SecureSys: Universal Hardening and Compliance Toolkit

Chirayu Rathi
Dept. of Information Technology
MPSTME NMIMS University
Mumbai, India
chirayu.rathi65@nmims.in

Aditi Jamsandekar
Dept. of Information Technology
MPSTME NMIMS University
Mumbai, India
aditi.jamsandekar89@nmims.in

Siddhi Jani
Dept. of Information Technology MPSTME
NMIMS University Mumbai, India
siddhi.jani86@nmims.in

Dr. Sneha Deshmukh
Dept. of Information Technology
MPSTME NMIMS University
Mumbai, India
sneha.deshmukh@nmims.edu

Abstract
Due to the high rate of cyber threats, there is a need for better measures to prevent attacks on enterprise systems and their inherent vulnerabilities. Automating security hardening across similar but geographically dispersed Linux distributions especially the Red Hat Enterprise Linux (RHL 8 and 9), Rocky Linux (8 and 9) and Oracle Linux 9, cut down the burden of meeting the required level of standards greatly. This study is concerned with developing such a system taking into consideration the specific computers operating on these distributions. The system is also designed to mitigate the risks posed to enterprises by restricting their exposure to attack vectors and aiding constant adherence to compliance by turning on features in the system that enforces configuring the enterprise system as per the CIS benchmark. The empirical results show that it is possible and practical to implement CIS compliance in different Linux distributions and that it enhances the process of curbing risks in secured environments.

Terms- Red Hat Enterprise Linux (RHEL 8 and 9), Rocky Linux (8 and 9), Oracle Linux 9, automation, CIS benchmark, compliance, hardening.

I. INTRODUCTION

With the recent increase in the level of cyber threats faced by organizations all over the world, protecting enterprise systems is important in keeping sensitive data intact and ensuring that normal business operations are not disrupted [1]. Cyber attacks on Linux systems as a way of implementing hacked services to clients has been rising dramatically since many organizations have embraced Linux operating systems for their strength and ability to scale [1]. However, the limpid nature of Setuid implementation within the various elements of the Linux environment, and especially among its distributions such as Red Hat Enterprise Linux (RHEL 8 and 9), Rocky Linux (8 and 9), and Oracle Linux 9 makes it difficult to apply such security mechanisms on all the above systems in uniform measure.

This study tackles the challenge of automating security hardening across various distros of linux to improve its security and reduce risks and vulnerabilities [2]. The benchmarks provided by the Centre for Internet Security (CIS) are well recognised tools to achieve secure configuration across various platforms, providing detailed instructions which when adhered to, greatly reduce the risks associated with the system [3]. Nonetheless, it is very exhaustive and error-prone to apply such configurations in large-scale enterprise setting and to keep such configurations maintained over time, leading to gaps in compliance with known security standards.

This project offers an approach that allows the implementation of CIS benchmark compliance measures on multiple Linux distributions automatically, thus enhancing the security compliance measures management and minimizing the potential exposure surface [1]. Instead of relying on extensive manual intervention for system enforcement of these configurations, systems can be configured and kept secure on a continual basis in accordance with desired standards. This paper provides an overview of the structure and implementation of this automation system, evaluates its efficiency in enabling CIS compliance, and reviews how it contributes to strengthening the security of Linux-based corporate systems [5]. The results of the study highlight the need for automating security hardening in such a way that it is efficient, consistent, and most importantly, can be deployed en masse in light of modern cyber security challenges.

II. LITERATURE REVIEW

A.Security Hardening of Linux Systems

Linux systems are increasingly used in most enterprises since they offer a high level of flexibility, scalability, and resilience, which in turn makes them a target for cyber threats. To protect Linux environments, it is necessary to carry out security hardening which involves embedding systems in order to reduce the level of intrusiveness and decrease the level of attack surfaces. As Ostrowski [11] notes, the hardening of the operating systems is a paramount requirement when one wants to set up secure execution environments, enforce access controls, and limit exposure to threats by minimizing the number of services operational. Likewise, Ayyoub et al. [12] look at the vulnerabilities of Linux servers showing that the cause is often their incorrect configuration and argue that the hardening of systems is crucial to managing threats within such spaces. These studies point to the fact that basic practices, namely setting file permissions ‘chmod’, applying iptables, or making services not running any more ‘systemctl stop <service_name>’, are critical in addressing the vulnerability of a Linux operating system. Arora et al. [4] endorse the necessity of implementing defense in depth strategies and enumerate the hardening techniques and practices targeted at mitigating typical Linux vulnerabilities present in enterprises.

