Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

AI Techniques for Operating System Security Compliance Automation: A Survey

Aniket Kardile
Student, Dept. of Computer Engineering
Pune Institute of Computer Technology
Pune, India

Tanay Duddalwar
Student, Dept. of Computer Engineering
Pune Institute of Computer Technology
Pune, India

Vedant Ghumare
Student, Dept. of Computer Engineering
Pune Institute of Computer Technology
Pune, India

Siddhesh Sangale
Student, Dept. of Computer Engineering
Pune Institute of Computer Technology
Pune, India

Mrs. Rajani Jadhav
Assistant Professor, Dept. of Computer Engineering
Pune Institute of Computer Technology
Pune, India

***Abstract* – Protecting an organization's IT infrastructure requires operating systems to comply with well-defined security standards and best practices. To achieve this, many organizations rely on security frameworks published by authorities such as the Center for Internet Security (CIS) and the Defense Information Systems Agency (DISA). These frameworks provide detailed benchmarks and guidelines that help organizations configure and maintain their systems securely. In practice, most of these security recommendations are written as textual rules in natural language. Because the guidelines are often unstructured, the process of auditing systems against them becomes complex and time-consuming. As a result, organizations frequently use these guidelines to design auditing procedures as well as remediation mechanisms that help correct security misconfigurations. This paper presents a survey of different techniques proposed for ensuring operating system security compliance, with particular focus on approaches that incorporate artificial intelligence. It reviews traditional compliance auditing practices, rule-based remediation mechanisms, and more recent methods that utilize machine learning. Special attention is given to techniques based on natural language processing and large language models that assist in automatically generating audit or remediation scripts. The survey also examines current research from the perspective of automation capabilities, advantages, and limitations. Several practical challenges are discussed, including concerns related to reliability, scalability, and potential security risks associated with AI-generated scripts. Finally, the paper highlights possible research directions that could contribute to improving automated compliance verification and remediation in operating systems.**

***Keywords* – Operating System Security Compliance, Security Auditing, Automated Remediation, Artificial Intelligence in Security, Large Language Models, Compliance Automation.**

I. INTRODUCTION

In a modern computing environment, operating systems are a fundamental part of the overall ecosystem, providing resource management for hardware devices as well as hosting enterprise-level applications. As a result, the operating system becomes a primary point of attack for cybercriminals. Many of the most common attacks, including privilege escalation, ransomware, and unauthorized access, are often a result of vulnerabilities or misconfigurations of the underlying operating system. As a result, operating system security is a critical aspect of any overall security posture, as a compromised operating environment will negate any security controls implemented at the application layer.

For improving system security and minimizing vulnerabilities, various authoritative bodies have developed standardized guidelines for system security. Some of the most widely used guidelines are provided by the Center for Internet Security (CIS) and the Defense Information Systems Agency (DISA), which provide comprehensive security benchmarks for various operating systems, including Windows Server, Linux, and Red Hat Enterprise Linux. These guidelines provide recommended configurations for user access control, file system permissions, network security, and various important system parameters.

Despite the availability of these standards, it is a tedious and labor-intensive process to implement and enforce operating system security policies. The CIS and DISA documents are generally hundreds of pages long and include detailed security rules written in a natural language. The security team has to interpret these guidelines and implement them as a series of executable scripts, for example, a bash shell script for Linux or a PowerShell script for Windows, to audit the system's configuration. In cases where a system is found to be non-compliant, additional remediation scripts have to be developed to fix the non-compliant issues. This is a tedious and labor-intensive process and is also extremely prone to human errors.

Recent developments in the field of artificial intelligence, especially in natural language processing (NLP) and large language models (LLMs), have created new opportunities to automate security compliance activities. It is evident that LLMs possess a high level of proficiency in processing natural

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

language input and generating system-level commands or scripts. Furthermore, a new concept, named "Retrieval Augmented Generation" (RAG), has come into existence, which is highly beneficial in enhancing the reliability of an AI system. It is possible to retrieve relevant information from external sources before generating a response, which can be used to interpret natural language-based security guidelines.

Considering these developments, a number of recent research studies have started exploring the possibility of using AI-based techniques in the automation of the security auditing and compliance management processes in operating systems. However, the overall landscape of these techniques is still fragmented, and a detailed analysis of the existing methodologies, tools, and research trends is required in this area.

