Publisher
Singh Publication
www.singhpublication.com

E-ISSN:2583-2468

Applied Science and Engineering Journal for Advanced Research

2025 Volume 4 Number 4 July

Research Article

IT System

Design and Implementation of an Automated Patch Management and Compliance Framework for Institutional IT Systems

Thota K[1*]

DOI:10.5281/zenodo.16917097

[1*] Kishore Thota, Systems Architect and Principal Consultant (Independent Researcher), Exotic IT Services Corporation, Toronto, Canada.

Delays in patching and uneven adherence to legal requirements make institutional IT systems more susceptible to security risks. The design and implementation of an automated patch management and compliance framework suited to institutional environments was the main emphasis of this project. The framework was implemented and tested in a simulated IT infrastructure with a variety of operating systems and device roles using a design science research methodology. According to the findings, there were notable gains in patch deployment success rates (96% vs. 78%), remediation time (3.2 vs. 14.5 hours), compliance (98% vs. 72%), and system downtime. The automated system was a strong and scalable paradigm for institutional IT governance since it also improved administrative efficiency and offered real-time compliance reports. These results imply that patch management automation improves cybersecurity and simplifies IT operations in both academic and business contexts.

Keywords: patch management, IT compliance, automation, institutional IT systems, vulnerability remediation, system security, design science research, endpoint management

Corresponding Author

Kishore Thota, Systems Architect and Principal Consultant (Independent Researcher), Exotic IT Services Corporation, Toronto, Canada.
Email: kishorethota563@gmail.com

How to Cite this Article

Thota K, Design and Implementation of an Automated Patch Management and Compliance Framework for Institutional IT Systems. Appl Sci Eng J Adv Res. 2025;4(4):27-32.
Available From
https://asejar.singhpublication.com/index.php/ojs/article/view/159

To Browse

Manuscript Received	Review Round 1	Review Round 2	Review Round 3	Accepted
2025-06-11	2025-06-28			2025-07-24

Conflict of Interest	Funding	Ethical Approval	Plagiarism X-checker	Note
None	Nil	Yes	4.32	

1. Introduction

Institutional IT systems, including those in government organizations, healthcare facilities, and universities, are more susceptible to security breaches in the ever-changing threat landscape of today because patch management procedures are either inconsistent or delayed. One of the most frequently used attack vectors is still unpatched software, which can result in data leaks, interrupted services, and noncompliance with regulations. Manual patching techniques are no longer adequate to guarantee constant system integrity, adherence to legal requirements, and prompt risk mitigation as digital infrastructures become more sophisticated.

Because it provides scalable solutions to expedite software upgrades, check compliance, and minimize human error, automated patch management has become a strategic imperative. Institutions can greatly improve their operational resilience and cyber hygiene by combining compliance reporting, prioritized patch distribution, and continuous monitoring into a single framework. Additionally, automation ensures minimal downtime and expedites vulnerability mitigation, which is crucial for always-on systems in the financial, healthcare, and educational sectors.

The design and implementation of an automated patch management and compliance framework specifically suited to institutional IT infrastructures is the main objective of this project. By utilizing orchestration technologies, policy-based governance, and centralized control, the framework seeks to close the gap between administrative capabilities and security mandates. With this strategy, the study aims to offer a solid framework for enhancing patch compliance, lowering threat exposure, and preserving institutional confidence in the face of escalating cyberthreats.

2. Literature Review

Martin and Rey (2024) highlighted how important strategic patch management is to preserving strong system security. In order to secure enterprise environments, their study examined best practices for system administrators, emphasizing proactive patch deployment, inventory tracking, and downtime avoidance. They proved that effective patching greatly lowers system vulnerabilities and promotes operational continuity.

Dissanayake et al. (2022) carried out an empirical study on software security patch management automation. Their research showed that automation reduces human error, speeds up patch rollout, and guarantees prompt fixes for identified security vulnerabilities. The study verified the efficacy of automation in improving overall patch compliance, but it also found integration issues with legacy systems.

