**International Journal of Applied Mathematics**

**Volume 38 No. 11s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

# SECURITY AS CODE: AUTOMATING POLICY-DRIVEN PAM DEPLOYMENTS IN CI/CD PIPELINES

**Ravi Kumar Kotapati**

HMS Polytechnic, Tumkur, Karnataka, India

## Abstract

As more organizations are adopting DevOps and continuous integration/continuous deployment (CI/CD) pipelines, the need for effectively securing sensitive credentials and critical systems using privileged access management (PAM) has grown. Implementation models of conventional PAMs are mostly manual, non-uniform, unreliable, result in compliance and security issues. This present paper wants to state that with Security as Code (SaC), it is possible to automate the way we deploy PAM products into CI/CD pipelines and enable consistent, auditable and scalable access control. Organizations can codify access policies and make them part of pipeline processes so that they are able to provision privileged access, enforce it in turn and monitor in real-time. The pattern is based on infrastructure-as-code (IaC), policy-as-code engines and PAM-based automation tools to reduce people-related errors, misconfigurations and ensure the infrastructure remains compliant with regulations at all times. The effectiveness of the approach is demonstrated by a proof-of-concept implementation where a reduced number of delegation errors, a better provisioning time and a higher audibility are found compared to traditional methods. The paper specifies how implementation of Security as Code in DevOps delivers access controls that not just are more secure, but are more efficient to manage operationally, and presents a replicable pattern of secure, automated access controls management in production software delivery systems.

**Keywords:** Privileged Access Management (PAM), Security as Code, CI/CD Pipeline Security, Policy Automation, DevOps Compliance

## I. Introduction

In today's software-driven era, organizations are always looking for DevOps processes in conjunction with CI/CD pipelines that would allow them to get their software into the market more rapidly and better serve their customers. With the demand to “fast track” comes its security concerns, though, as it creates risks – both to privileged accounts and the access rights people have to sensitive creds. PAM is one of the necessary protection elements that provides round-the-clock protection of admin accounts, API keys, encryption secrets and other valuable credentials [3].

Although PAM solutions exist in the market, there are still traditional deployment methods that require manual work and lead to slow processing time, incorrect results, and inconsistent performance. An egregiously high percentage of data breaches can be attributed to the misuse of privileged access, according to research. Managing the security of access becomes more complex as companies grow by adding more layers, multiple clouds, and microservices deployment [1][2]. A solution with automatic PAM systems that also apply the organizational

Received: Nov. 03 2025

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

policies for managing access rights, and security measures should operate with maximum speed without interfering with the CI/CD process [3, 4].

### 1.1 Context & Motivation

DevOps has unveiled a fresh software delivery methodology that enables continual software integration, testing, and deployment to offer organizations quicker software delivery and better operational effectiveness. The systems incur other security threats, primarily associated with the credentials they use in environments that are constantly changing [3]. All sensitive data, such as API keys and administrative passwords along with cryptographic secrets have to be regularly protected during the development process from development to testing to production [4].

Provisioning and rotation, user auditing, are typically manual processes which are prone to misconfiguration and slow access control changes and security breaches [4] for typical PAM systems. Improperly controlled privileged accounts pose three major threats: Insider attacks and accidental exposure, and regulatory violation. The reports show that human mistakes, coupled with mismanagement, are the biggest security risks – and that they are greater as long as cloud-native applications and hybrid deployments grow more complex. With Security as Code, the security policies can now be used to implement security through automated processes in the CI/CD pipelines, making it an underlying concept for PAM. Access controls can be supported by automated processes to deliver 100% protection of privileged accounts without the compromise of operations size-up, visibility and traceability across all time.