Hence, the current paper aims at providing a thorough survey of the existing techniques in the field of automating operating system security compliance, especially focusing on the use of AI-based techniques in the field of audit and remediation automation. This survey also aims at comparing the existing techniques, highlighting the current challenges, and suggesting potential research directions in the field of more reliable and automated compliance systems.

II. BACKGROUND

1. Operating System Security Compliance

The operating systems provide the foundation for modern computing systems, managing hardware resources, as well as supporting enterprise-level applications and services. The operating systems, being a fundamental part of modern computing systems, are often a target for cyber-attacks, which take advantage of vulnerabilities, configuration issues, or insufficient access controls. To address the identified cyber threats, organizations implement various security compliance standards that provide guidelines for recommended operating system configurations and operational guidelines. The security compliance standards, developed by organizations such as the Center for Internet Security (CIS) and the Defense Information Systems Agency (DISA), provide detailed guidelines in the form of operating system benchmarks, as well as security technical implementation guides (STIGs) for various operating systems, including Windows Server, Linux, and Red Hat Enterprise Linux. The guidelines provide a set of rules for securing various operating systems, including user accounts, password policies, file systems, as well as network configurations. The guidelines help organizations improve the overall security posture of operating systems, thus becoming compliant with industry best practices.

2. Security Auditing and Remediation

The process of evaluating a system's configuration and state of operation against a set of security standards is called security auditing. Auditing of a system usually involves a series of checks on the system's parameters, configuration files, and access control mechanisms to evaluate whether the system is compliant with a set of security standards and guidelines. When a system is found to be non-compliant, remediation activities need to be performed to bring the system back to a compliant state. Traditionally, this is achieved through custom scripts, which can be Bash scripts for Linux systems and PowerShell scripts for Windows systems. However, this process is often challenging and cumbersome, especially when dealing with large-scale infrastructures and a wide range of compliance standards.

3. Security Auditing and Remediation

To simplify the process of managing systems, organizations are increasingly turning to configuration management tools. Tools such as Ansible, Puppet, Chef, and Chef InSpec offer various mechanisms to define configuration policies for systems, which can be easily implemented across multiple systems. These tools allow administrators to define policies, implement them, and monitor the systems to check for compliance. The Infrastructure as Code (IaC) method also offers a way to automate systems, which improves the process of configuration management. Despite the availability of tools to simplify the process of configuration management, administrators still need to interpret the security guidelines, which is a difficult process.

4. Artificial Intelligence in Security Automation

Recent developments in the field of artificial intelligence have created new opportunities to automate complex security management activities. Natural language processing, a subfield of artificial intelligence, can now interpret unstructured textual data, such as security guidelines or documents. Large Language Models (LLMs) have shown promising results in understanding natural language-based instructions, such as generating executable code, including system administration commands or scripts. Furthermore, the incorporation of a subfield of natural language processing, named "Retrieval Augmented Generation" (RAG), can significantly improve the reliability of LLM-based systems, which can retrieve relevant information from external sources before generating outputs. Using a combination of semantic search and generation, an AI-based system can interpret security compliance guidelines, such as generating audit or remediation procedures, which is being explored to enhance the efficiency, accuracy, and scalability of operating system security compliance automation.

III. LITERATURE SURVEY

This section discusses the existing research related to security compliance automation, AI-based script generation, and retrieval-augmented knowledge systems. The existing research is divided into various categories based on the technology used to automate the security auditing and remediation processes.

1. Operating System Security Auditing and Compliance Tools

There are several studies focusing on the automated auditing of the security of operating systems using pre-defined security standards. Sedano and Salman have proposed a framework for auditing the security of Linux systems using the CIS security benchmark standard. This study has shown the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

potential of automated auditing tools in simplifying the process of verification and compliance using the compliance rules defined in the CIS benchmark standard. This study has shown the potential of automated auditing tools in simplifying the process of verification and compliance using the compliance rules defined in the CIS benchmark standard [2].

Karthiban et al. have proposed an automated CIS benchmark auditing and remediation tool, focusing on the Windows operating system. This tool has the potential to automatically scan the configuration of the operating systems and identify the potential security threats arising due to the violation of the security benchmark standard. This tool has the potential to automatically scan the configuration of the operating systems and identify the potential security threats arising due to the violation of the security benchmark standard [3].