B.Automation of Security Hardening Processes

Given the increasing complexity of IT infrastructures, particularly distributed systems employing Linux Operating systems,

there is a need for enhanced management strategies towards security, which in this case includes the element of automation. In his article, Julisch [8] states that while indeed helpful in improving the security posture of systems, the main purpose of compliance automation is to ease the burden of workforce engaged in compliance efforts in large organizations. According to Fenz and Montesino [5], information security management is another field which is accelerated due to automation, as it allows for controlled application and minimizes risks brought by humans. In an example of what can be wrought by the use of automation, Leiritie [2] cites automating implementation of CIS benchmark compliance on Linux OS variants using Ansible and Shell scripts. The research concludes that this especially for cloud-based and physical deployment models, this automation delivers the same outcome and uses less operational time. Montesino and Fenz [7] delve deeper into the prospects of information security automation and state that the extent of such automation in the security controls can be of great advantage in enhancing the availability and reliability of such controls.

C.CIS Benchmarks Enforcements

The Center for Internet Security (CIS) benchmarks are a range of approved security standards that help improve system security on different operating systems Linux distributions such as Red Hat, Rocky Linux, and Oracle Linux. The benchmarks provide guidelines for implementing secure baseline setups in different areas, for example user management, file access restrictions and networking, all of which are critical for the security of corporate infrastructures. Jõgi [3] views CIS benchmarks as important in mitigating the dangers of inappropriate settings due to misconfiguration, and providing a starting point for secure settings of Linux devices. CIS benchmarks are ready to use tools and the step by step instructions enhance the security of the systems being installed. This is also supported by Nepal [1] who examines the security maintenance problems that differ from one distribution to the other and offers a counter solution in the use of CIS benchmarks.

D.Innovative Security Implementation

Analysis of approaches to securing containerized workloads within sytems calls for modern approaches. Mattetti et al. [14] provide a description of LiCShield framework that is aimed towards facilitating the security of containers through pre-generated security profiles for Linux Containers. With the use of AppArmor and SELinux, LiCShield is able to curb the actions of containers and protect the host resources from malicious access, hence security hardening even goes to containerized applications. Likewise, Jang [13] introduced additional protection mechanisms for Linux network kernel modules that targeted the networks with advanced kernel hardening approaches. The paper however warned against sytem outages and advocated for system stability via kernel configurations. The kernel hardening approaches, however, include protection of the network module against exploitable weaknesses, outlining how critical kernel support is in the management of networked resources on Linux.

E.Challenges in Multi-Distribution Hardening

One of the barriers to effective security hardening in different versions of Linux is maintaining a uniform security policy without losing the essences of each version. As Nepal explains, variations in the default policies of systems such as Red Hat and Oracle Linux complicates the work of enforcing obligatory policies of protection, which leads to more chances of protection gaps. The challenge is addressed in systems like SecureSys that provide the user with a flexible and fully automated toolkit for maintaining the security configuration across multiple distributions with the least variation from the norm. Eppel also explains the persistence of problems with manual hardening from division of infrastructures into manageable parts in such a way as to secure the entire area. This is because a lot of manual work entails processes which are prone to failure and such failures increase the risk of insecurity of the entire area.

F.Challenges of Hands-on Security Hardening

The manual setting of security parameters is tasking and subject to mistakes, especially in situations where uniformity and accuracy matter most. Eppel [6] elaborates on the challenges of dealing with large-scale systems, stating that it is paramount to automate mostly to maintain order and minimize the risks of misconfigurations and other oversights. Similarly, Montesino and Fenz [7] conclude that total automation is impossible because some risks will always be present. Nonetheless, it adds a level of consistency to the system that would be impossible to achieve with only human control, especially for systems at the enterprise level.

G.Further Investigation on Security Analysis

In the same vein, Bleikertz [9] adds value to existing studies devoted to cloud and security by focusing on the automated security analysis methods of infrastructure clouds, especially at this time when more and more organizations are moving to hybrid cloud deployment. The visceral study explains processes that can be automated in detecting the presence of security threats and the enforcement of security controls within cloud terrains. Similar studies demonstrate that physiological imposition can be achieved through particular methods of automation, for example, container hardening, and kernel protection (Mattetti et al. [14]; Jang [13]).

