Volume 4, No. 2 February 2025 - (443-452)
p-ISSN 2980-4868 | e-ISSN 2980-4841
https://ajesh.ph/index.php/gp

OSCAT: A Comprehensive Tool for Automated CIS Benchmark Auditing

Ahmad Sholihin[1*], Muhammad Salman[2]
Universitas Indonesia, Indonesia
Emails: ahmad.sholihin31@ui.ac.id[1], muhammad.salman@ui.ac.id[2]

ABSTRACT

Servers play a crucial role in modern digital infrastructure, but misconfiguration in their settings causes security gaps that can potentially lead to serious threats such as data leakage and service disruptions. The Center for Internet Security (CIS) has developed benchmark standards to improve server security, but the manual audit process for server configurations is often time-consuming and prone to human error. This study aims to develop and evaluate the One Stop CIS-Benchmark Auditing Tool (OSCAT), a CIS Benchmark-based server configuration audit automation tool. The research method used is an experimental approach with testing on the Red Hat Enterprise Linux 7 (RHEL 7) operating system using the CIS Benchmark version 3.1.1 security standard. OSCAT is designed with an architecture consisting of Audit Files, Audit Engine, Output Processor, Report Generator. Testing was carried out by auditing 248 security controls and comparing the results with a manual audit to assess the effectiveness of OSCAT in improving the efficiency and accuracy of the audit process. The results showed that OSCAT is able to automate the server configuration audit process with a high level of accuracy, reducing manual workload and increasing efficiency in identifying and handling security risks. Of the 248 security controls tested, OSCAT successfully classified the audit results into Pass, Fail, and Manual Review categories, with advantages in speed and consistency over manual methods. The implications of this research show that OSCAT can be an effective solution for organizations to improve the security of their systems through faster and more accurate automated audits.

Keywords: Automation, CIS benchmarks, Cybersecurity, OSCAT Server Configuration Audit.

INTRODUCTION

As the number of internet users in Indonesia continues to grow, reaching 221.5 million or 79.5% of the total population in 2023 (APJII, 2024), the role of servers in powering digital ecosystems becomes increasingly critical. From web servers hosting websites to database servers storing critical information, servers are essential components of modern internet infrastructure (Fox & Hao, 2017). However, misconfigurations in server settings can expose vulnerabilities, leading to significant harm such as security breaches, data loss, and service disruptions, even might also lead to catastrophic data leakage issues for enterprises (Loureiro, 2021).

Server configuration plays a pivotal role in ensuring the security and reliability of information systems. Misconfigurations can lead to a wide range of security vulnerabilities, such as weak passwords, open ports, and outdated software. These vulnerabilities can be exploited by malicious actors to gain unauthorized access to systems, steal data, or launch further attacks. A notable example is the Heartbleed vulnerability (Bug, 2024); (CISA.gov, 2024), which affected millions of servers and exposed sensitive information.

Server hardening involves implementing various security measures to protect servers from attacks. This includes enforcing strong password policies, disabling unnecessary services and ports, and regularly patching systems with the latest security updates. Additionally, network security measures, such as strong encryption protocols and network segmentation, are crucial to protect sensitive data. Implementing robust user access controls, including multi-factor authentication and least privilege principles, can further bolster security (Kumar & Soumya, 2014).

To ensure consistent and effective security practices, various industry standards and benchmarks have been developed. The Center for Internet Security (CIS) provides a comprehensive set of benchmarks for securing IT systems and controls. CIS benchmarks offer detailed guidelines for configuring servers, network devices, and applications to minimize vulnerabilities (Security, 2024a).

While manual auditing can be effective, it is time-consuming and prone to human error. To address these challenges, automated tools have emerged as a powerful solution. One Stop CIS-Benchmark Auditing Tool (OSCAT) is one such tool designed to streamline the server configuration auditing process.

OSCAT leverages the CIS benchmarks to automate the process of assessing server configurations. By automating the auditing process, OSCAT helps organizations to significantly improve efficiency, accuracy, and consistency in identifying and mitigating security risks of server configuration.

Based on the above background, the purpose of this study is to develop and evaluate the effectiveness of the One Stop CIS-Benchmark Auditing Tool (OSCAT) as a CIS Benchmark-based server configuration audit automation tool. Thus, the benefit of this study is to contribute to organizations in improving the security of their systems through more efficient server configuration audits. With the implementation of OSCAT, organizations can save time and resources in conducting server security audits, as well as ensuring compliance with security standards set by CIS Benchmark. In addition, this research also provides insights for cybersecurity developers in developing more sophisticated tools to overcome challenges in server security management.