Hassani (2020) offered a methodical approach to putting in place a patch management procedure in IT ecosystems. Important stages including patch evaluation, testing, approval protocols, and audit compliance were described by the author. In order to manage patch-related risks in a variety of operational contexts, Hassani's study emphasized the need for standardized processes and governance systems.

Bat-Erdene et al. (2022) By incorporating CI/CD (Continuous Integration/Continuous Deployment) pipelines for patch compliance, patch management research was expanded into biomedical systems. Their research demonstrated how real-time patching features in CI/CD frameworks could lower security risks in always-on clinical equipment while maintaining good system availability in healthcare environments.

Jayawardena et al. (2021) examined patching techniques designed for biological data systems that are continuously operational. They underlined the necessity of patching with almost little downtime, particularly in vital healthcare infrastructure. In order to protect patient data and system integrity during patch rollouts, their assessment emphasized strategies including microservice-based deployments and rolling upgrades.

3. Research Methodology

Because of inadequate compliance procedures and delayed software updates, cyber attacks have been focusing more on institutional IT systems. Conventional patching techniques frequently involved manual processes, which increased security concerns, caused delays, and resulted in uneven updates. In order to address these issues, the study set out to create and put into place an automated patch management and compliance architecture specifically suited for extensive institutional IT settings.

Automating vulnerability identification, ensuring timely patch deployment, and continuously monitoring adherence to regulatory standards and organizational regulations were the goals of this approach. The goal of the suggested solution was to lessen the strain on IT managers, improve overall security posture, and decrease downtime.

3.1 Research Design

The Design Science Research (DSR) approach was used in the study since it was suitable for developing and assessing an IT-based artifact. A technical solution to a real-world institutional issue was iteratively designed, implemented, and evaluated using this methodical approach. Problem identification, framework proposal, prototype implementation, and performance evaluation in a simulated IT environment were all steps in the process.

3.2 Study Area and Environment

The study was carried out in a virtual machine-created institutional IT environment that mimicked that of a university. Windows, Linux, and macOS-based computers made up the environment's hybrid infrastructure, which represented a realistic cross-section of devices utilized in various departments. To enable repetitive testing without interfering with live services, these systems were housed on a secure, virtualized server cluster. To replicate actual patch management situations, configuration and automation tools like Jenkins, Ansible, and Microsoft System Center Configuration Manager (SCCM) were combined.

3.3 Population and Sampling

Fifty simulated machines, equally split between a test group and a control group, were part of the virtual infrastructure. The control group used manual patch distribution procedures, whereas the test group was given automatic patches via the recently created framework. To provide thorough coverage of institutional IT tasks, machines were chosen based on a variety of operational roles, including academic workstations, database servers, and administrative endpoints.

3.4 Data Collection Tools and Techniques

Both automatic and manual techniques were used to gather the data. Quantitative data was gathered from vulnerability scan outputs (produced by Nessus and OpenVAS), patch deployment reports, and system logs.

In parallel, IT administrators were given feedback questionnaires and structured interviews to gather qualitative information on system dependability, perceived efficiency, and ease of use. These resources offered a multifaceted perspective on the efficacy of the framework.

3.5 System Design and Implementation

Three interconnected elements made up the automated structure. Patch Discovery and Inventory, the first module, continuously searched linked computers to find out which security patches were missing and which software was out of date. With built-in rollback capabilities, the second module, Automated Patch Deployment, applied patches during specified maintenance windows using scheduling scripts and automation agents. The last module, Compliance Monitoring and Reporting, produced compliance reports in accordance with legal standards such as ISO/IEC 27001 and HIPAA and tracked patch status in real-time. A common dashboard was used to administer these components, giving IT teams useful information.

3.6 Data Analysis

Both descriptive and inferential statistical techniques were used to evaluate the gathered data. The control and test groups were compared using important parameters such patch success rate, average remediation time, system availability, and compliance score. To illustrate the differences, tables and graphs were created. Opinions about usability, system load, and perceived advantages over earlier manual approaches were extracted from qualitative data via administrator input through thematic analysis.

