH root login” as mandated by MLPS 2.0 security standards. The study documented false positives (erroneously

Table 9. Experimental environment

Endpoint	Control items	Testing tools
ARM + kylin	58	OpenSCAP, Lynis, InSpec, Wazuh, self-developed agent
Windows 10	69	
ARM + UOS	58	
Windows 11	69	
X86+UOS	58	

Source(s): Authors’ own work

flagging compliant configurations as violations) and false negatives (failing to identify non-compliant configurations) across different tools, along with their root causes.

(4) Security Scan Execution

The experiment adopted an "individual tool-individual system" isolated scanning paradigm to avoid potential system resource contention and result interference caused by concurrent multi-tool scanning. Prior to each scanning task, the hash values of critical configuration files on the target system were verified through tool validation, while system commands were employed to record process states and network port occupancy, establishing a baseline document of the initial system state to ensure consistent environmental conditions for each scan. For each combination of tool and target system, three complete scanning cycles were consecutively executed, with the average values of metrics such as time consumption and number of detected items from these three scans taken as valid data to mitigate the impact of random errors on experimental results.

4.3 Experimental results and analysis

To achieve precise validation of open-source tool detection efficacy, a stratified random sampling strategy was employed, selecting 20% of control items from each system scan result as verification samples. The ground-truth configuration status was obtained through command-line configuration queries or direct system configuration file inspection. These data were cross-referenced with tool-generated scan results to determine detection accuracy.

In terms of coverage, the experimental results are shown in Figure 2. OpenSCAP demonstrated excellent performance on Windows platforms, achieving a coverage rate of 92–93%. However, it exhibited significant performance degradation in domestic KylinOS/UnionTech OS environments, with a minimum coverage rate of 74.7%. InSpec achieved optimal cross-platform consistency through its flexible detection framework, with a coverage rate of 86–91%. In contrast, Lynis and Wazuh struggled to meet the requirements of complex environments due to functional limitations and platform compatibility issues, respectively. The proposed in-house solution maintained stable coverage across all test environments, ranging from 78% to 85%, and particularly demonstrated comprehensive superiority over comparable open-source tools on domestic platforms. This highlights its robust cross-platform adaptability and localized optimization advantages.

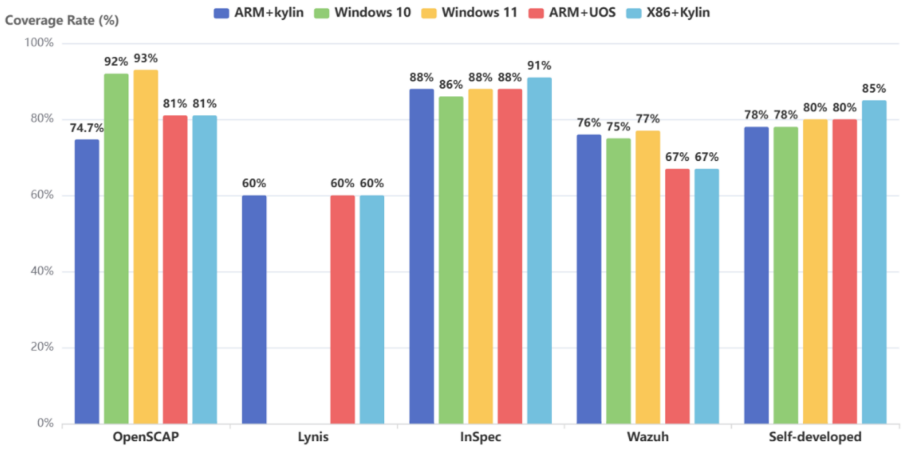

Figure 2. Coverage rate. Source: Authors' own work

Analysis of the false positive rate data in Figure 3 reveals significant platform dependency in the detection accuracy of endpoint security tools. InSpec demonstrates optimal and stable false positive control (1–3%), exhibiting consistent performance across platforms. Open-source tools generally experience higher false positive rates on domestic KylinOS/UnionTech OS platforms: OpenSCAP increases from 3% on Windows to 7–8% in domestic environments, while Wazuh reaches a peak of 15% on X86+Kylin platforms, indicating a lack of adaptation in existing open-source solutions to China's domestic IT ecosystem. The proposed in-house solution achieves the best accuracy on domestic platforms (4-6%)—although its Windows false positive rate (8%) is higher than OpenSCAP's—while maintaining the most well-balanced cross-platform performance overall.