III. PROPOSED METHODOLOGY

With the increase in the number of cyber threats, it becomes imperative for organisations to establish strong security protection measures for their Linux based operating systems. The benchmarks from the Centre for Internet Security (CIS) are benchmarks that provide a systematic and well acknowledged approach in protecting system configurations. However, enforcing these standards through the various distributions of Linux such as Red Hat Enterprise Linux (RHEL), Rocky Linux and Oracle Linux, laboriously and in most cases inaccurately, is virtually impossible. This project presents a work around this shortcoming, by presenting a mechanism for automated security hardening across these

distributions based on CIS standards, but also with some aspects of user interaction and configurable levels. Each CIS-specified module is carried out with the help of a Bash script dedicated to the hardening activity and asking for user intervention whenever the need for customizing a particular installation arises [2]. In this system, three modes of operation are available: simple scan, user-defined scan of single scans, and full system scan. This being an automated approach is meant for enhancing security compliance, lessening vulnerability and improving the security of the organization [5][7].

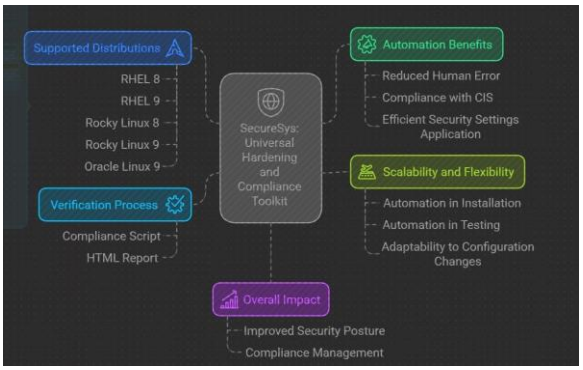

Fig 1 : Overview of SecureSys

A. User Options for Security Hardening

The system offers three user-defined hardening modes, providing flexibility to accommodate different security requirements and operational constraints:

1. Basic Mode (Core Hardening Modules): This mode configuration undertakes the most basic security measures advanced on certain critical CIS-defined modules in order to achieve a satisfactory secure level in the quickest time possible. The modules include: Filesystem Configuration: Configures the system to limit certain filesystems (for example, cramfs, hfs, squashfs) and restricts USB storage access in order to reduce the risk for the given system [1].

Package Management: Checks that the package manager repositories are configured, as well as that the GPG keys are checked and that the package update enforcement is applied.

Mandatory Access Control (MAC): Sets up SELinux to enforcing mode and attaches the relevant policies for effective security [2].

Protecting the Bootloader: A password is applied to the bootloader in order to mitigate any changes that would occur at the time of system boot by unauthorized individuals.

Hardening The Process: Imposes restrictions including address space layout randomization (ASLR), ptrace_scope, and limitations on core dumps for the purposes of securing memory [11][13].

General System Cryptographic Policies: Updated policies will use up to date cryptography standards and do away with old and weak ones.

Warning Banners: Security warnings are shown for both local and remote logins to deter unwarranted attempts of access.

GNOME Display and Manager GDM – configuration: Takes away auto mounting and allows screen locks for all physical access openings.

Service Management: Assess and switch off least required services e.g., auto file system, dynamic host configuration protocol, file transfer protocol, and telnet to enhance security [4].

Network Configuration: Sets up network parameters, such as kernel parameters ip_forwarding with the exception of unnecessary kernel modules that is dccp, tipc [10].

Firewall Configuration: It includes creating firewall policies using either firewalld or nftables depending on the default firewall service for the specific distribution.
Access Control: Other measures include hardening SSH server settings (e.g. disabling PermitRootLogin) imposing sudo policies and using strict access control measures [14][4].
User Accounts and Environment: Password policy implementation, creation of root and system accounts and user access restriction.
Logging and Auditing: Deployment of auditd to monitor systems, alongside configuring log retention policies for effective audit trail purposes.
System Maintenance: It includes interfering with the rights to access and modify certain critical system files (such as `/etc/passwd`, `/etc/shadow`) and protecting user folders.

2.Customized user defined hardening: In this mode, the user may choose to use only selected modules. This provides operational compliance flexibility. Users can pick and choose individual modules available in the CIS benchmarks. This is perfect in circumstances when for instance, only certain configurations are needed or hardening is limited by the operations in place.

3.System wide mode (Total Hardening): In this mode all the CIS benchmark modules applicable to the given distribution are executed, applying system-wide all the hardening configurations without exception. It offers the highest security coverage and guarantees complete compliance to the CIS benchmarks for a specific distribution.

