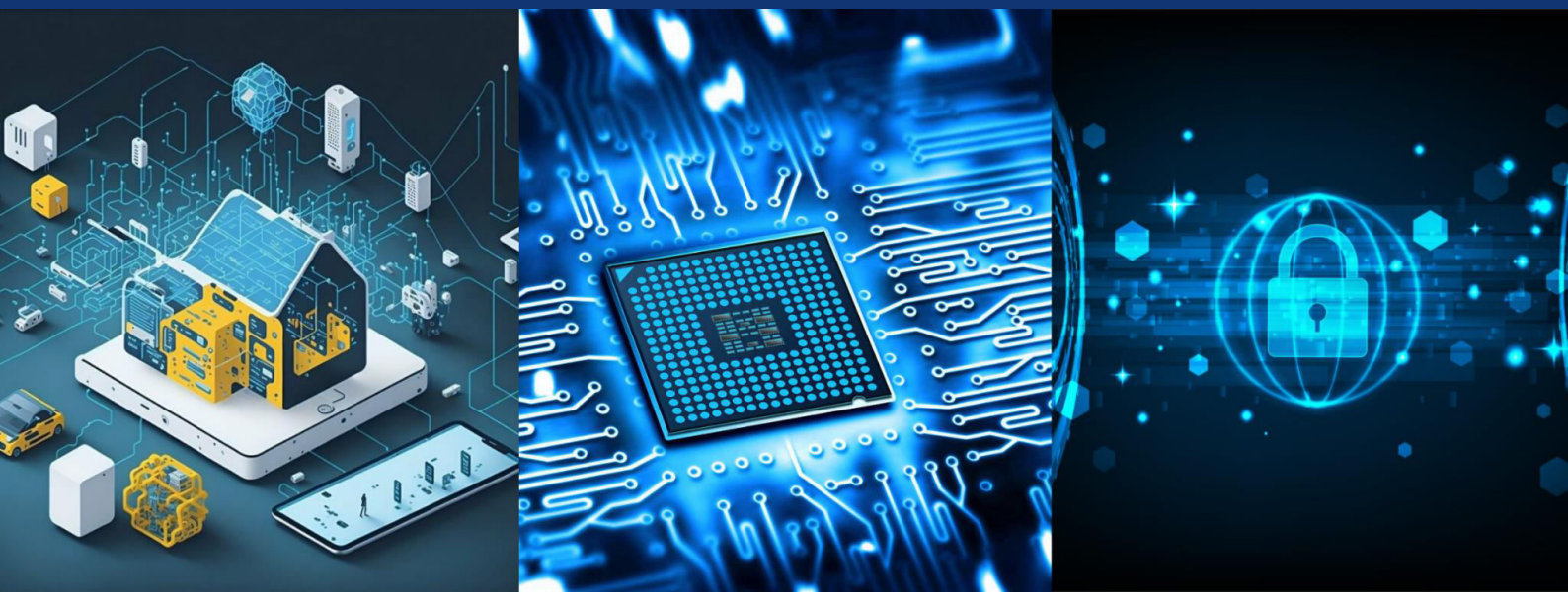
IJIRCCE

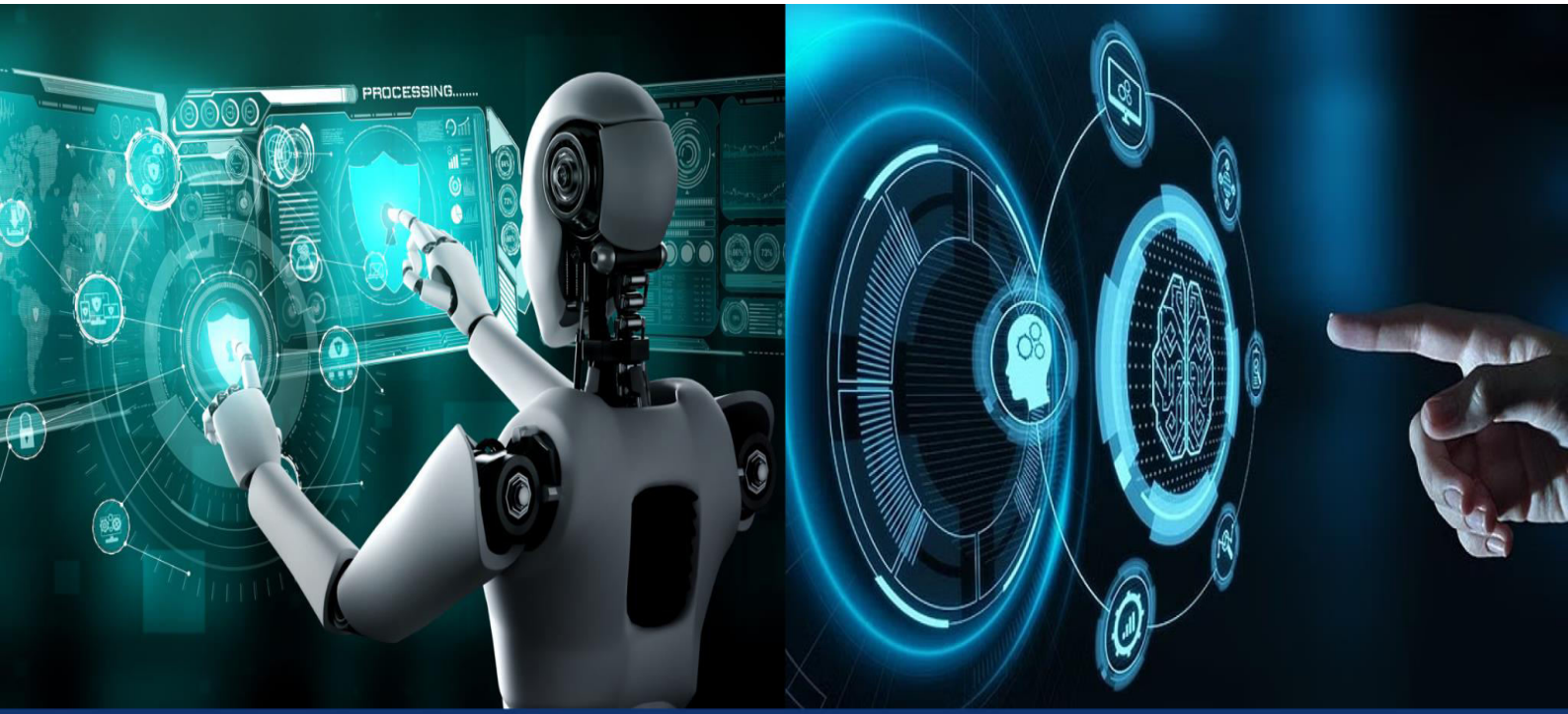
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 14, Issue 4, April 2026

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404032

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

FalconX: An Enterprise Security Audit System

Dharmabhaskar Panchgalle[1], Jay Navale[1], Piyush Latne[1], Rohit Chavhan[1], Prof. D. G. Jadhav[2]

Department of Information Technology, Sinhgad College of Engineering, Maharashtra, India[1]

Guided, Department of Information Technology, Sinhgad College of Engineering, Maharashtra, India[2]

ABSTRACT: In today's rapidly evolving digital landscape, enterprises operate in complex hybrid environments that significantly increase their exposure to cybersecurity threats such as misconfigurations, unauthorized access, and evolving vulnerabilities [2][5][11]. This paper presents FalconX, an automated Enterprise Security Audit System (ESAS) designed to streamline vulnerability assessment, compliance validation, and continuous monitoring within a unified platform. Traditional security auditing approaches are largely manual, time-consuming, and lack real-time visibility, making them inefficient for modern infrastructures [3]. This paper presents FalconX, an automated Enterprise Security Audit System (ESAS) designed to streamline vulnerability assessment, compliance validation, and continuous monitoring within a unified platform. The system integrates industry-standard tools such as Nmap and OpenVAS with a Flask-based architecture to perform audits and generate centralized reports. FalconX introduces a hybrid approach combining automated vulnerability scanning, rule-based compliance