Scan duration directly impacts the applicability of tools in production environments. As shown by the scan time data in Figure 4, significant variations in efficiency exist among the evaluated tools. Lynis demonstrates optimal efficiency on supported Linux platforms, while InSpec's interpreted execution nature results in the longest scan times, making it unsuitable for

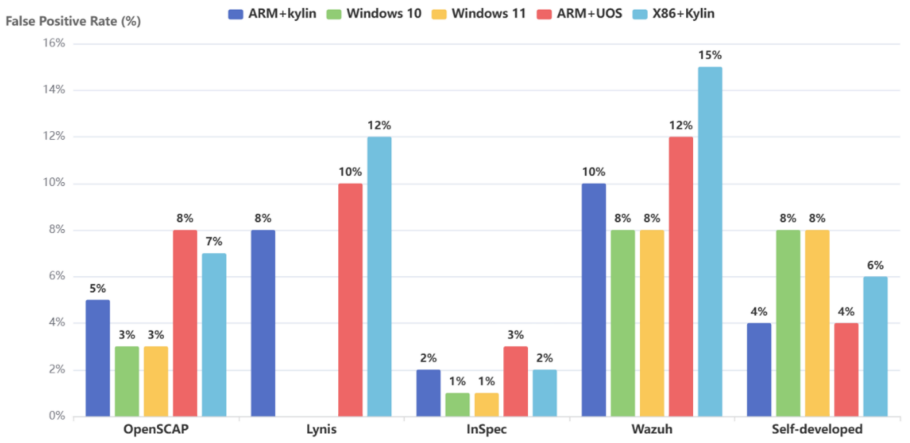

Figure 3. FPR. Source: Authors' own work

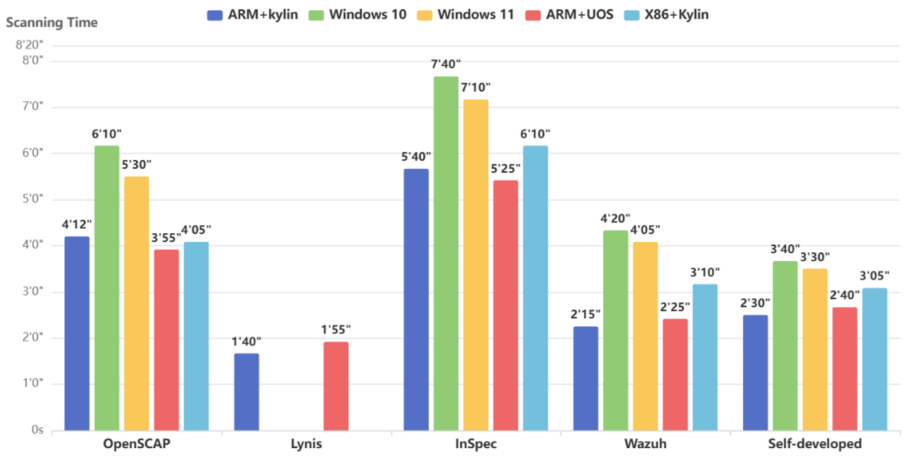

Figure 4. Scanning time. Source: Authors' own work

latency-sensitive scenarios. OpenSCAP and Wazuh perform in the mid-range. The proposed in-house solution strikes the optimal balance between efficiency and effectiveness.

To validate the compliance index in the scoring model, an additional Windows 10 device (designated as Windows10-1) was added to the previous test environment. This device was deliberately configured with multiple security deficiencies to simulate potential security risks arising from configuration oversights or non-compliant operations. The experimental results are shown in the table below.

The experimental results in Table 10 show that the compliance index of Windows 11 is significantly higher than that of the Windows 10 client system, indicating that the newer operating system is stricter in terms of default security baseline design and built-in security control mechanisms, offering a clear advantage in overall security. The score differences between the two Windows 10 devices reflect variations in the security status of endpoint devices. This scoring model can effectively identify security level degradation caused by configuration oversights or non-compliant operations, demonstrating good detection sensitivity and practical applicability.

The aforementioned experimental results reveal a clear performance trade-off among existing open-source tools concerning coverage integrity, detection accuracy, and operational efficiency. OpenSCAP delivers superior coverage on Windows yet underperforms on domestic platforms; Lynis offers high efficiency but is hampered by significant coverage gaps; InSpec achieves high accuracy at the cost of low efficiency; and Wazuh demonstrates mediocre performance across all measured metrics without a distinct leading advantage. In contrast, the proprietary agent developed in this study demonstrates significant advantages in coverage integrity, detection accuracy, and lightweight operation, achieving full compliance with MLPS 2.0 standard. It not only provides comprehensive support for MLPS 2.0 controls and effectively suppresses false positive rates but also operates without requiring additional installation dependencies, maintains minimal resource consumption, and exhibits superior adaptability to domestic IT environments, making it highly suitable for large-scale deployments. Furthermore, the compliance index derived from the scoring model accurately reflects endpoint security postures. These characteristics collectively establish the proprietary agent as a robust and efficient technical solution for implementing the MLPS framework, with its advantages being particularly pronounced in domestic and resource-constrained enterprise environments. Future work will focus on exploring the agent's application potential in other cybersecurity domains and further optimizing its performance to address a broader spectrum of enterprise security requirements.