B.Additional Hardening Modules

Since modern enterprise systems have more threats outside of the scope of CIS benchmarks, this system includes additional hardening modules to cover these threats [3][12]:

Web Server Hardening: Configures and secures the settings of the web server including permissions and access in order to protect against common web attacks.
Rootkit Detection: Protects system integrity through detection of irreparable modifications such as roots, rootkits, kernel level fixes, and any other invisible threats.
Database Hardening: Prioritizes safe configuration of database systems, use of encryption in connection, access control, and establishing high-level security measures.

C.Disallowing Disk Partitioning

The CIS benchmarks do, in fact, integrate configurations for the disk partitioning, but this module is intentionally left out in this methodology as the approach bears a significant risk of rendering systems inoperable in the event of erroneous changes made to disk partitions. While partitioning usually calls for consideration of the specific systems in place, omission of this aspect guarantees the security of vital data associated with the system and reduces the level of risk of operational interference.

IV. RESULTS

Following the deployment of the CIS benchmark hardening scripts to the nominated Linux operating systems, an extensive validation and reporting

mechanism is put in place to authenticate and record the outcomes [3]. This ensures that each security setting has indeed been implemented correctly with satisfactory output for subsequent examination, system tuning, or reporting to interested parties.

A.Validation Process

After every Bash script pertaining to the module is completed, a validation script that is primarily written in Python, will be invoked to analyze the usefulness of the configuration [3]. This validation script checks in detail whether each of the hardening settings has been implemented as per the CIS benchmarks or not with respect to the parameters that have been outlined [6]. This is why there is an execution of hardening scripts and an automated validation; both- in a way increase the correctness by preventing the chances of misconfigurations, or even partial configurations.

The validation script is expected to:

Confirm Configuration Integrity: Every parameter set through the Bash script will be measured against the CIS benchmarks ensuring that indeed all the set changes fit well within the required security boundaries.

Flag Issues: The script will treat a particular configuration as unsatisfactory and indicate the particular metric and the anticipated and actual outcomes if it is out of the benchmark specifications – allowing the admin to find and fix only the failed configuration.

B.Implementation of Reporting in Real Time through the Local Webpage

When the validation concludes successfully, a local web page is auto launched with pertinent information on the system's compliance level. This web page has the following features:

Only Discrepancies: In order to help focus on actionable results, the web page displays only those cases where configurations do not conform to their expected values. This straightforward view, devoid of successful configurations, allows the administrators to quickly locate and address issues requiring their concern.

The Report may be Saved as a PDF File: The page contents, as well as the reported discrepancies, may be filed or printed out by the user in a PDF format for easy circulation and proper storage. This PDF report of the discrepancies is done in such a way that technical people and non-technical people will not have a problem understanding it since the structure remains the same.

This local webpage and the accessible report present an issue in a manner which inspires looking up a solution, allows keeping tabs on how well there has been compliance to the issue in question, as well as provides communication with the concerned parties in an easier way.

C.Excel Report

Further to the site and the report in PDF form, there is also an in-depth Excel report saved as ‘CIS_<OS NAME>_<OS VERSION>_Hardening_Report.xlsx’; such a report presents the results of the configurations with more details, by modules, and can be considered as a full log of the process of security hardening. The table of contents of this Excel report predominantly contains:

Different Worksheets for Different Modules: All CIS benchmark modules have a separate sheet in the Excel workbook which allows performing analysis in a modular way. This organization allows the reviewers to concentrate, for example, only on the extrapedia's file system configuration or the firewall settings, which helps to resolve particular issues easily and quickly.

Positive and Negative Results: In this case, each worksheet always has both worked out, and failed configurations, which gives the perspective of what the system is capable of. Worked out configurations state which criteria are met; while failed ones indicate what parameters are not adequate and need to be improved.

Elaborate Details: Module sheets also contain module related info such as specific CIS requirements - target states and the actual states of controls – making it very easy to perform such analysis to check if security policies are implemented properly or facilitate any correction in future.

At the same time, the Excel report incorporates coherent, informative data on compliance level for each CIS benchmark module which increases the hardening status steps traceability and responsibility. This is also useful for the administrators and acts as a detailed documentary evidence of security compliance for appraisal and interaction with stakeholders.