verification aligned with standards such as ISO 27001 and NIST [4][12], and Role-Based Access Control (RBAC) to ensure secure and controlled audit execution. Experimental evaluation demonstrates a reduction in audit time and improved operational efficiency compared to traditional methods [6]. The system provides scalable and customizable security auditing capabilities, making it suitable for modern enterprise environments.

KEYWORDS: Enterprise Security, Audit, Compliance, Vulnerability Scanning, Risk Management.

I. INTRODUCTION

The rapid adoption of advanced digital infrastructures has significantly increased cybersecurity risks, operational dependencies, and exposure to regulatory challenges in modern enterprises [2][5][11]. Organizations today operate in complex hybrid environments that integrate on-premises systems, cloud platforms, and distributed applications. While this enhances scalability and efficiency, it also introduces critical security challenges such as misconfigurations, unauthorized access, privilege misuse, and network-based attacks [13]. In such environments, security auditing is essential to maintain system integrity and ensure compliance with established standards [3][4].

Traditional auditing techniques rely on periodic manual assessments conducted by security professionals, involving system inspections, compliance verification, vulnerability analysis, and report generation. However, these approaches are time-consuming, labor-intensive, and provide only a static view of the system's security posture [3]. As enterprise environments continuously evolve, vulnerabilities may remain undetected between audit cycles, increasing the risk of potential security breaches [5].

