2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) | 979-8-3503-8944-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACCAI61061.2024.10602274

System Hardening using CIS Benchmarks

Ambika P. H[1], and *Dr.G.Sujatha[2]

Department of Networking and Communications, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai, India

E-mail :ap9672@srmist.edu.in, sujathag@srmist.edu.in
**Corresponding author: Dr.G.Sujatha*

Abstract-Automated system hardening with Center for Internet Security (CIS) benchmarks is crucial for enhancing cybersecurity, and organizations face significant hurdles in this endeavor. Manual configuration is complex, error-prone, and time-consuming, making it difficult to scale as IT environments grow. The static nature of manual processes inhibits the ability to adapt quickly to evolving threats, leaving systems vulnerable. Resource limitations often hinder automation efforts, leading to incomplete or delayed compliance. Additionally, manual documentation and tracking of configuration changes present accountability challenges in incident response scenarios. To address these issues, organizations require an efficient and scalable solution for automating CIS benchmark compliance. Such a solution can streamline the hardening process, reduce errors, enhance security, and facilitate continuous monitoring and reporting, ensuring systems are fortified against both known and emerging threats.

Keywords— Automated System Hardening, CIS Benchmarks, Ansible, Security Configuration Management, Windows Server Security, RHEL Security, Vulnerability Management, Compliance, Security Automation, Continuous Monitoring.

I.INTRODUCTION

Automated system hardening, guided by CIS benchmarks [1], streamlines cybersecurity efforts by using technology to automatically configure and maintain systems according to industry standards. This approach significantly reduces the risk of human error, enhances efficiency, and adapts to the ever-changing threat landscape. Traditionally, implementing CIS benchmarks demanded manual effort, leaving room for inadvertent errors and making it an arduous and time-consuming task. Automation simplifies this process by orchestrating the application of security measures consistently across a multitude of systems. This not only expedites the hardening process but significantly enhances its precision. The core principle underlying CIS benchmarks is adherence to best practices, informed by extensive research and expertise. These practices encompass a broad spectrum of security measures, from access control to network configurations, designed to create robust defense layers within the Windows environment. However, the scale and complexity of modern IT infrastructures demand a more efficient approach, and automation offers just that.

By employing automation tools like Ansible [2], organizations can automatically configure security settings, conduct regular vulnerability scans, and remediate identified issues. This proactive approach reduces the window of vulnerability, minimizes the attack surface, and ensures systems remain in a secure state. Moreover, it enables organizations to rapidly respond to emerging threats and evolving compliance requirements.

II.RELATED WORKS

A.LITERATURE SURVEY ON AUTOMATED IMPLEMENTATION OF WINDOWS-RELATED SECURITY-CONFIGURATION GUIDES

The literature survey conducted by Stockle,¨Grobauer, and Pretschner [3] focuses on the automation of security configuration implementation in relation to Windows-based systems. The survey recognizes the significance of established security guides, including the CIS benchmarks and Microsoft's Security Compliance Manager [1], which offer recommended practices for safeguarding Windows environments. The manual implementation of these security configurations is acknowledged as a labor-intensive and error-prone process that often results in inconsistent configurations across systems. As a solution to these challenges, the authors propose the adoption of automation techniques. The survey provides an overview of existing approaches and tools employed in automating the implementation of security configurations. It explores the utilization of scripting languages, such as PowerShell, and introduces the concept of system hardening, which involves automating the application of predefined security configurations. Various tools and frameworks aimed at automating security configuration implementation are also discussed, including Microsoft's Security Compliance Manager, Group Policy Objects (GPOs), and configuration management frameworks like Ansible and Chef.

The literature survey provides valuable insights into the significance of security configuration guides for Windows systems and investigates the application of automation in implementing these configurations. It introduces various tools, frameworks, and approaches for automating security configuration implementation, highlighting both the advantages and challenges associated with automation

in this context. This survey serves as a valuable resource for organizations seeking to improve the security of their Windows environments through automated security configuration implementation.

B. Literature survey on Windows 10 security hardening using device guard whitelisting and AppLocker blacklisting

The literature survey conducted by Durve and Bouridane [4] focuses on enhancing Windows 10 security through the combined use of Device Guard whitelisting and AppLocker blacklisting techniques. The authors acknowledge the growing need for robust security measures in modern operating systems, particularly Windows 10. They emphasize the importance of leveraging built-in features such as Device Guard and AppLocker to strengthen system security. Device Guard is a security feature that employs code integrity policies to restrict the execution of only trusted applications on Windows 10 devices. The authors delve into the implementation of Device Guard in enforcing whitelisting policies, which entail maintaining a list of authorized applications while preventing unauthorized or malicious software from running.