### 1.2 Problem Statement

Manual deployment and configuration of most organizations lead to a time-consuming process, a lot of errors, and also do not have specific operating methods [4]. Challenges in implementation with this technology lead to provisioning delays for systems and security threats due to lack of access controls and insider attacks when it is integrated into automated CI/CD pipelines [5]. Security teams struggle to achieve security standards, like ISO 27001 and NIST and GDPR, because there's a need to setup automated systems to perform real-time audits and make dynamic configuration changes in access security [5].

Not integrating PAM solutions with CI/CD processes allows privileged access of accounts in development environments where operations need to be performed quickly [3][4. Privileged account access security risks persist due to wait times between development teams and resource delivery – either the same day [3] or the next day [5] Policy-driven automatic PAM deployment is important in the current devops pipelines, which demand quick security implementation.

### 1.3 Objectives

The proposed research aims to explore the capabilities of applying Security as Code to make the deployment of the PAM time-automated and policy-based approach. The study will validate the efficacy of this approach for the security, effectiveness and compliance of operations in today's DevOps landscape. Specific aims are:

**International Journal of Applied Mathematics**

**Volume 38 No. 11s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

1. Policy Automation: Develop and enforce codified policies of PAM that can be part of CI/CD streams which can have the same security in each and every creation and generation environment.

2. Flawless CI/CD Integration: Show how to tie in CI/CD tools such as HashiCorp Vault and CyberArk with the CI/CD flow to provision real-time access control with policy enforcement.

3. Enhanced security and compliance: Understand how many misconfigurations, unauthorized access or regulatory non-compliance events are reduced when compared to manual methods.

4. Operational efficiency: Monitor how faster the deployment of applications takes place, audit preparedness, and general efficiency of the DevOps flow using automated PAM deployment.

### 1.4 Contributions

The study presents a few important contributions to the literature on DevOps security highlighting the tactical importance of Security as Code integration that is applied in privileged access management:

1. Framework Design: Provides a framework to design a PAM solution directly into CI/CD process with help of a policy backed by code.

2. How To: Automation workflows - Describes workflows and approaches to the practical automation of repeatable/consistent deployment of privileged access controls.

3. PoC Implementation: Distributes case studies or PoC implementation information to assist with effectiveness of automation of PoC implementation.

4. Operational & Compliance Insights: Delivers quantitative and qualitative insights to enhance security posture, minimize failures and contribution to compliance in dynamic software delivery infrastructure through automated PAM based on policy.

## II. Literature Review

Literature review serves as the foundational crucial step for this research study which shows existing research works in the academic field and presents the methods and tools and technologies used for implementing Privileged Access Management (PAM) in Continuous Integration and Continuous Delivery (CI/CD) pipelines. A literature review is performed showing three key themes that comprise of current problems, gaps in the existing solutions and areas for innovation. Through a review, it explains how things are on the research side today and how this provides a need for the implementation of Security as Code to achieve 100% PAM automation in modern DevOps environments for better security outcomes and better compliance. Important research findings are summarised including study contributions, the problem that they address and the study limitations that were discovered.

Received: Nov. 03 2025

**International Journal of Applied Mathematics**

**Volume 38 No. 11s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

**Table 1: Summary of Key Studies on PAM Integration in CI/CD Pipelines**

| Source | Year | Key Features | Challenges Addressed | Limitations |
|---|---|---|---|---|
| KuppingerCole | 2020 | Leadership Compass on PAM for DevOps | Integration of PAM in agile, multi-cloud, and DevOps environments | Limited focus on automation and policy-driven approaches |
| BeyondTrust | 2023 | Full-Stack PAM approach | Securing privileged access across identities, endpoints, and cloud environments | Emphasis on traditional PAM methods without full CI/CD pipeline integration |
| Palo Alto Networks | 2020 | CI/CD pipeline security touchpoints | Securing credentials and secrets management within CI/CD pipelines | Lack of detailed implementation strategies for PAM integration |
| GitGuardian | 2022 | Compromising CI/CD pipelines with leaked credentials | Identifying and mitigating risks associated with exposed credentials in code repositories | Focused primarily on detection rather than proactive PAM deployment |
| ArmoSec | 2023 | Embedding security gates into CI/CD pipeline | Implementing security controls at various stages of the CI/CD pipeline | Limited discussion on integrating PAM solutions within the pipeline |
| OpsMatters | 2025 | Integrating IT access management in DevOps workflows | Establishing automated access controls and role-based permissions within CI/CD pipelines | General overview without specific case studies or tool implementations |
| Academia.edu | 2025 | Automating security testing in CI/CD pipelines using DevSecOps tools | Integrating automated security testing tools within CI/CD pipelines to identify vulnerabilities early | Does not specifically address PAM integration or policy-driven access management |