To address these limitations, this paper presents FalconX, an implemented Enterprise Security Audit System (ESAS) that automates vulnerability assessment, compliance validation, and audit reporting within a unified framework. FalconX integrates tools such as Nmap and OpenVAS with rule-based verification aligned to standards like ISO 27001 and NIST [4][12]. The system incorporates Role-Based Access Control (RBAC) to ensure secure audit execution, restricting unauthorized actions and allowing controlled access to audit functionalities.

By enabling centralized reporting through an interactive dashboard, FalconX enhances visibility into enterprise security posture while significantly reducing manual effort. This paper presents the design, implementation, and evaluation of FalconX, demonstrating its effectiveness as a scalable and efficient solution for modern enterprise security auditing [1].

II. RELATED WORK

Security auditing is essential for ensuring that enterprise systems comply with regulatory and organizational security requirements [3][4]. It involves evaluating system configurations, policies, and practices to identify vulnerabilities and

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404032

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

ensure adherence to standards such as ISO 27001, NIST, and PCI DSS [4][12]. Effective auditing helps organizations reduce risk, maintain compliance, and build trust with stakeholders, especially in critical sectors like banking, healthcare, and e-commerce [5].

Several tools support different aspects of security auditing. Network scanners such as Nmap are used for identifying open ports and services, while vulnerability assessment tools like OpenVAS and Nessus detect known security weaknesses [6][15][16]. However, these tools operate independently and lack integration with compliance validation, centralized reporting, and controlled access mechanisms.

Existing approaches also provide limited support for unified auditing and secure access control [7][8]. To overcome these limitations, FalconX integrates vulnerability scanning, compliance validation, and Role-Based Access Control (RBAC) into a single platform, enabling efficient, centralized, and secure enterprise security auditing.

III. METHODOLOGY

Design Considerations:

- Sources of data collection include APIs, vulnerability scanning tools, log analysis modules and manual inputs [6].
- Compliance rules are based on established frameworks such as ISO 27001 and NIST [4][12].
- The audit interface provides a dynamic dashboard for visualization of audit results.
- Data validation includes both system configurations and policy-based rules [9].
- The system is designed to minimize manual intervention while maintaining simplicity and efficiency.
- Role-Based Access Control (RBAC) is implemented to ensure controlled access to audit functionalities.