RESEARCH METHOD

This study used an experimental method with a quantitative approach to develop and test the One Stop CIS-Benchmark Auditing Tool (OSCAT) as a CIS Benchmark-based server configuration audit automation tool. The population in this study includes Linux-based operating systems used for enterprise server needs, with the research sample being the Red Hat Enterprise Linux 7 (RHEL 7) operating system tested in a virtual environment. Data was collected through the results of an automated audit generated by OSCAT of 248 security controls based on CIS Benchmark version 3.1.1. In addition, audit reports in Excel and PDF formats were used as documentation, and manual audit results were used as a comparison to validate the accuracy of OSCAT.

The data sources used in this study consisted of primary and secondary data. The primary data was obtained directly from the audit results conducted by OSCAT, while the secondary data came from the CIS Benchmark version 3.1.1 security standards and the results of the manual audit. To ensure valid research results, OSCAT audited the server security configuration by recording the security status in the Pass, Fail, and Manual Review categories. This audit process includes stages ranging from OSCAT architecture design, system implementation, to analysis of audit results by comparing the effectiveness of OSCAT against manual audits.

The data analysis technique is carried out quantitatively with several approaches. Descriptive analysis is used to calculate the number of security controls that pass (Pass), fail (Fail), and require manual review (Manual Review). In addition, a comparison was made of the effectiveness of OSCAT with the manual audit method to measure the speed and accuracy in identifying security vulnerabilities. An evaluation of the advantages and limitations of OSCAT was also carried out based on the results of the pilot test to assess the extent to which OSCAT can improve the efficiency and accuracy of the CIS Benchmark-based system security audit process..

RESULT AND DISCUSSION

CIS Benchmark

The Center for Internet Security (CIS) is a nonprofit organization dedicated to safeguarding IT systems and data (Prastika et al., 2019). Founded in 2000, CIS collaborates with a global community of experts to develop and maintain best practices for cybersecurity (Microsoft, 2024). Their primary goal is to help individuals, businesses, and governments protect themselves from cyber threats (Cisecurity.org, 2024).

One of the primary products that CIS offers is CIS Benchmark. CIS Benchmarks has detailed configuration guidelines for securing various IT systems and software. These benchmarks provide a comprehensive set of security controls to help organizations harden their systems and reduce vulnerabilities (Services, 2024).

CIS Benchmarks are regularly updated (Security, 2024b) to address emerging threats and vulnerabilities. By using CIS Benchmarks, organizations can significantly improve their cybersecurity posture and protect their valuable assets.

Proposed Architecture

The architecture of the proposed work is shown in Figure 1. The work consists of four components i.e., Audit Files, Audit Engine, Output Processor, Report Generator. The following section will describe each of the component.

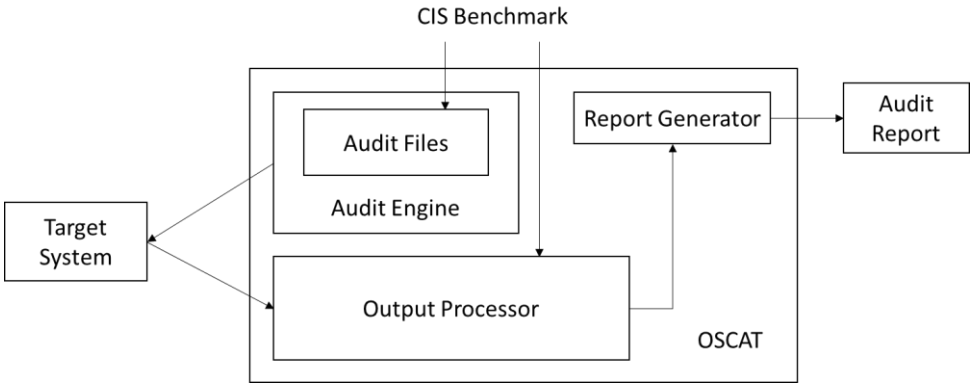

Figure 1. OSCAT Architecture

Audit Files

The OSCAT architecture has a set of files for each benchmarks containing audit commands, scripts, and/or queries, stored in a special folder where it keeps all the instructions needed to check if a system is secure according to the CIS Benchmark. Each instruction is saved in a separate file, and the file name matches the number of the security rule it is checking.