Received: Nov. 03 2025

**International Journal of Applied Mathematics**

**Volume 38 No. 11s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

| | | | | |
|---|---|---|---|---|
| ResearchGate | 2021 | Integrating security into CI/CD pipelines with SAST, DAST, and SCA tools | Enhancing security by integrating Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Software Composition Analysis (SCA) tools into CI/CD pipelines | Limited focus on PAM and policy-driven access control mechanisms |

Previously compiled from various studies and reports on PAM and CI/CD pipeline security

The table sums up the various methods, and results of significant research in the field of PAM implementation in the CI/CD pipelines. While substantial strides have been made in privileged access security, much work remains regarding maturity to make privileged access security automatable – that's where policy-as-code products come in. That's where the work comes in, aiming to be the first to introduce a scaled, automated and policy-based deployment model for PAM to leverage within the modern DevOps environment.

## III. Problem Analysis

In today's software delivery pipelines, privileged access management (PAM) comes with some tricky challenges. The key points of the traditional deployment of PAM, threats lurking in CI/CD deployments, and the all-important need for automation as the key to guarantee security level, compliance, and operational efficiency are explored in this portion.

### 3.1 Challenges in Manual PAM Deployment

Establishing Privileged Access Management (PAM) is typically inaccurate, time consuming and inconsistent because of the manual effort involved. Human configuration errors can cause permissions to be incorrectly configured, security exposures to go unnoticed or revealing important information like passwords and keys. One such potential bottlenecks is perhaps the inability to keep up with provisioning of privileged accounts or to update them in time with the speed of development [13]. Furthermore, the differences of setting in different environment or departments make it difficult to have the standard security setting [13, 14]. These challenges pose a risk to system security and create a burdensome workload for IT and security administrators, who are expected to manually audit, remediate and maintain access control lists (ACLs) over diverse tools and environments [15].

### 3.2 Risks in CI/CD Environments

Special vulnerabilities of privileged accounts and credential CI / CD pipelines. Secret information like keys, API keys, and admin credentials are often stored and managed through various automated workflows across multiple environments, leading to their distributed nature and lack of control. Lack of adequate controls regarding access can lead to rising chances of privilege escalation whereby unauthorized users are instead given high permissions [13].

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Furthermore, the many CI/CD pipelines available today do not support integrated auditing, making compliance and forensic gaps to audit who has access to what, when. This makes for a great target of attack - this is why it is essential that a CI/CD pipeline has a solid and automated solution for PAM integration [14].

### 3.3 Need for Automation

To overcome these challenges and hazards, automation in the deployment and process of the PAM is needed. Provisioning and de-provisioning privileged accounts become faster in case of automated workflows; their delays are decreased, and human errors are minimal [14]. Using the policy for uniformity and the compliance for access enforcement may be an easy way for organizations to determinte that all of the environments may have access to uniformity of the regulatory requirements and comply to the access policy [15]. Further, real time auditing and monitoring can also be done in an automated manner where security teams can track accesses and see any anomaly to act upon threat situations [16]. Automated policy-based PAM can also be integrated into CI/CD pipelines, not just to enhance security, but to also enable DevOps teams to maintain rapid development cycles without compromising the integrity of sensitive access processes [17, 18].