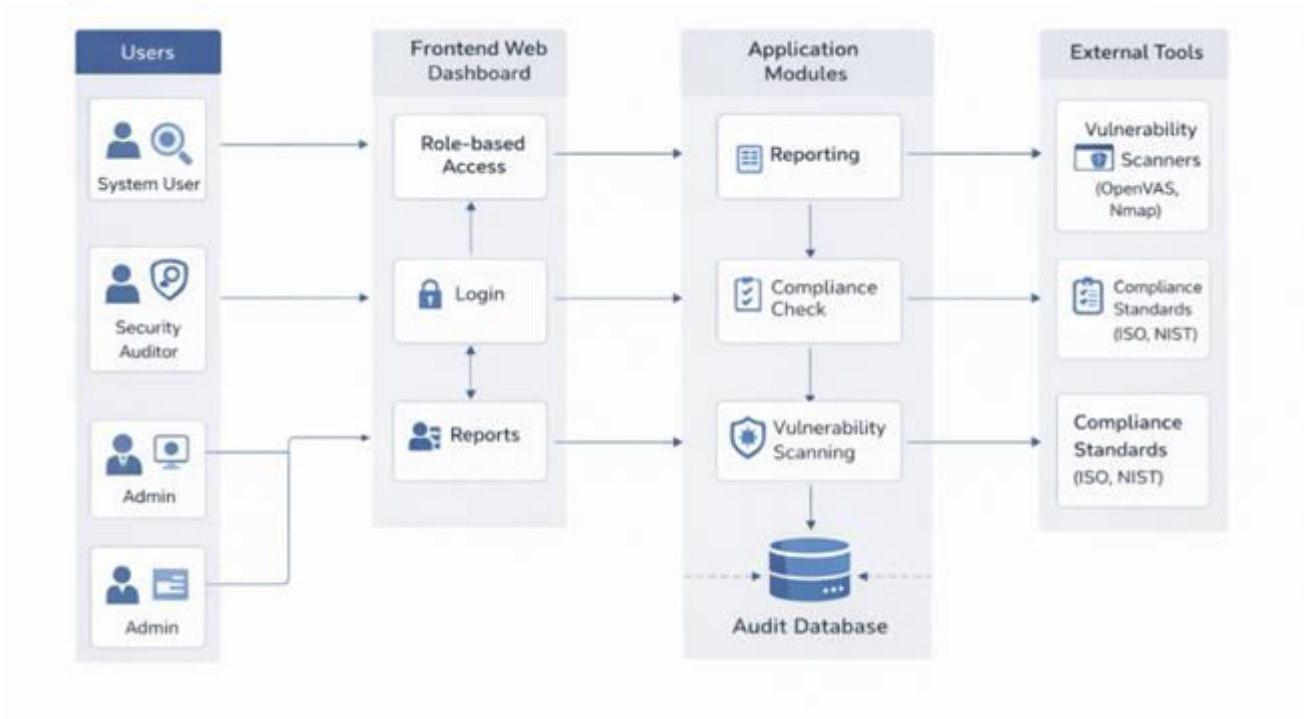


Fig. 1. System Architecture of FalconX Enterprise Security Audit System

Explanation of the Proposed Algorithm:
The objective of the methodology is to automate enterprise security auditing by collecting relevant system data, validating it against predefined compliance rules, and presenting the results through a centralized interface. FalconX follows a structured three-step approach to ensure efficient and reliable auditing.

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404032

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Step 1: Data Collection
• Collect data from enterprise systems using multiple methods:
1. APIs: Retrieve system configuration and related information where applicable.
2. Vulnerability Scanning: Use tools such as Nmap to identify open ports, services, and potential security exposure [6].
3. Log Analysis: Perform basic analysis of system or application logs to identify possible misconfigurations or anomalies [12].
4. Manual Inputs: Allow users to provide configurations or policies that cannot be automatically retrieved.
• Combine all collected data into a unified dataset for further evaluation.

Step 2: Compliance Validation Against Rules
• Compare collected data with predefined compliance rules based on selected security standards [4].
• For each system configuration or log entry:
• If the data satisfies the rule → label as COMPLIANT
• If the data does not satisfy the rule → label as NON-COMPLIANT
• Calculate compliance score for each system/module:
• Compliance Score (%) = (Compliant Checks ÷ Total Checks) × 100
• Identify and list non-compliant configurations along with their associated risk levels.

Step 3: Interface Update and Audit Reporting
• Display audit results through a dashboard interface.
• Present the following details:
• Compliance scores and basic risk levels
• List of non-compliant configurations
• Summary of identified vulnerabilities
• Allow authorized users to:
• Trigger audits
• View audit results
• Download audit reports
• Restrict viewer role to only access audit results without permission to perform audits (RBAC enforcement).

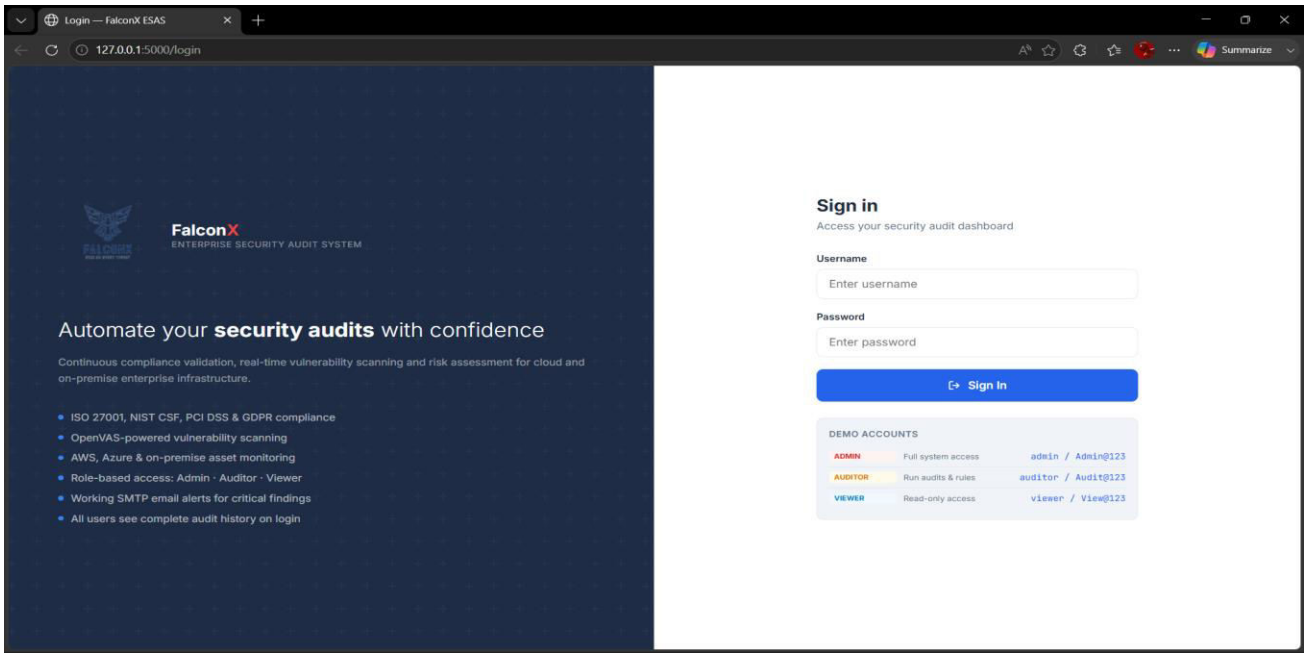


Fig. 2. FalconX Login Interface with Role-Based Access Control

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404032

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. PSEUDO CODE

Step 1: Data Collection

• Collect data from multiple sources including APIs, vulnerability scanning tools, system logs, and manual inputs [6].
• Combine all collected data into a unified dataset for further processing.