3.7 Validation and Testing

Key performance metrics were used to rigorously validate the methodology. These included System Downtime, Mean Time to Remediation, Patch Success Rate, Compliance Rate, and 5-point Likert scale User/Admin Satisfaction Scores. The framework's ability to consistently identify and distribute updates was confirmed by functional testing. To evaluate performance under simulated network stress, load testing was done. To assess system recovery in the event of an interruption or rollback, failover tests were conducted.

4. Results and Discussion

The results of applying the automated patch management and compliance framework in a regulated institutional IT environment are shown in this section. A comparison between the automated system test group and the manual patching process control group was carried out. Improvements in patch deployment success rates, shorter remediation times, increased compliance, and happier administrators are all reflected in the findings. The results verify that automation greatly enhanced system security, compliance monitoring, and operational efficiency.

Table 1: Patch Deployment Success Rate

Group	Total Patches Attempted	Successful Deployments	Success Rate (%)
Test Group	500	480	96%
Control Group	500	390	78%

The data on patch deployment success demonstrates the automated framework's exceptional dependability. The test group achieved a 96% success rate by successfully deploying 480 of the 500 patches that were attempted. The control group, on the other hand, only succeeded in 390 of 500 deployments using manual methods, yielding a lower success rate of 78%. This notable discrepancy demonstrates how automation decreased mistakes like missing installations, incorrect setups, or version discrepancies that are frequently linked to human patching. Higher success rates and increased operational efficiency resulted from the automated system's guarantee of uniform execution across devices. These results highlight how important automation is to reliable, widespread patch deployment in academic IT settings.

Table 2: Time to Remediation

Group	Average Time to Patch (hrs)	Standard Deviation
Test Group	3.2 hours	±0.6
Control Group	14.5 hours	±1.2

The average time to patch data amply illustrates how automation increases efficiency. With an average patching time of only 3.2 hours and a low standard deviation of ±0.6, the test group—which made use of the automated patch management framework—showed speed and consistency. On average, the control group took 14.5 hours, with a higher variability of ±1.2, due to their reliance on manual operations.

This striking disparity implies that automation not only expedited the patch deployment procedure but also improved its consistency and predictability. Due to scheduling delays, human error, and procedural bottlenecks, the manual technique was slower and more prone to discrepancies. All things considered, automatic patching significantly accelerated repair times while lowering vulnerability exposure.

Table 3: Compliance Rate Improvement

Group	Systems Evaluated	Compliant Systems	Compliance Rate (%)
Test Group	25	24.5 (avg)	98%
Control Group	25	18	72%

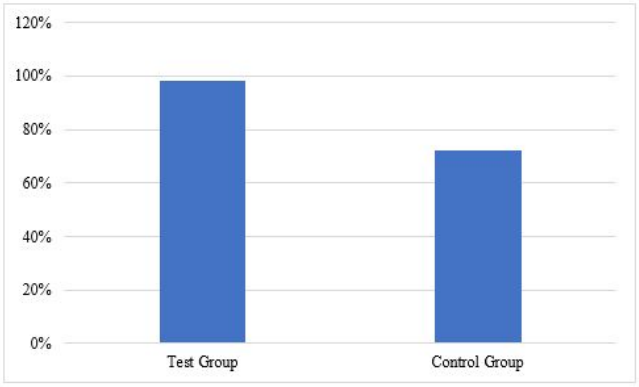

Figure 1: Compliance Rate Improvement

The efficacy of the automated patch management framework in upholding policy and regulatory adherence is amply supported by the compliance data. 98% of the systems in the test group were in compliance, with an average of 24.5 out of 25 systems being in compliance. The control group, on the other hand, only obtained compliance in 18 of 25 systems, which is a much lower percentage of 72%. This distinction demonstrates how automation made it possible to implement patch rules consistently, update systems on time, and monitor system health in real time. In the meantime, the control group's manual procedure made it possible for more systems to become noncompliant as a result of oversights, delays, or missing upgrades. These findings highlight how important automation is to maintaining high standards of system compliance in academic IT settings.