5. Conclusion

By leveraging an extensible and adaptable detection engine integrated with a quantitative scoring model, organizations can achieve significant improvements in operational efficiency and the precise identification of high-risk vulnerabilities. This approach provides a scientific basis for establishing secure and controllable endpoint environments. However, compatibility challenges persist when detecting certain security configuration items on endpoint platforms equipped with domestic chips. Future work will focus on research into adaptation for domestic hardware architectures, aiming to promote the standardization and performance optimization of the detection engine alongside domestic CPUs, operating systems, and underlying

Table 10. Compliance index across different endpoints

Windows10-1	Windows 10	Windows 11
72.4	76	87.2

Source(s): Authors' own work

firmware. This will ultimately lead to the development of a baseline verification mechanism tailored to the needs of the Xin-Chuang ecosystem.

At the same time, research will explore the integration of baseline assessment tools with domestic security solutions such as Endpoint Detection and Response (EDR) and antivirus systems. The goal is to enable policy coordination and data interoperability, thereby forming a cohesive closed-loop management capability (Axxiome, 2021). With the increasing adoption of technologies such as quantum computing and large AI models (Gartner, 2023), security baseline management is set to evolve further toward predictive defense and adaptive strategies, solidifying its role as an indispensable "security foundation" in the enterprise's digital strategy.

References

Axxiome (2021). Advanced threats: Fileless attacks, Available from: https://www.sohu.com/a/489640391_120205504 (accessed 5 May 2024).

Bai, J., Tian, K., & Li, B. (2024). *Digital system security hardening technologies*. Beijing: Publishing House of Electronics Industry.

Gan, B., Sun, Y. X., Ling, X., & Li, X. X. (2025). Network security baseline configuration for terminals based on domestic operating systems. *Cybersecurity and Informatization*, *4*, 163–166.

Gartner. (2023). *Innovation insight for unified endpoint security.* Stamford, CT: Gartner Research.

GitHub Community (2022). PowerShell baseline compliance scanner, Available from: https://github.com/baseline-scanner (accessed 1 October 2023).

IBM Security. (2023). *Cost of a data breach report 2023*. Armonk, NY: IBM Corporation.

Joint Task Force. (2020). *NIST SP 800-53 Rev.5: Security and privacy controls for information systems and organizations*. Gaithersburg, MD: National Institute of Standards and Technology.

Li, M., & Wang, Q. (2022). Research on Windows terminal compliance hardening for classified protection 2.0 based on GPO. *Computer Engineering*, *48*(6), 120–128.

Li, C., & Xia, W. (2018). Research on endpoint security based on Windows log analysis. *Cyberspace Security, 9*(3), 70–77.

Microsoft (2023). Modifying audit policy via registry and group policy operations with secedit command-line, Available from: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/secedit (accessed 22 August 2023).

Microsoft (2025). *Get-service. PowerShell documentation*, Available from: https://docs.microsoft.com/zh-cn/powershell/module/microsoft.powershell.management/get-service (accessed 15 January 2025).

National Information Security Standardization Technical Committee. (2017). *GB/T 35283-2017 Information security technology—core configuration baseline structure specifications for computer terminals*. Beijing: China Standards Press.

National Information Security Standardization Technical Committee. (2019). *GB/T 22239-2019 Information security technology—baseline for classified protection of cybersecurity.* Beijing: China Standards Press.

National Information Security Standardization Technical Committee (2024a). *GB/T 29240-2024 Cybersecurity technology—General security technical specifications for terminal computers*. Beijing: China Standards Press.

National Information Security Standardization Technical Committee (2024b). *GB/T 30278-2024 Cybersecurity technology—core configuration specifications for government computer terminals*. Beijing: China Standards Press.

NSFOCUS (2023). Implementation of security assessment solutions for financial XC systems, Available from: https://www.xckfsq.com/news/show.html?id=21551 (accessed 10 December 2023).

Zhang, W., & Chen, L. (2021). Design of Windows security baseline detection tool under classified protection 2.0. *Journal of Cyber Security, 16*(3), 45–53.

Further reading

Microsoft Corporation (2021). Microsoft security compliance toolkit, Available from: https://learn.microsoft.com/en-us/windows/security/operating-system-security/device-management/windows-security-configuration-framework/security-compliance-toolkit-10 (accessed 1 October 2023).

Corresponding author

YaQiong Xu can be contacted at: mocco87@163.com

Fan Miao is a senior engineer at China Railway Academy of Sciences Corporation Limited, specializing in research on data security and privacy protection. His innovative achievements have won him 3 provincial and ministerial-level scientific awards, 9 invention patents, and 3 technical honors from China Academy of Railway Sciences Corporation Limited.