To make sure the results are as accurate as possible, OSCAT uses the exact same instructions that are recommended in the CIS Benchmark. This way, there is less chance of mistakes happening because someone accidentally changed an instruction or script modification.

Audit Engine

The audit engine is the fundamental component driving the OSCAT audit process. It initiates the audit by authenticating to the target system, ensuring secure access and authorization. Once authenticated, the engine transfers the necessary audit files to the target system, preparing the environment for the execution phase.

The audit engine then executes the transferred audit files, which contain scripts designed to assess the system's security posture against predefined benchmarks. Upon completion of the execution phase, the engine meticulously logs the results of the audit, including any identified vulnerabilities or misconfigurations, into the OSCAT database. This comprehensive logging enables detailed analysis and reporting, facilitating informed decision-making and remediation efforts.

Output Processor

The output processor plays as “the judge" for the security audit. After the audit engine has finished running all the checks, the output processor takes the results and decides if the system passed or failed each individual test by the following process:

1) Collects Results, which will gather all the information produced by the audit.
2) Compares to Standards, which will check these results against the correct answers (expected output) stored in the database.
3) Makes a Decision, which will decide if the results match the expected outcomes, the control is marked as a "pass"; otherwise, it's a "fail". In controls that marks with “manual” however, the decision should be decided by human. OSCAT will only show the output as a consideration for security analyst to decide whether it is “pass” or “fail”.

Report Generator

The report generator is a vital tool that consolidates the audit results into a clear and concise format. It generates detailed reports in either Excel or PDF format, encompassing essential information such as the control number, title, description, rationale, result, output, and remediation recommendations. This comprehensive report provides a clear overview of the system's security posture, enabling informed decision-making and effective remediation efforts.

System Flowchart

Fig. 2 shows the flowchart of OSCAT. Like other common scanning tools, the user needs to input scan details such as scan name, target details, and credentials. In cases where the target is a database system, the user also needs to input both database and server credentials. This is because assessing a database system requires some controls to be checked at the OS level. For example, in CIS Oracle MySQL Community Server 5.7 Benchmark v2.0.0, control "1.1 Place Databases on Non-System Partitions" requires an OS-level command to verify if the database server is placed on a non-system partition. Therefore, in addition to database credentials, OS credentials are also required to assess such controls.

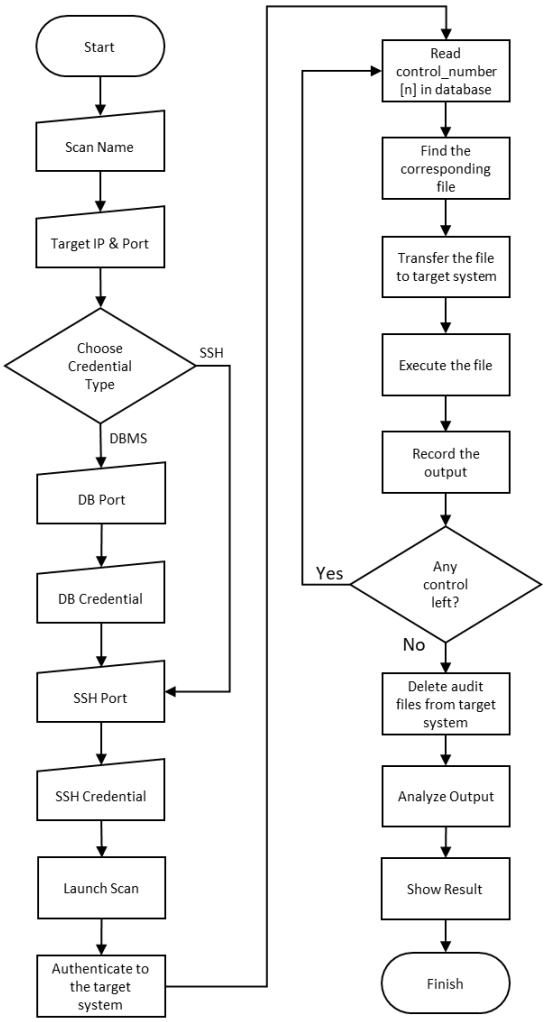

Figure 2. OSCAT's Flowchart

Additionally, OSCAT iterates through each control number in the database. For each iteration, OSCAT locates the corresponding file with the same name as the control number, transfers it to the target system, executes it, and logs the response. The loop terminates when all control numbers have been processed. This ensures that all controls are executed and reported accurately. After executing all controls, OSCAT removes all audit files from the target system to minimize file traces.