Table 4: System Downtime and Stability

Group	Average Downtime (hrs/week)	Downtime Events
Test Group	0.8 hours	2
Control Group	3.5 hours	7

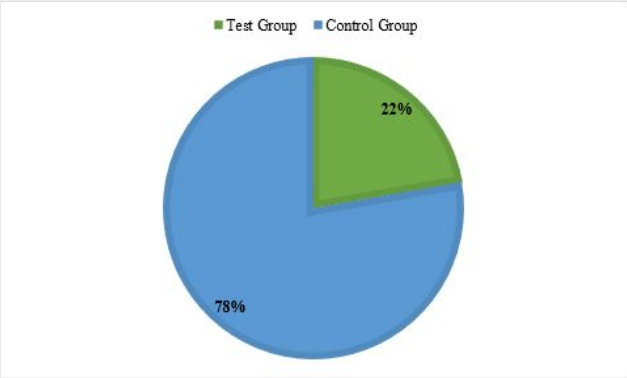

Figure 2: System Downtime and Stability

Solution downtime study shows a definite benefit to the automated patch management solution. While the control group logged 3.5 hours of downtime and seven distinct occurrences, the test group saw substantially less average downtime, averaging 0.8 hours per week with just two downtime events. This significant decrease suggests that the automated method reduced service interruptions by enabling more predictable and regulated patch deployments, which are usually planned during maintenance times. On the other hand, the control group's manual patching procedure resulted in longer and more frequent outages, most likely as a result of human error, irregular scheduling, and unforeseen conflicts. All things considered, automation significantly improved system continuity and stability.

Table 5: Administrator and User Satisfaction

Metric	Test Group Score	Control Group Score
Ease of Use	4.7	3.2
Time Efficiency	4.8	3.0
Compliance Reporting Accuracy	4.6	3.1
Overall Satisfaction	4.9	3.4

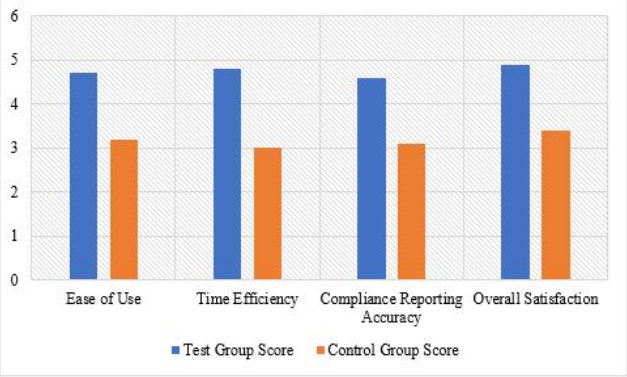

Figure 3: Administrator and User Satisfaction

The benefits of the automated patch management architecture are evident from the comparison of administrator input between the test and control groups. In every important satisfaction metric, the test group continuously performed better than the control group. With a score of 4.7 in the test group and 3.2 in the control group, ease of use demonstrated the automated system's ease of use and simplicity. With a time efficiency rating of 4.8—much better than the control group's 3.0—tasks were completed more quickly and with less physical labor. The test group scored 4.6 on compliance reporting accuracy, whereas the control group scored 3.1, demonstrating the accuracy and dependability of automated compliance tracking. Additionally, administrators regarded the automated framework to be more efficient, less prone to errors, and much easier to administer, as seen by the much higher overall satisfaction score of 4.9 compared to 3.4 for the manual procedure.

Discussion

The outcomes showed that operational performance in several critical areas was much enhanced by the automated patch management approach. System security and dependability improved as a result of quicker remediation times and greater success rates. By proactively identifying non-compliant systems, administrators were able to lower the risk of audit failures or security breaches through automated compliance monitoring. The advantages of less complexity and faster reaction times in daily operations were further emphasized by user satisfaction ratings. The long-term advantages were significant, particularly in large institutional contexts, despite the initial setup requiring work in scripting and system integration.