V. CONCLUSIONS AND FUTURE WORK

Automated security hardening in accordance with CIS benchmarks for a number of Linux distributions was the first objective of this work [1][9]. In this regard, aside from each module Bash scripts were implemented, also a validation Python script was developed which allows systems to maintain compliance, speed up validation process and enhance reporting through a web report missing indiscrepencies only and an alleged full report in an Excel spreadsheet. Such an organized approach reduces the chances of configuration errors, allows three modes of operation, gives visibility to both technical and non-technical audiences and most importantly allows for better management of security and compliance [8].

The next step will be to increase the capabilities of the tool in such a way that it will be able to work with cloud and hybrid infrastructures, include elements of machine learning for anomalies detection and expand to continuous compliance [2]. Other activities will tackle eliminating causes for administrative errors through remediating non-effective configurations, provide dynamic interfaces on the web for users to make changes as needed, and develop other modules on cloud-security frameworks. All these would help in increasing the versatility and efficiency of the tool so that it does not become inapplicable in situations of advanced adaptive information technologies.

VI. REFERENCES

[1] Nepal, A. "Linux Server & Hardening Security." Thesis, Western Governors

University, Aug. 2013. Available: ResearchGate.

[2] Leiritie, T. "Automated Hardening of Linux Infrastructure." Bachelor's Thesis, Turku University of Applied Sciences, 2023.

[3] Jõgi, M. "Establishing, Implementing and Auditing Linux Operating System Hardening Standard for Security Compliance." Master's Thesis, University of Tartu, 2017.

[4] Arora, N., Bhosale, T., Sharma, V., and Supe, J. "Linux Hardening." Int. J. Recent Innov. Trends Comput. Commun., vol. 2, no. 5, pp. 1019–1022, May 2014.

[5] Montesino, R., and Fenz, S. "Automation Possibilities in Information Security Management." Proc. Eur. Intell. Secur. Inform. Conf., Athens, Greece, Oct. 2011, pp. 39–46, doi:10.1109/EISIC.2011.39.

[6] Eppel, N. "Security Absurdity: The Complete, Unquestionable, and Total Failure of Information Security." Security Absurdity.com. [Online]. Available: http://www.securityabsurdity.com/failure.php.

[7] Montesino, R., and Fenz, S. "Information Security Automation: How Far Can We Go?" Proc. ARES 2011 - 6th Int. Conf. Availability, Rel. Secur., Vienna, Austria, Aug. 2011, pp. 48–54, doi:10.1109/ARES.2011.48.

[8] Julisch, K. "Security Compliance: The Next Frontier in Security Research." Proc. NSPW 2008 - New Secur. Paradigms Workshop, Lake Tahoe, CA, USA, Sep. 2008, pp. 71–82, doi:10.1145/1595676.1595690.

[9] Bleikertz, S. "Automated Security Analysis of Infrastructure Clouds." Master's Thesis, Technical University of Denmark, Dept. of Informatics and Mathematical Modelling, and Norwegian University of Science and Technology, Dept. of Telematics, Jun. 2010.

[10] Anthony Pell. "Network Security with _proc_sys_net_ipv4." March 4, 2002

[11] Ostrowski, J. "OS Hardening: Making Systems More Secure." Seminar Paper, Ausgewählte Kapitel der IT-Security, Jan. 2020 (OS_Hardening).

[12] Ayyoub, B., Abu-Ein, A., Zahran, B., Nader, J., and Al-Hazaimeh, O. "Enhance Linux Security Server Misconfigurations and Hardening Methods." Inf. Sci. Lett., vol. 12, no. 3, pp. 1285–1298, Mar. 2023, doi:10.18576/isl/120319 (45d15xm4fz54i6).

[13] Jang, S. "Design of the Kernel Hardening Function in Linux Network Module." Int. J. Comput. Sci. Netw. Secur., vol. 6, no. 8B, pp. 135–138, Aug. 2006(Design of the Kernel Ha…).

[14] Mattetti, M., Shulman-Peleg, A., Allouche, Y., Corradi, A., Dolev, S., and Foschini, L. "Security Hardening of Linux Containers and Their Workloads." Proc. IEEE Conf. Commun. Netw. Secur. (CNS), Florence, Italy, Sep. 2015,

doi:10.1109/CNS.2015.7346869 (LiCShiled-submitted).
[15] https://learn.cisecurity.org/benchmarks - CIS Standards
[16] https://github.com/ansible-lockdown/RHEL9-CIS
[17] https://github.com/rediculum/RHEL8_Lockdown