Step 2: Verification Against Compliance Rules
•For every compliance rule [4]:
• Evaluate relevant data from the dataset.
• If the data satisfies the rule → mark as COMPLIANT.
• If the data does not satisfy the rule → mark as NON-COMPLIANT

Step 3: Calculate Compliance Score

• Compliance Score (%) = (Number of Compliant Checks ÷ Total Checks) × 100

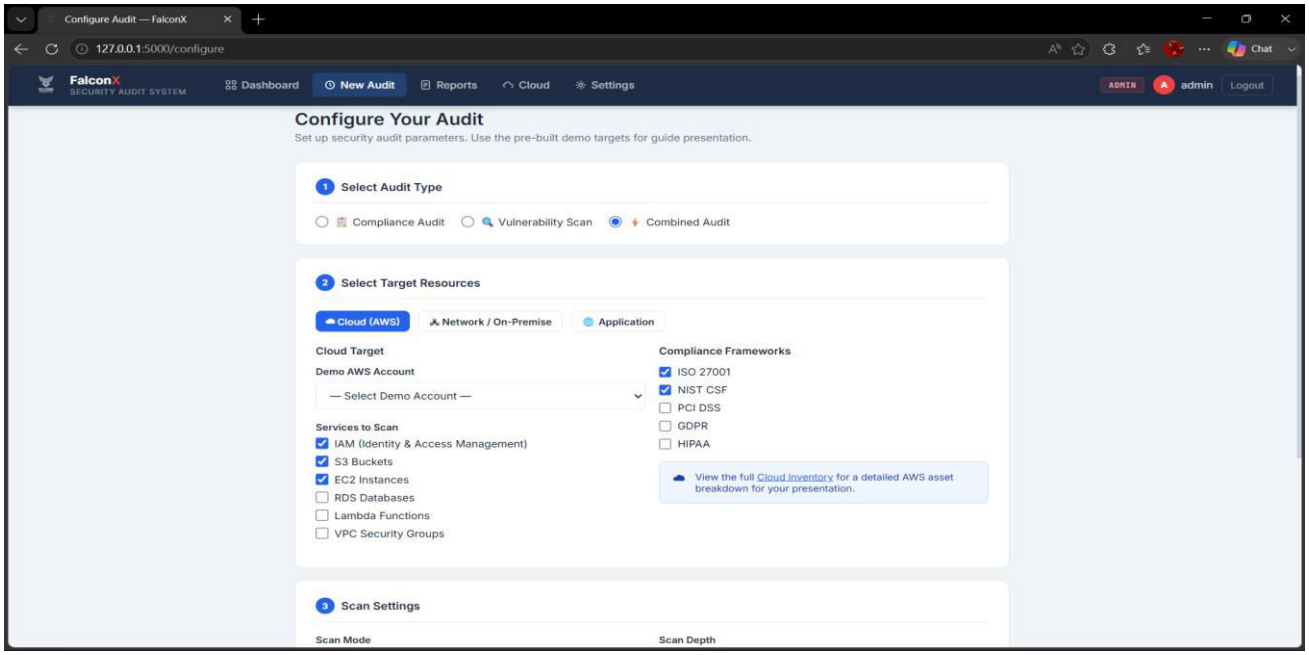


Fig. 3. Audit Configuration Interface in FalconX

Step 4: Audit Reporting and Dashboard Update
• Generate audit results based on compliance evaluation.
• Display results on the dashboard including compliance status and basic risk levels.
• Generate and store detailed audit reports for authorized users.

Step 5: Continuous Monitoring
• Allow users to trigger audits periodically or as required.
• Update dashboard and reports based on latest audit results.
• Highlight critical non-compliant configurations for further action.