AppLocker, on the other hand, is a security feature that allows administrators to create rules governing which applications can run on a Windows 10 device. The authors explore the use of AppLocker in implementing blacklisting policies, which involve explicitly prohibiting the execution of specific applications or software. By combining Device Guard whitelisting and AppLocker blacklisting techniques, organizations can establish a multi-layered defense strategy that effectively mitigates the risk of unauthorized or malicious software execution while accommodating legitimate applications. The survey also acknowledges the challenges associated with implementing and managing Device Guard and AppLocker policies. These challenges encompass the complexity of policy creation, potential conflicts with legitimate applications, and the need for regular policy updates to address emerging threats adequately. Furthermore, the authors stress the significance of continuous monitoring, audits, and policy reviews to ensure the ongoing effectiveness of the security hardening measures.

The literature survey highlights the importance of Windows 10 security enhancement through the combined use of Device Guard whitelisting and AppLocker blacklisting. It provides valuable insights into the benefits, challenges, and recommended practices for implementing and managing these security features. This survey serves as a reliable resource for organizations seeking to bolster their Windows 10 security through effective security hardening measures.

C. Literature survey on Auditing Linux Operating System with Center for Internet Security (CIS) Standard

The literature survey conducted by Sedano and Salman [5] focuses on the auditing of Linux operating systems using the Center for Internet Security (CIS) standards [1]. The authors acknowledge the increasing importance of securing Linux operating systems due to their widespread use in various environments. The CIS standards provide comprehensive guidelines and best practices for securing Linux systems. The survey highlights the need for auditing tools that can assess the compliance of Linux systems with the CIS standards. These tools help organizations identify security vulnerabilities, misconfigurations, and deviations from the recommended security settings. The authors propose the development of an auditing tool specifically designed for Linux operating systems, utilizing the CIS standards as the benchmark. This tool would automate the auditing process and provide organizations with a systematic way to evaluate the security posture of their Linux systems.

The survey outlines the benefits of using such an auditing tool, including improved efficiency, accuracy, and consistency. By automating the auditing process, organizations can save time and resources while ensuring a higher level of security and compliance. Additionally, the authors address the challenges associated with auditing Linux systems using the CIS standards. These challenges include the complexity of Linux configurations, the diversity of Linux distributions, and the need for continuous updates to accommodate new CIS standards and evolving security threats.

The survey emphasizes the significance of auditing Linux operating systems based on the CIS standards. It highlights the benefits of using an automated auditing tool to assess compliance, identify vulnerabilities, and enhance the security of Linux systems. The survey serves as a valuable resource for organizations seeking to strengthen the security of their Linux environments by implementing an effective auditing solution aligned with the CIS standards.

III. Methodology

A. Proposed Solution

Automated system hardening using CIS benchmarks with Ansible is a robust approach to fortifying the security of IT systems. It harnesses the power of Ansible automation [2], coupled with the comprehensive security recommendations outlined in the Center for Internet Security (CIS) benchmarks. By creating Ansible playbooks and roles [6], this solution automates the application of CIS guidelines to target systems, thereby reducing manual configuration efforts and minimizing human errors. Customization options allow tailoring security configurations to specific organizational

requirements, ensuring flexibility. Continuous monitoring tools like Nessus or OpenSCAP continuously assess system security, automatically remediating any deviations from the desired state. Comprehensive logs and reports provide a trail of security changes and compliance status. Ultimately, this solution enhances security, consistency, and compliance across the IT infrastructure while significantly improving operational efficiency.