Result Analysis

The core of OSCAT's result analysis lies in comparing the actual output of each executed control against a predefined expected output derived from the corresponding CIS Benchmark guideline. To facilitate this comparison, OSCAT employs eight distinct algorithms, designed to address the various types of checks specified in the CIS Benchmark guidelines. These algorithms are detailed below:

Table 1. Result Analysis Algorithms

Algorithm	Description
Null	Pass if the output string O is empty.
Not Null	Pass if the output string O is not empty.
Contain	Pass if the output string O contains the expected substring E.
Does Not Contain	Pass if the output string O does not contain the expected substring E.
Exact	Pass if the output string O is exactly equal to the expected string E.
More Than	Pass if the numerical output O is greater than the expected numerical value E.
Less Than	Pass if the numerical output O is less than the expected numerical value E.
Manual	Some controls require human interpretation or subjective assessment. These controls are flagged for manual review by a security auditor.

Following the audit process, OSCAT generates a comprehensive report in Excel format. This report details each evaluated control, including its number, title, description, rationale, the actual output observed on the target system, the expected output according to the CIS Benchmark, the audit result (Pass/Fail), and specific remediation recommendations for any identified non-compliance. The tabular format of the report facilitates efficient review and analysis of the audit findings by security analysts.

To evaluate OSCAT's effectiveness, a series of audits were conducted on a freshly installed Red Hat Enterprise Linux 7 (RHEL 7) system within a virtualized environment. Linux, especially RHEL 7, was selected due to its widespread use in server environments and its relevance to enterprise infrastructure to support the increasingly rapid development of internet technology (Sedano & Salman, 2021). The RHEL 7 system was deployed with its default installation configuration. The audits were performed against version 3.1.1 of the CIS Benchmark for Red Hat Enterprise Linux 7, which comprises 248 controls organized into six distinct domains. The results of these audits, broken down by domain, are presented in the table below:

Table 2. Audit Result On Rhel 7

Domain	Pass	Fail	Manual	Total
Access, Authentication and Authorization	27	22	0	49
Initial Setup	32	24	3	59
Logging and Auditing	4	27	2	33
Network Configuration	33	12	3	48
Services	26	1	1	28
System Maintenance	28	0	3	31
Grand Total	**150**	**86**	**12**	**248**

The audit results presented in Table 2 illustrate the effectiveness of OSCAT in evaluating the security compliance of a Red Hat Enterprise Linux 7 (RHEL 7) system against CIS Benchmark version 3.1.1. The table categorizes the findings into Pass, Fail, and Manual Review, based on OSCAT's automated assessment of 248 security controls across six different security domains.

From the results, the highest number of passing controls (33 passes) was observed in the Network Configuration domain, indicating that RHEL 7, in its default installation, aligns well with CIS Benchmark recommendations for securing network settings. Conversely, the Logging and Auditing domain exhibited the highest number of failed controls (27 fails), highlighting significant gaps in security logging and monitoring configurations, which are critical for detecting unauthorized access and anomalies in server activities. The Access, Authentication, and Authorization domain also showed a considerable number of failed controls (22 fails), suggesting that additional configuration adjustments are required to strengthen user access policies and authentication mechanisms.

Furthermore, Manual Review was necessary for 12 controls, particularly in domains such as Initial Setup (3), Logging and Auditing (2), Network Configuration (3), Services (1), and System Maintenance (3). These controls required human intervention to verify compliance due to their complexity or dependency on organizational policies. Overall, 150 security controls (60.5%) passed, demonstrating OSCAT's effectiveness in identifying security-compliant configurations, while 86 controls (34.7%) failed, indicating areas that need improvement. The manual reviews (4.8%) highlight cases where automated analysis alone was insufficient, reinforcing the need for human expertise in certain security evaluations.