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404032

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

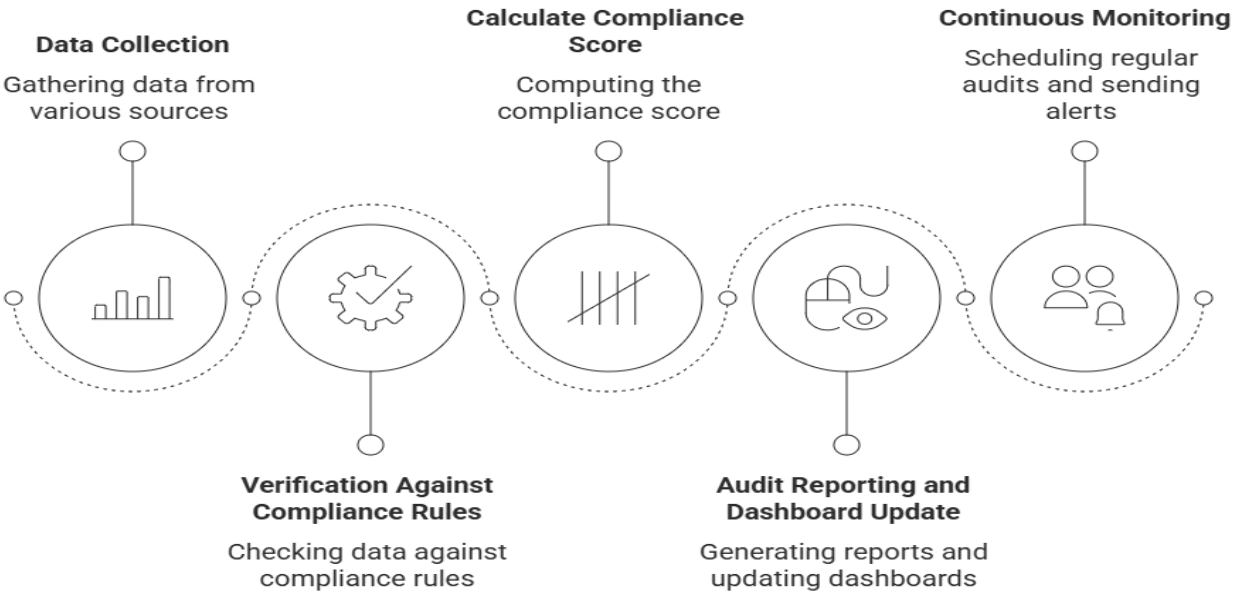


Fig. 4. Enterprise Security Audit Process

V. EXPERIMENTAL RESULTS AND ANALYSIS

The FalconX system was implemented and tested on sample enterprise scenarios to evaluate its effectiveness in automating security auditing. The system utilizes vulnerability scanning (using Nmap), rule-based compliance validation aligned with ISO 27001 and NIST guidelines [4][12], and a dashboard interface for reporting audit results.

In the first test scenario, FalconX evaluated password policy compliance by analyzing system configurations against predefined rules such as minimum password length and basic security requirements [4]. The system automatically verified these configurations and displayed the compliance status on the dashboard, enabling quick identification of non-compliant settings.

In the second scenario, network auditing was performed using Nmap to identify open ports and active services on target systems [6]. The scan results were compared against predefined compliance rules to detect unnecessary exposed services. The system successfully identified non-compliant ports and reflected the results in the audit dashboard, assisting in enforcing secure configuration practices [9].

Metric	Manual / Traditional Auditing	FalconX System	Significance
Audit Execution Time	High (Takes hours to days due to manual processes)	Low (Completed in minutes using automation)	Faster audit cycles and improved efficiency
Accuracy & Consistency	Moderate (Depends on auditor expertise)	High (Rule-based validation ensures uniform results)	Minimizes human error and improves reliability
Vulnerability Detection	Limited (Manual identification or separate tools)	Integrated (Uses Nmap for detecting open ports/services)	Centralized and efficient detection process
Compliance Verification	Manual checklist-based validation	Automated rule-based validation (ISO/NIST aligned)	Ensures systematic and repeatable compliance checks

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404032

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Reporting Mechanism	Manual report preparation	Dashboard-based automated reporting	Improves visibility and ease of analysis
Access Control	Limited or not enforced strictly	Role-Based Access Control (Admin, Auditor, Viewer)	Ensures secure and controlled system usage

Compared to traditional manual auditing methods, FalconX significantly reduces the time required for performing security audits. Tasks that typically require extensive manual effort can be completed within a shorter duration using automated processes. Additionally, rule-based validation ensures consistency in results and reduces dependency on individual interpretation.

However, certain limitations exist. Some aspects of security auditing, such as physical security controls and user behavior analysis, cannot be fully automated and require manual verification. Furthermore, the system is currently tested on a limited scale, and performance in large-scale enterprise environments may require further optimization.