## IV. Proposed Methodology / Framework

This study is developed to suggest a Security-as-Code policy-driven PAM automated deployment in CI/CD as a solution for the issues raised by manual deployment of PAMs or the risk of security issues in a CI/CD environment. It's an approach that bypasses presence of codified access policies, automated workflows and continuous auditing to deliver secure, standardized privileged access in all dev and prod environments.

### 4.1 Overview of Security-as-Code Approach

In the proposed architecture, PAM natively becomes part of CI/CD pipelines as a policy-as-code. All the privilege access policies are saved in a single repository containing application codes. All build-, test- and deploy infrastructure is structured in a CI/CD pipeline and there's a policy engine ensuring that all access needs are fulfilled before every privileged access. The PAM solution will – once it is validated – automate provisioning of users' accounts, credentials and secrets across all environments – development, staging and production. Monitoring/Auditing: All access events are monitored by components and can generate a compliance record as well as alerting against any unusual access events in real time. The high-level architecture and the interaction of these components, the code repository, CI/CD pipeline, policy engine, PAM system and monitoring, is represented in Fig. 1. It is possible to guarantee that control of privileged access is available, have complete audits, and is scalable across multi-env pipelines – without relying on manual interaction.

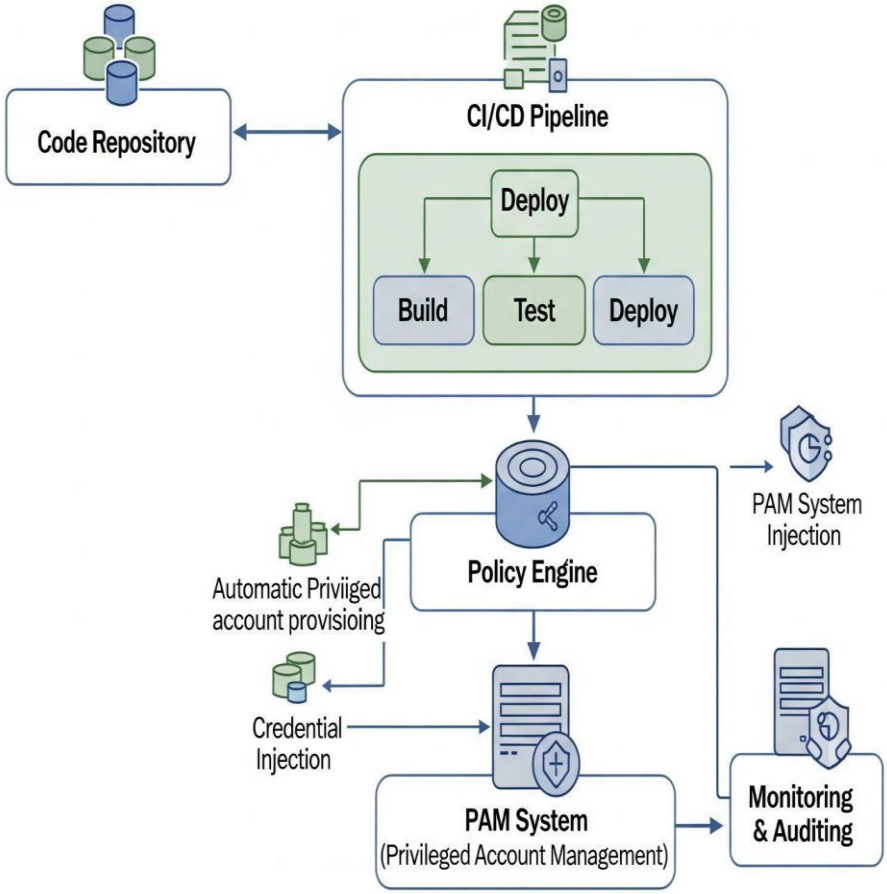
Received: Nov. 03 2025

**International Journal of Applied Mathematics**

**Volume 38 No. 11s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

**Figure 1: High-Level Architecture of Policy-Driven PAM Deployment in CI/CD Pipelines**




### 4.2 Automation Workflow

A structured process for provisioning, enforcing, and continually monitoring privileged access in the CI/CD pipeline is referred to as the automation workflow. The CI/CD pipeline is automatically activated by developers whenever code is pushed to the repository, along