This is a traditional approach to automating the process of compliance, where the rules and scripts have been manually implemented.

2. AI-Driven Compliance and Security Policy Automation

Recent research has also focused on the application of Large Language Models (LLMs) in automating security compliance activities. In this regard, Ahmed et al. proposed an approach whereby LLMs can be leveraged to automatically interpret CIS Critical Security Controls, generating validation logic to verify compliance. According to this research, LLMs can automatically translate security documents, which are often represented in natural language, into executable validation logic, thus enhancing the automation of security compliance activities [1].

In a similar trend, Fernández Saura et al. proposed an approach to automating security policy enforcement using LLMs. According to this research, LLMs can automatically understand complex security policy descriptions, thus generating corresponding system commands to execute the policies. This demonstrates the ability of LLMs to address the existing gap between security policy descriptions and their enforcement in the system [4].

In a recent study, Agarwal et al. proposed a novel approach to automating regulatory compliance, termed RAGulating Compliance, which combines the capabilities of Retrieval-Augmented Generation models and knowledge graphs. According to this research, this approach enables the accurate interpretation of compliance requirements, thus minimizing the chances of error in automated policy enforcement [5].

3. Semantic Retrieval and Embedding Techniques

The concept of semantic embedding plays a vital role in modern information retrieval systems used in AI-based automation tools. Reimers and Gurevych have introduced a method of "Sentence-BERT," which is an efficient implementation of the BERT algorithm used to create sentence embeddings for semantic similarity tasks. The algorithm helps to calculate the semantic similarity between sentences, making it an important part of document retrieval systems.

Karpukhin et al. have introduced an efficient information retrieval tool called "Dense Passage Retrieval," which helps to retrieve relevant documents from large-scale text corpora. The method of information retrieval is based on neural embeddings, which is an efficient method of information retrieval. The method is an improvement over previous tools, as it calculates the semantic similarity between query sentences and documents using dense vectors. The method is an important part of modern knowledge retrieval tools.

4. Retrieval-Augmented Generation for Knowledge-Based Systems

Retrieval-Augmented Generation (RAG) has been identified as one of the most important approaches in the integration of knowledge retrieval with other AI models. The RAG architecture was proposed by Lewis et al. The architecture integrates dense retrieval models with sequence-to-sequence generative models. The language model, therefore, has the capability to retrieve relevant information from other sources of knowledge before responding to a question. The accuracy of the language model in responding to questions has thus been enhanced through the incorporation of this architecture [6].

Shuster et al. have used the concept of retrieval-augmented models in the reduction of hallucinations in large language models. The work has shown that the RAG system has the capability to generate reliable results, thus ensuring that there is no incorrect information being provided to the users. The importance of this aspect of the RAG system cannot be emphasized enough, considering the importance of accuracy in the cybersecurity domain, where incorrect output might lead to incorrect configuration of systems [9].

IV. RESEARCH GAP

However, despite the improvements made in security compliance automation, some drawbacks still exist in the conventional methods of security compliance and configuration management. The conventional security compliance tools and configuration management methodologies are mostly rule-based, which means that these tools and methodologies depend on the use of scripts for security audits and configuration management. The conventional methods of security compliance and configuration management face a major drawback in the translation of security guidelines, such as CIS and DISA, into scripts, as it is a time-consuming and complex process. In addition, the conventional methods of security compliance and configuration management are not able to interpret complex natural language, which is commonly used in security compliance guidelines.

Recent research has also attempted to address these problems using Artificial Intelligence and Large Language Models (LLMs) in script generation and security policy interpretation. However, these approaches using Artificial Intelligence have problems like reliability, hallucinations, and a lack of grounding in authoritative security documents. There is also a lack of research in the application of semantic retrieval-based methods, like Retrieval-Augmented Generation (RAG), in the field of compliance automation, which could help in more accurate script generation. Therefore, more research is needed in the development of Artificial Intelligence-based systems, which can use semantic retrieval

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

and knowledge grounding in the generation of scripts, in the automation of operating system security compliance.

V. PROPOSED METHODOLOGY

To address the challenges of traditional compliance analysis, this study introduces an intelligent compliance automation framework based on the Retrieval-Augmented Generation (RAG) approach. Security compliance standards like CIS Benchmarks consist of many configuration rules, audit guidelines, and remediation guidelines, usually found in lengthy unstructured documents. Understanding these documents and creating executable scripts through traditional methods takes a lot of time and requires specialized knowledge.