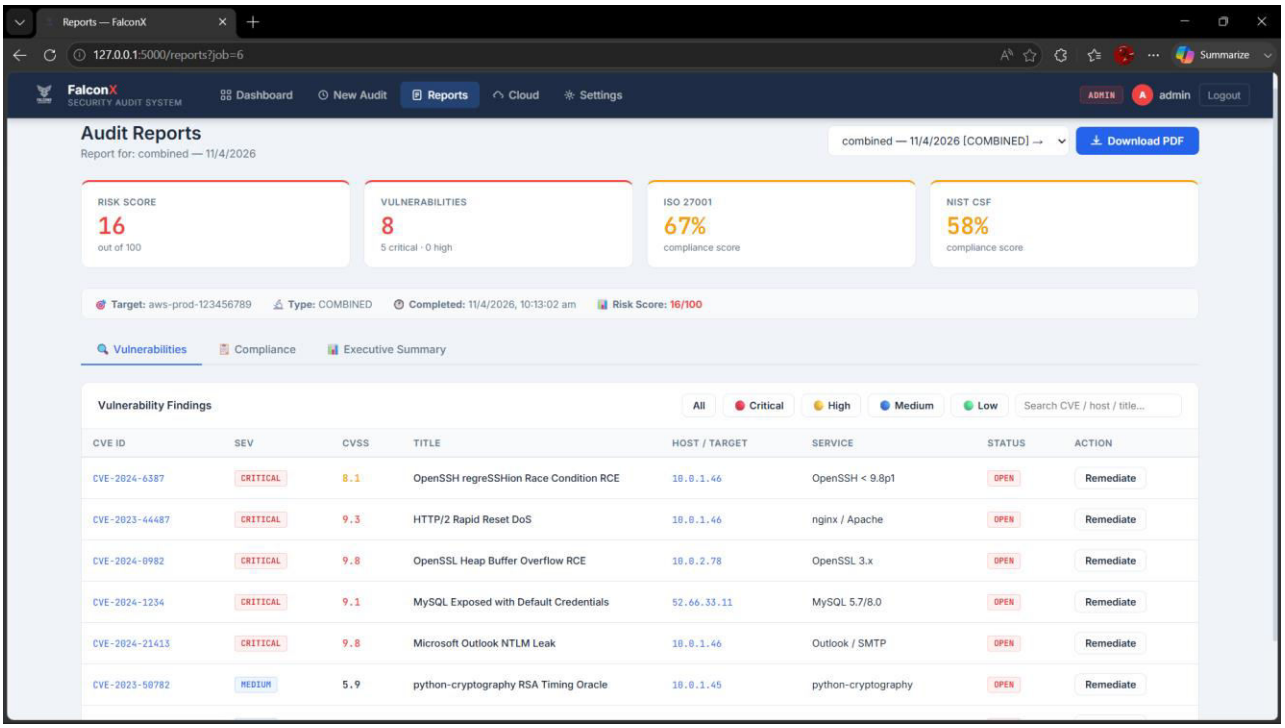


Fig. 5. FalconX Audit Report Showing Detected Vulnerabilities and Risk Scores

VI. CONCLUSION AND FUTURE WORK

This paper presented the design and implementation of FalconX, an automated Enterprise Security Audit System developed to enhance the efficiency and reliability of security auditing in enterprise environments. By integrating vulnerability scanning, rule-based compliance validation, and centralized reporting, FalconX addresses the limitations of traditional manual auditing approaches. The system was able to evaluate basic security controls such as password policies and network configurations, providing improved visibility and reducing the need for manual intervention.

The implementation demonstrates that FalconX can significantly streamline the auditing process by reducing audit time and ensuring consistent evaluation through rule-based mechanisms. Additionally, the incorporation of Role-Based Access Control (RBAC) enhances system security by restricting audit operations to authorized users, making the system suitable for controlled enterprise environments.

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404032

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

However, certain limitations remain. Some aspects of security auditing, such as physical security controls and user behavior, cannot be fully automated and require manual validation. Furthermore, the current implementation has been tested on a limited scale, and performance in large-scale enterprise environments requires further evaluation.

Future work will focus on extending support for additional compliance frameworks, improving scalability, and enhancing system capabilities through better data analysis techniques [7][8]. These improvements can further strengthen FalconX as a practical and scalable solution for enterprise security auditing.