IV. Architecture Diagram

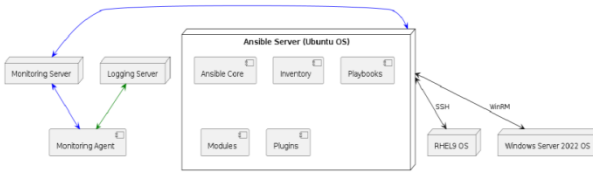

Fig. 1. Architecture Diagram

A. Ansible Core

Ansible Core acts as the engine that orchestrates automation tasks across your infrastructure. It's the brains behind the operation, responsible for:

- Parsing Playbooks
- Communication and Execution
- Variable Management
- Error Handling and Recovery

B. Inventory

The inventory [7] serves as a blueprint for your managed infrastructure. It defines the set of hosts and groups that Ansible will target for automation tasks. Ansible offers flexibility in defining the inventory:

- Static Files: Inventories can be created as simple text files containing hostnames, IP addresses, or group definitions.
- Dynamic Inventories: Ansible supports dynamic inventories that can be generated from external sources like cloud APIs, databases, or configuration management tools. This allows for automatic discovery and management of infrastructure changes.
- Inventory Plugins: Ansible offers a rich set of inventory plugins that extend the capabilities of basic inventory files. These plugins can source inventory data from various platforms like AWS, Azure, or custom scripts.

C. Playbooks

Playbooks [8] are the heart of Ansible automation. They are YAML files that define the configuration you want to achieve on your managed hosts. A playbook is essentially a recipe containing a series of tasks executed in the desired order. Playbooks typically include:

- Hosts: The specific targets (hosts or groups) for the automation tasks defined within the playbook.
- Tasks: These are the individual actions that Ansible will perform on the managed hosts. Tasks leverage Ansible modules to achieve specific configuration or management goals.
- Roles: Playbooks can leverage roles, which are reusable modules that encapsulate functionalities for complex configurations. Roles promote modularity and code reuse across playbooks.
- Variables: Variables, defined within playbooks or included from external sources, allow for dynamic configuration and customization.

D. Modules

Modules in Ansible [9]represent the fundamental units of automation tasks, each serving as a self-contained script written in various programming languages, predominantly Python. These scripts are designed to interact with the underlying operating system, applications, or cloud services running on managed hosts.

- File Management
- Package Management
- User and Group Management
- Service Management
- Network Configuration

E. Plugins

Ansible's plugin [9] architecture allows for extending its functionalities beyond core features. Plugins can be categorized into various types:

- Inventory Plugins
- Connection Plugins
- Module Utilities
- Callback Plugins

F. Linux OS [RHEL9]

Automating system hardening in RHEL 9 [10] is crucial for enhancing security. By leveraging Ansible, administrators can deploy security configurations efficiently. Ansible enables the creation of custom playbooks based on industry standards like CIS benchmarks. These playbooks ensure consistency and reliability while meeting specific organizational security needs. Automation offers benefits such as increased efficiency, reduced errors, and better compliance with security standards. It minimizes manual tasks and enforces standardized security measures, thus reducing the risk of breaches and bolstering defense against cyber threats in modern IT environments.

G. Server OS [Windows Server 2022]

A Server OS i.e Windows server 2022 [11] is designed to operate on servers, which are computers dedicated to providing services or resources to other computers, often over a network. Server OSs prioritize stability, reliability, and performance in serving multiple users or applications concurrently. They are typically managed remotely, either through command-line interfaces (CLI) or web-based administration tools, rather than through a graphical user interface (GUI).

V. Continuous Monitoring

Nessus is a critical component of continuous monitoring and system hardening efforts aligned with CIS benchmarks. It continuously scans your systems, identifies vulnerabilities and compliance gaps, provides real-time alerts, and offers recommendations for remediation. By using Nessus in conjunction with CIS benchmarks, organizations can effectively monitor, assess, and harden their systems, enhancing overall security and compliance posture. Below diagram shows the possible security defenses achieved by hardening the system fig 2.

- Scheduled Scanning
- Vulnerability Detection
- Compliance Checks
- Real-Time Alerts
- Integration with Remediation Tools

VI. Logging And Auditing

Nessus [12] provides auditing and logging features that help organizations track and document vulnerability assessment activities, findings, compliance status, and remediation efforts. These logs and reports are essential for auditing and compliance purposes, providing a clear record of the security posture of an organization's IT environment over time.

Fig. 2. Hardening Significant Advancement

By architecting tailored Ansible playbooks and roles [13], this solution seamlessly and consistently enforces the CIS guidelines across a diverse spectrum of operating systems, including Windows Server 2022, RHEL9. The net result is a reduction in manual configuration efforts, a minimization of the potential for human errors, and a stronger security posture for your organization.