Discussion

The results indicate that OSCAT provides a highly effective and efficient method for auditing server configurations based on CIS Benchmarks. The high pass rate in several domains demonstrates its accuracy in identifying compliance with security standards. However, the failure rates in specific categories such as 'Logging and Auditing' highlight areas where misconfigurations are prevalent and require further attention. This finding is consistent with prior research, such as that by (Balta et al., 2025), which emphasizes the importance of rigorous log management practices in cybersecurity. Additionally, the need for manual reviews in some controls aligns with previous studies by (Nagar, 2018), which suggest that automation cannot entirely replace human judgment in complex security assessments.

CONCLUSION

This research presented the design and implementation of OSCAT, a comprehensive automated host configuration auditing tool based on CIS Benchmark. OSCAT is designed to automate the host configuration auditing process, reducing manual effort and increasing efficiency. The core functionalities of OSCAT include authentication, file transfer, file execution, output processing, and result reporting, all of which have been discussed in detail. By automating script execution and result analysis, OSCAT empowers security analysts to focus on high-level tasks and strategic decision-making rather than low-level manual auditing.

While OSCAT offers a robust foundation for automated CIS Benchmark auditing, several areas for future development remain. These include extending OSCAT's capabilities to support a wider range of target systems, such as Windows Servers, Microsoft SQL Server, and network appliances, as well as integrating privilege escalation to audit hardened systems with restricted root access. Additionally, developing advanced reporting and visualization features can provide more insightful analysis of audit results, while integrating OSCAT with popular vulnerability management tools can enhance the remediation process. By addressing these areas, OSCAT can evolve into an even more powerful and versatile tool for securing IT environments.

REFERENCES

APJII. (2024). *APJII Jumlah Pengguna Internet Indonesia Tembus 221 Juta Orang*. Apjii.or.Id. https://apjii.or.id/berita/d/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang

Balta, D. D., Kaç, S. B., Balta, M., Oğur, N. B., & Eken, S. (2025). Cybersecurity-aware log management system for critical water infrastructures. *Applied Soft Computing*, *169*, 112613.

Bug, H. (2024). *The Heartbleed Bug*. Heartbleed.Com. https://heartbleed.com/

CISA.gov. (2024). *OpenSSL "Heartbleed."* Www.Cisa.Gov. https://www.cisa.gov/news-events/alerts/2014/04/08/openssl-heartbleed-vulnerability-cve-2014-0160

Cisecurity.org. (2024). *CIS Critical Security Controls*. Www.Cisecurity.Org. www.cisecurity.org

Fox, R., & Hao, W. (2017). *Internet infrastructure: networking, web services, and cloud computing*. CRC Press.

Kumar, K. P., & Soumya, E. (2014). The Major Traits of Cyber Security: Case Study on Server Hardening. *International Journal of Advanced Trends in Computer Science and Engineering*, *3*(1), 196–200.

Loureiro, S. (2021). Security misconfigurations and how to prevent them. *Network Security*, *2021*(5), 13–16.

Microsoft. (2024). *Center for Internet Security (CIS) Benchmarks*. Learn.Microsoft.Com. https://learn.microsoft.com/en-us/compliance/regulatory/offering-cis-benchmark

Nagar, G. (2018). Leveraging Artificial Intelligence to Automate and Enhance Security Operations: Balancing Efficiency and Human Oversight. *Valley International Journal Digital Library*, 78–94.

Prastika, D. P., Triyono, J., & Lestari, U. (2019). *Audit Dan Implementasi Cis Benchmark Pada Sistem Operasi Linux Debian Server (Studi Kasus: Server Laboratorium Jaringan Dan Komputer 6 Institut Sains & Teknologi Akprind Yogyakarta)*.

Security, C. for I. (2024a). *CIS Benchmarks*. Cisecurity.Org. https://www.cisecurity.org/cis-benchmarks/

Security, C. for I. (2024b). *Customize CIS Benchmarks with New Tailoring Feature in CIS WorkBench*. Www.Cisecurity.Org. available: https://www.cisecurity.org/insights/blog/customize-cis-benchmarks-cis-workbench

Sedano, W. K., & Salman, M. (2021). Auditing Linux operating system with center for internet security (CIS) standard. *2021 International Conference on Information Technology (ICIT)*,

466–471.
Services, A. W. (2024). *What is CIS Benchmarks?* Aws.Amazon.Com. https://aws.amazon.com/what-is/cis-benchmarks/