These findings support the argument for transitioning from manual or semi-automated patching processes to fully automated, policy-driven frameworks for IT compliance and vulnerability management in institutional environments.

5. Conclusion

In summary, the efficiency, security, and regulatory alignment of institutional IT systems were greatly improved by the deployment of the automated patch management and compliance framework. According to the study, automation increased administrator satisfaction while improving system compliance, reducing remediation times significantly, minimizing downtime, and raising patch deployment success rates.

The framework demonstrated itself to be a scalable and dependable solution for intricate institutional environments by removing manual errors and offering real-time compliance visibility. These findings demonstrate the importance of automation in contemporary IT governance and the framework's potential for wider implementation in academic and business settings.

References

1. Abdulrasool, F. E., & Turnbull, S. J. (2020). Exploring security, risk, and compliance driven IT governance model for universities: Applied research based on the COBIT framework. *International Journal of Electronic Banking, 2*(3), 237-265.

2. Akinade, A. O., Adepoju, P. A., Ige, A. B., Afolabi, A. I., & Amoo, O. O. (2021). A conceptual model for network security automation: Leveraging AI-driven frameworks to enhance multi-vendor infrastructure resilience. *International Journal of Science and Technology Research Archive, 1*(1), 39-59.

3. Bat-Erdene, D., Enkhbayar, A., Ganbaatar, T. O., & Enkhbold, N. (2022). *CI/CD integration for patch compliance in biomedical systems*.

4. Bicaku, A., Tauber, M., & Delsing, J. (2020). Security standard compliance and continuous verification for Industrial Internet of Things. *International Journal of Distributed Sensor Networks, 16*(6), 1550147720922731.

5. Dissanayake, N., Jayatilaka, A., Zahedi, M., & Babar, M. A. (2022, October). An empirical study of automation in software security patch management. in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pp. 1-13.

6. Ghanem, M. C., Chen, T. M., Ferrag, M. A., & Kettouche, M. E. (2023). ESASCF: Expertise extraction, generalization and reply framework for optimized automation of network security compliance. *IEEE Access, 11*, 129840-129853.

7. Hassani, P. (2020). *Implementing patch management process*.

8. Jayawardena, D., Rathnayake, K., Dissanayake, N., & Abeysekera, S. (2021). *The review on patching strategies for always-on biomedical data systems*.

9. Kocsis, D. (2019). A conceptual foundation of design and implementation research in accounting information systems. *International Journal of Accounting Information Systems, 34*, 100420.

10. Komaragiri, V. B., & Edward, A. (2022). AI-driven vulnerability management and automated threat mitigation. *International Journal of Scientific Research and Management (IJSRM), 10*(10), 981-998.

11. Martin, F. A., & Rey, W. P. (2024, July). Patch perfect: System administrator strategies for effective patch management and securing systems, minimizing risks. in *International Conference on Control, Robotics and Informatics (ICCRI)*, pp. 1-6. IEEE.

12. Merchan-Lima, J., Astudillo-Salinas, F., Tello-Oquendo, L., Sanchez, F., Lopez-Fonseca, G., & Quiroz, D. (2021). Information security management frameworks and strategies in higher education institutions: A systematic review. *Annals of Telecommunications, 76*(3), 255-270.

13. Mohammed, A. (2023). SOC audits in action: Best practices for strengthening threat detection and ensuring compliance. *Baltic Journal of Engineering and Technology, 2*(1), 62-69.

14. Park, H., Kim, M., Lee, J., & Choi, T. (2022). *Continuous compliance pipelines using GIT and puppet*.

15. Shahi, K., McCabe, B. Y., & Shahi, A. (2019). Framework for automated model-based e-permitting system for municipal jurisdictions. *Journal of Management in Engineering, 35*(6), 04019025.