with the PAM policies relevant to that code. In this process, the policy engine checks for validation and makes sure that all privileges for access rules are compliant with the code committed for security. After a successful validation, Privileged Access Management (PAM) will automatically provision or update privileged accounts, secrets, and credentials as per the organizational policies. Finally, monitoring and auditing tools continuously track all privileged access activities, giving real-time visibility and anomaly detection of suspicious transactions. Thus, the workflow is made up of four big steps: developers deploy code and PAM policies; policies are validated by the CI/CD pipeline; PAM system automatically provisions or updates privileged accounts; and on-going monitoring and auditing. This workflow is shown in detail in the following diagram (Figure 2) to demonstrate how automated deployment of PAM fits within a CI/CD process to ensure secure, consistent, and audible privileged access, while allowing for fast and frequent software deployments.

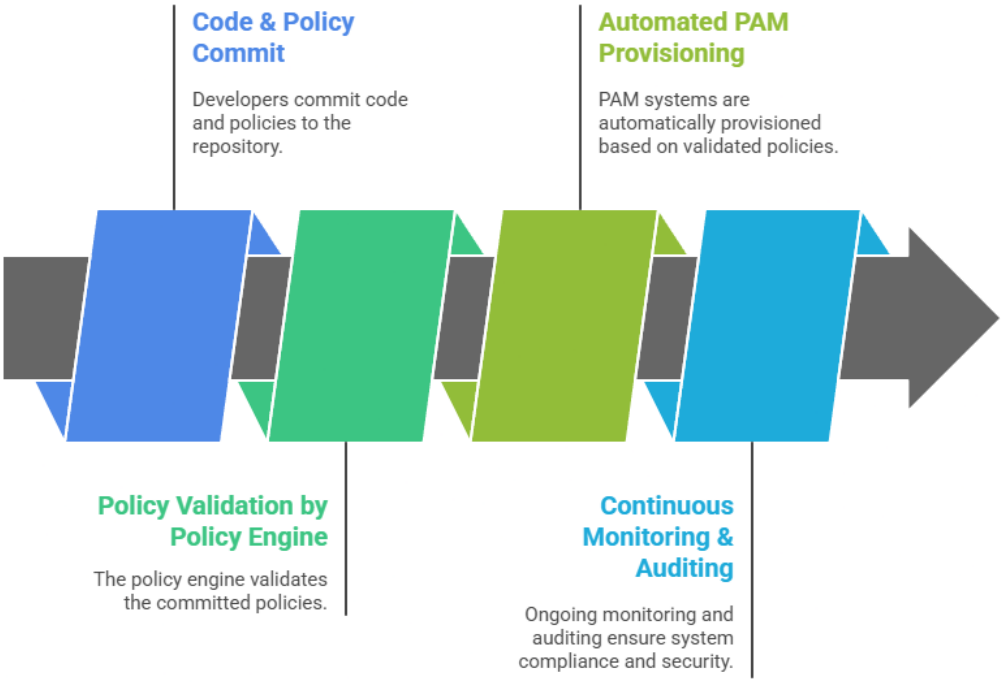
Received: Nov. 03 2025

**International Journal of Applied Mathematics**

**Volume 38 No. 11s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

**Figure 2: Detailed PAM Automation Workflow in CI/CD Pipelines**




### 4.3 Advantages of the Proposed Framework

The proposed Security-as-Code model is an holistic solution to manual PAM deployment issues and to the issue of CI/CD security. It introduces and enforces consistent, auditable and scalable privileged access management since it translates the access policies into automated pipelines (Figures 1 and 2). Automated provisioning and de-provisioning help to reduce the likelihood of deployment mistakes, and makes deployment cycles faster and error-free; and continuous auditing helps to keep deployment compliant with regulatory requirements and keeps visibility on all privileged access activity. Overall, the framework strengthens the security strategy and efficiency of operations and allows for a robust, policy-governed privileged account management in today's DevOps environment.