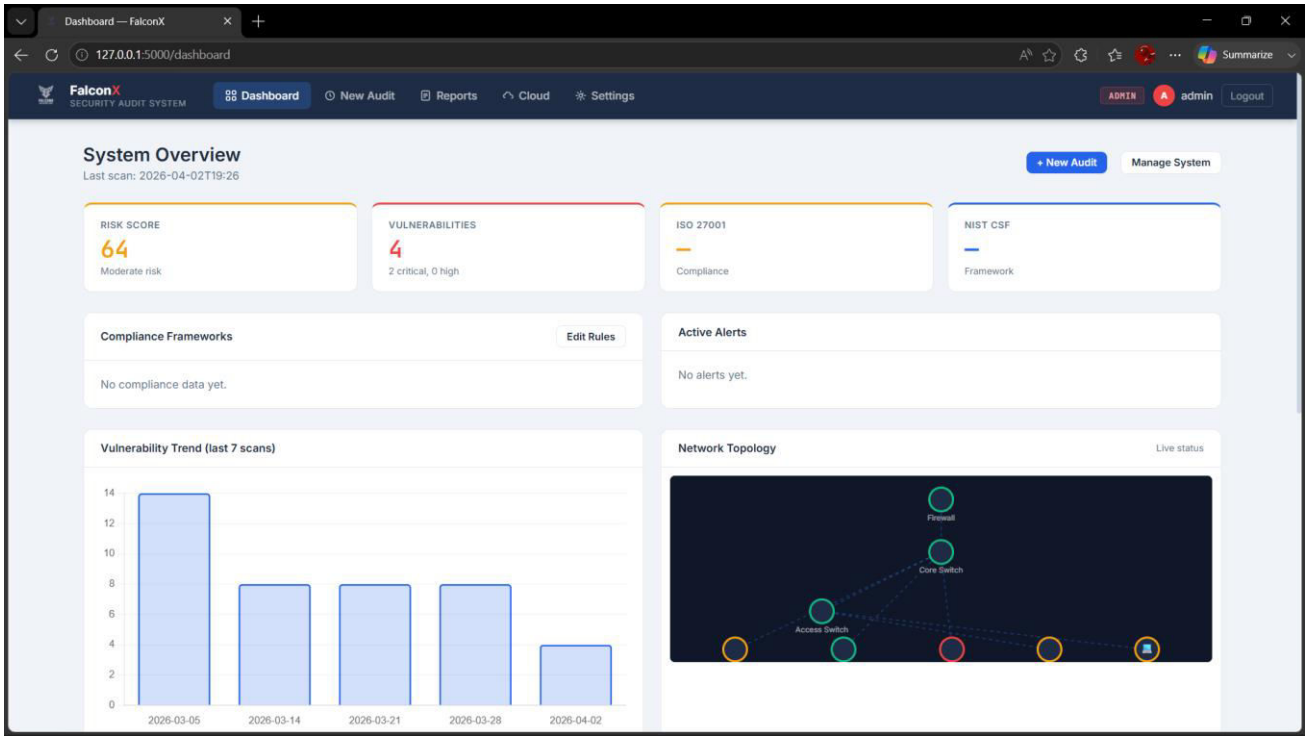


Fig. 6. FalconX Dashboard Displaying System Overview and Risk Metrics

REFERENCES

[1] D. Panchgalle, J. Navale, P. Latne, and R. Chavhan, "Enterprise security audit system," International Journal of Innovative Research in Computer and Communication Engineering, vol. 13, Issue 11, November 2025.
[2] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*. Sebastopol, CA, USA: O'Reilly Media, 2009.
[3] K. Julisch, "Security compliance: The next frontier in security research," in *Proc. New Security Paradigms Workshop (NSPW)*, 2009, pp. 71–74.
[4] International Organization for Standardization, *ISO/IEC 27001:2013 – Information Security Management Systems Requirements*, ISO, 2013.
[5] Symantec Corporation, *State of Enterprise Security Report 2010*, Symantec, 2010.
[6] E. A. Altulaihan, A. Alismail, and M. Frikha, "A survey on web application penetration testing," *Electronics*, vol. 12, no. 1229, pp. 1–25, 2023.
[7] Z. Hu, R. Beuran, and Y. Tan, "Automated penetration testing using deep reinforcement learning," in *Proc. IEEE EuroS&P Workshops*, 2020, pp. 1–10.
[8] W. B. Shahid *et al.*, "Deep learning-based framework for web attacks detection, mitigation, and attacker profiling," *Journal of Network and Computer Applications*, vol. 198, p. 103270, 2022.
[9] S. K. Lala, A. Kumar, and T. Subbulakshmi, "Secure web development using OWASP guidelines," in *Proc. ICICCS*, 2021, pp. 1–6.
[10] A. Shostack, *Threat Modeling: Designing for Security*. Indianapolis, IN, USA: Wiley, 2014.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[11] J. Vehent, *Securing DevOps: Security in the Cloud*. Shelter Island, NY, USA: Manning Publications, 2018.
[12] R. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (IDPS)," NIST Special Publication 800-94, 2007.
[13] OWASP Foundation, "OWASP Top 10: The Ten Most Critical Web Application Security Risks," 2021.
[14] G. McGraw, "Software security," IEEE Security & Privacy, vol. 2, no. 2, pp. 80–83, 2004.
[15] Tenable, "Nessus Vulnerability Scanner," [Online]. Available: https://www.tenable.com
[16] Qualys Inc., "Qualys Cloud Platform," [Online]. Available: https://www.qualys.com

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details