Windows Server 2022, facilitates a stringent security framework. By applying CIS benchmarks through Ansible, a variety of tasks are automated, including the meticulous curation of service restrictions, the precise configuration of audit policies, the ironclad enforcement of password policies, and the deft management of Windows Firewall settings. With Ansible in the driver's seat, these security configurations are consistently and efficiently disseminated across Windows Server 2022 environments [11].

The CIS benchmarks provide invaluable guidelines for securing these diverse operating systems. These benchmarks encompass a wide array of security recommendations, from essential system configurations to advanced security measures. For Windows Server 2022 [11], the benchmarks cover crucial areas such as system hardening, account policies, network security, and audit policies, ensuring that these servers are resilient against a range of cyber threats. In the case of RHEL9 [10], the benchmarks address user access controls, software restrictions, system updates, and browser security, fostering a robust defense against security vulnerabilities commonly targeted in enduser environments [14]. Compliance with these CIS benchmarks is not only an industry best practice but is often required for regulatory compliance, making them a foundational resource for organizations looking to fortify their Windows-based systems against evolving cybersecurity threats.

By following the CIS benchmarks, organizations can significantly enhance their cybersecurity posture, reduce vulnerabilities, and maintain alignment with established security standards.

By harnessing Ansible in tandem with CIS benchmarks, the hardening process encompasses user access control fine-tuning, the meticulous implementation of software restrictions, proactive system update management, and the establishment of impregnable browser security. The automation prowess of Ansible ensures the uniform application of these configurations across diverse workstation landscapes.

Comprehensive Logging and Reporting solution offers an array of comprehensive logs and reports. These invaluable resources chronicle all security changes, be it in Windows Server 2022, RHEL9 with reference to Table 1. These logs furnish a crystal-clear view of the compliance status, facilitating audits and validating the consistent enforcement of security policies across the IT ecosystem.

Table I Hardening Comparison

Operating System	Hardening Method	Time taken
Windows Server 2022	Manual	180 minutes
	Automated	30 minutes
RHEL9	Manual	30 minutes
	Automated	12 minutes

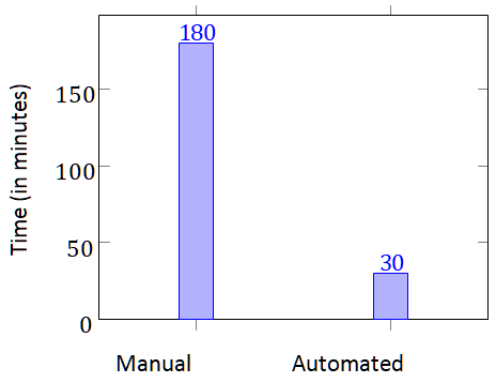

Fig 3. Windows Hardening

In essence, the adoption of automated system hardening through CIS benchmarks and Ansible is an indispensable strategy for bolstering the security of IT systems, regardless of the Windows operating system version. It forges an environment marked by heightened security, unwavering consistency, and unwavering adherence to industry benchmarks. In the process, it empowers organizations to take proactive control of their systems, nurturing a resilient security posture while upholding a strong commitment to industry-recognized benchmarks.

VII. Result And Discussion

In our investigation, we found some really positive outcomes when we used Ansible and CIS Benchmarks to automate system hardening on Windows Server 2022 and RHEL 9. By automating the implementation of CIS Benchmarks, we managed to significantly reduce the manual work needed for system hardening compared to the traditional manual setup methods Fig 3 and Fig 4. It was really satisfying to see how much more consistent our security configurations became across all our test machines. And, of course, we definitely noticed a tangible improvement in our compliance with important security standards.

These results really highlight the benefits of automation in cy-[3] bersecurity. Ansible playbooks made it so much easier to apply CIS Benchmarks, which helped minimize the chances of human error and[4] ensured that our security measures were consistent across the board. Plus, automation made it much simpler to keep up with the ever-[5]changing landscape of security regulations.

[6] Our research suggests that organizations can really enhance their cybersecurity efforts by embracing automation for system hardening with CIS Benchmarks and Ansible. Looking ahead, it would be[7] interesting to explore how we can integrate this approach with other tools for vulnerability management, creating an even stronger overall[8] security strategy. [9]

VIII. Conclusion

[10] The integration of automated system hardening through the application of CIS benchmarks with Ansible is a potent and adaptive approach to fortify the security of IT systems.