The proposed framework can automate this process by turning unstructured compliance documents into a structured knowledge repository and generating real-time audit and remediation scripts with a large language model. This framework uses techniques such as document preprocessing, semantic representation learning, and context-aware script generation.

The overall workflow includes two main phases: **Knowledge Base Construction** and **Real-Time Script Generation**. In the first phase, the compliance knowledge repository is organized and indexed. In the second phase, contextual information is gathered, and compliance scripts are generated based on the input from the query.

1. Phase 1: Knowledge Base Construction

The knowledge base construction phase is responsible for preparing the compliance documents for efficient semantic retrieval. This phase is carried out offline only once whenever new compliance standards or updated benchmark documents are introduced. The main aim of this phase is to convert the unstructured compliance documents into a structured knowledge representation form.

1.1 Document Ingestion and Content Extraction: The initial step involves incorporating compliance benchmark documents, which typically include security configuration guidelines, explanations, audit procedures, and details on how to address any issues found. These documents are handled to retrieve information and transform it into a format that machines can read easily. The information that has been extracted is regarded as the raw dataset used to develop compliance knowledge. Each rule description along with its associated information is treated as a knowledge unit, and it is kept for future semantic search purposes.

1.2 Semantic Segmentation of Compliance Rules: Because compliance documents are often long and include detailed technical information, the extracted data is broken down into smaller meaningful parts. This is done to make sure that the part of the compliance rule or configuration guideline being shown is clear and important. Breaking the compliance document into smaller parts improves the accuracy of the retrieval process because the system can match user queries with specific sections of the document rather than treating the entire document as one single unit. Each segment is a separate piece of knowledge that contributes to the development of the compliance knowledgebase.

1.3 Semantic Representation and Knowledge Indexing: Each of these divided text parts is shown as a semantic vector through an embedding model. These vectors represent the meaning of the compliance rules and help compare the meaning of the query and the document content. The created vectors are kept in a knowledge database that uses vectors, and it allows for searching based on how similar the vectors are. This enables the effective recognition of document parts that are semantically connected to a query through the use of vector database indexing. This vector-based knowledge repository serves as the retrieval part of the proposed RAG framework and is mainly used to supply factual information during the creation of scripts.

2. Phase 2: Real Time Script Generation

The second phase runs in real time and is triggered by the user's request to get information and scripts related to a specific compliance rule. This stage includes combining semantic retrieval with large language models to create organized and context-sensitive results.

2.1 User Query Representation: This process begins with a user submitting a query that relates to a compliance rule or a system configuration requirement. The query may be expressed in natural language or may reference a specific compliance standard. To enable matching based on meaning with the indexed knowledge base, the query is transformed into a vector form that exists in the same semantic space as the document fragments.

2.2 Context Retrieval via Similarity Search: The proposed framework uses semantic similarity searching instead of traditional keyword-based methods to retrieve relevant information regarding compliance. The system matches the user's query in vector form with the document embeddings that were created when the documents were indexed, to find relevant information from the documents. The similarity score is determined by comparing the query vector with the document vectors to assess how relevant the information in the documents is. The information from the relevant document is obtained by using the similarity score that is calculated to create the scripts.

2.3 Context-Augmented Prompt Engineering: The collected parts of the document are then merged with the initial question from the user to create a fresh prompt. This process is basically the enhancement phase of the RAG method. The inclusion of factual information derived directly from relevant compliance documents provides a solid foundation for the language model's prompt. This grounding helps make the outputs from the language model more accurate and also lowers the chance of incorrect outputs being generated.

2.4 Script Generation Using a Language Model: The context-enriched prompt is then transmitted to the Large Language Model through the API. Based on the retrieved information, the LLM performs the generation phase of the RAG process to create structured audit and remediation scripts. The large language model functions as a reasoning system because its output is based on retrieved factual information

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

rather than depending on built-in memory parameters. This ensures that the scripts created are not only in line with the requirements but also relevant and consistent with the intended standard.