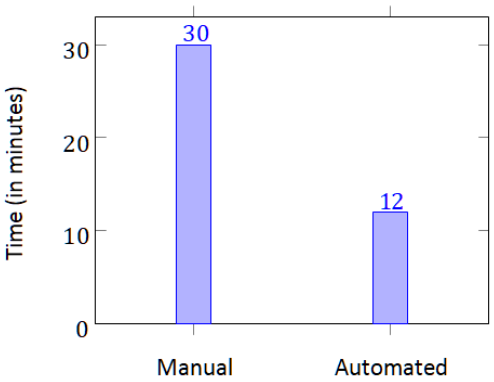

Fig 4. RHEL Hardening

It brings together the efficiency of Ansible's automation capabilities with the depth of security wisdom provided by the Center for Internet Security (CIS) benchmarks. This dynamic synergy is instrumental in ensuring the security and compliance of various Windows operating systems, including Windows Server 2022 and RHEL9. This approach not only streamlines the implementation of CIS recommendations but also significantly reduces manual configuration efforts, mitigating the potential for human errors. The flexibility to customize security configurations to suit an organization's specific requirements ensures that security measures align seamlessly with unique operational needs. The incorporation of continuous monitoring tools, such as Nessus and OpenSCAP [15], ensures that security is not just a one-time effort but an ongoing commitment. These tools continuously assess security postures, promptly detecting and remediating deviations from the desired state. The comprehensive logs and reports generated by this approach are invaluable for auditing and validation, providing transparency into compliance and security changes.

The combination of CIS benchmarks with Ansible automation enhances security, consistency, and compliance across the IT infrastructure. It optimizes operational efficiency, minimizes security risks, and empowers organizations to not only secure their systems but also maintain a resilient and unwavering security posture. This approach embodies a commitment to industry standards and best practices, safeguarding organizations in an ever-evolving threat landscape.

REFERENCES

[1] "Cis benchmarks," https://www.cisecurity.org/cis-benchmarks, accessed: April 5, 2024.

[2] Ansible Documentation. (latest) Windows Setup. Accessed: April 5, 2024. [Online]. Available: https://docs.ansible.com/ansible/latest/osguide/windowssetup.html P. Stockle, B. Grobauer, and A. Pretschner, "Automated implementation of windows-related security-configuration guides," pp. 598–610, 2020.

[3] R. Durve and A. Bouridane, "Windows 10 security hardening using device guard whitelisting and applocker blacklisting," pp. 56–61, 2017.

[4] A. Khurat and P. Sangkhachantharanan, "An automatic networking device auditing tool based on cis benchmark," pp. 409–412, 2021.

[5] Ansible Documentation. (latest) Getting Started with Ansible Playbooks. Accessed: April 5, 2024. [Online]. Available:

[6] https://docs.ansible.com/ansible/latest/gettingstarted/getstartedplaybook.html

[7] Ansible. Ansible Inventory. Accessed: April 5, 2024. [Online]. Available: https://docs.ansible.com/ansible/latest/inventoryguide/introinventory.html

[8] Ansible Playbook Guide. Accessed: April 5, 2024. [Online]. Available: https://docs.ansible.com/ansible/latest/playbookguide/playbooks.html

[9] Ansible Module. Accessed: April 5, 2024. [Online]. Available: https://docs.ansible.com/ansible/latest/plugins/module.html

[10] Center for Internet Security (CIS). CIS RHEL Benchmark. Accessed: April 5, 2024. [Online]. Available: https://www.cisecurity.org/benchmark/redhatlinux

[11] CIS Microsoft Windows Server Benchmark. Accessed: April 5, 2024. [Online]. Available: https://www.cisecurity.org/benchmark/microsoftwindowsserver

[12] Tenable. (latest) Nessus Essentials. Accessed: April 5, 2024. [Online]. Available: https://www.tenable.com/products/nessus/nessus-essentials

[13] W. K. Sedano and M. Salman, "Auditing linux operating system with center for internet security (cis) standard," pp. 466–471, 2021.

[14] OpenSCAP vuln. OpenSCAP Vulnerability. Accessed: April 5, 2024. [Online]. Available: https://www.open-scap.org/resources/documentation/performvulnerability-scan-of-rhel-6-machine/

[15] OpenSCAP documentation. OpenSCAP Starter. Accessed: April 5, 2024. [Online]. Available: https://static.open-scap.org/openscap1.3/oscapusermanual.html