2.5 Response Validation and Delivery: The output is checked to make sure it follows the correct structure and includes all the necessary parts required for automating compliance. After this stage, the scripts are returned to the user via the system's interface. By using semantic retrieval and reasoning based on language models, the proposed framework greatly reduces the amount of human effort needed to analyze compliance standards and create scripts for conducting audits. The process is completed within seconds**.**

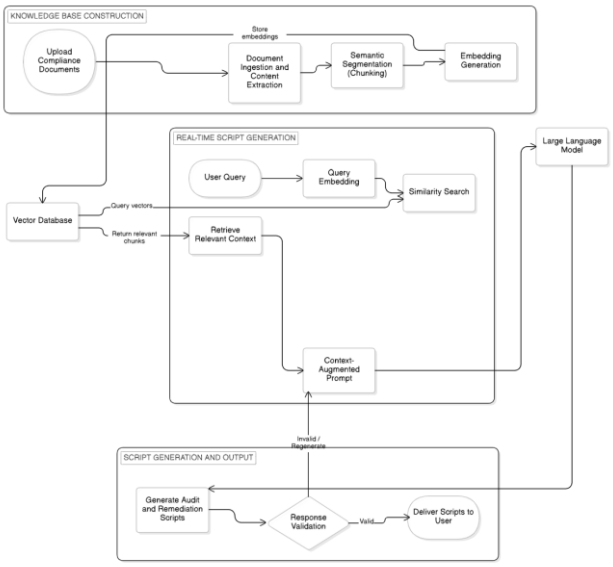

Fig. 1: End-to-End Flow of Proposed Methodology for Operating System Compliance Using AI-Techniques

VI. CONCLUSION

Operating system security compliance is a significant requirement for safeguarding enterprise infrastructure from potential vulnerabilities due to improper configurations and inadequate security policies. There are a number of security standards, such as CIS Benchmarks and DISA, which provide well-defined guidelines for securing operating systems. However, developing audit and remediation scripts from these voluminous documents is a tedious process. The survey was conducted by reviewing the research literature available on the automation of security compliance, which includes traditional rule-based audit tools, configuration management frameworks, and recent research using AI.

This analysis reveals that while conventional approaches are reliable for compliance verification, they demand manual effort in script development or maintenance. Recent advancements in artificial intelligence, specifically Large Language Models or retrieval-based approaches, are believed to hold great promise in automating the interpretation of security guidelines or script generation at the system level. As a result of the above analysis, the paper also discusses a possible methodology based on semantic retrieval or generative models, which could be beneficial in automating compliance enforcement, thereby reducing the effort required in managing the operating system security in the future.

REFERENCES

[1] M. Ahmed, J. Wei, and E. Al-Shaer, "Prompting LLM to enforce and validate CIS critical security controls," *Proc. ACM Symp. on Access Control Models and Technologies (SACMAT)*, 2024, doi:10.1145/3649158.3657036.

[2] W. K. Sedano and M. Salman, "Auditing Linux operating system with Center for Internet Security (CIS) standard," *Proc. IEEE Int. Conf. on Information Technology (ICIT)*, 2021, doi:10.1109/ICIT52682.2021.9491663.

[3] R. Karthiban, S. Archana, N. Harish Kumar, M. K. Kavin Nandha, R. Keren, and K. Keerthi Raghavan, "Automated CIS benchmark auditing and remediation tool: A Windows system security assessment solution," *International Journal of Innovative Research in Technology (IJIRT)*, 2024.

[4] P. Fernández Saura, K. R. Jayaram, V. Isahagian, J. B. Bernabé, and A. Skarmeta, "On automating security policies with contemporary LLMs," *arXiv preprint*, 2025, doi:10.48550/arXiv.2506.04838.

[5] B. Agarwal, H. S. Jomraj, S. Kaplunov, J. Krolick, and V. Rojkova, "RAGulating compliance: A multi-agent knowledge graph for regulatory QA," *arXiv preprint*, 2025, doi:10.48550/arXiv.2508.09893.

[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020, doi:10.48550/arXiv.2005.11401.

[7] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," *Proc. 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019, doi:10.18653/v1/D19-1410.

[8] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, "Dense passage retrieval for open-domain question answering," *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, doi:10.48550/arXiv.2004.04906.

[9] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval augmentation reduces hallucination in conversation," *arXiv preprint*, 2021, doi:10.48550/arXiv.2104.07567.