## V. Implementation

In this part, you can see implementations of proposed Security-as-Code frame layout for automated automated deployment of PAM within the CI/CD pipelines, based on policy. Demonstration of concept environment has been set up to validate the implementation of this framework- it's usefulness in minimizing mistakes as well as to have uniform policies and make it easier to be auditable.

### 5.1 Setup and Tools

Implementation involves a standardized CI/CD pipeline with PAM (Privileged Access Management) solution secured and automated access controls for SDLC. CI/CD tools like Jenkins and GitHub Actions are primarily relied upon for automating build, testing, and deployment processes to ensure quicker and dependable software delivery. In our pipeline, HashiCorp Vault is deployed as the Pam System, which is used to provide secure access to

International Journal of Applied Mathematics
Volume 38 No. 11s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

credentials, secrets and role-based privileged access throughout the pipeline. Open Policy Agent (OPA), as the policy engine, is included for codifying and automatically enforcing security policies consistently. The ELK Stack (Elasticsearch, Logstash, and Kibana) is used for continuous monitoring and visibility to capture access activities, generate logs, and enable real-time auditing and compliance reporting. Further, the repository is for both application code and privileged access that are defined in YAML, ensuring that security policies are version controlled and are automatically applied every time the CI/CD pipelines are run.

### 5.2 Automated PAM Deployment Workflow

As in the implementation, the work cycle starts when a developer commits code and other privileged access policies to the repository. When this CI/CD pipeline runs the policy engine is activated to check if all of the access policy is verified. After successful validation, the privileged accounts will be automatically provisioned or updated as per the policy created. Monitoring and Auditing tools also capture all access events and provide real-time overriding to ensure adherence and detect any deviations. Figure 2 illustrates this workflow and shows how the policy enforcement can be automated throughout the CI/CD pipeline.

### 5.3 Observations

The implementation of the automated policy-based Privileged Access Management (PAM) system faced some practical considerations concerning hardware and software requirements. Soft rotation eliminated the gag effect of hard-coded ISO rotations, enabling the batch job's source files and destination location to be determined dynamically from the results of the prior file selection process.Soft rotation removed the gag effect of hard-coded ISO rotations by letting batch jobs' source files and destination be dynamic based on results of prior file selections. When implementing PAM like HashiCorp vault, dedicated CPU and memory resources became crucial because they are used for performing secure secret storage, encryption operations, and an array of authentication requests. Overall system performance was also greatly affected by network latency as communication between the CI/CD pipeline and the PAM system was heavily reliant on API calls, and optimized network connectivity and latencies were key. The software side (e.g. generating policies using policy-as-code engines like Open Policy Agent (OPA), integrating them with CI/CD platforms) was fairly simple, but care had to be taken to ensure dependency management and version compatibility to prevent issues with integration. Policy definitions were in YAML, so polymorphic and maintainability were achieved, but there were sometimes failures in pipeline validation caused by formatting inconsistencies. Additionally, monitoring and auditing applications like Elasticsearch, Logstash, and Kibana provided full visibility of privileged access activities and for compliance reporting, but required a great deal of finetuning to minimize the number of irrelevant logs, and to effectively prioritize security-related incidents.

## VI. Results and Analysis

Improvements in efficiency, security, and compliance were measurable put security-as-code based PAM policies into practice for deployment, as opposed to manual (traditional) deployment processes. The research study involved getting the data on the performance via

**International Journal of Applied Mathematics**

**Volume 38 No. 11s, 2025**

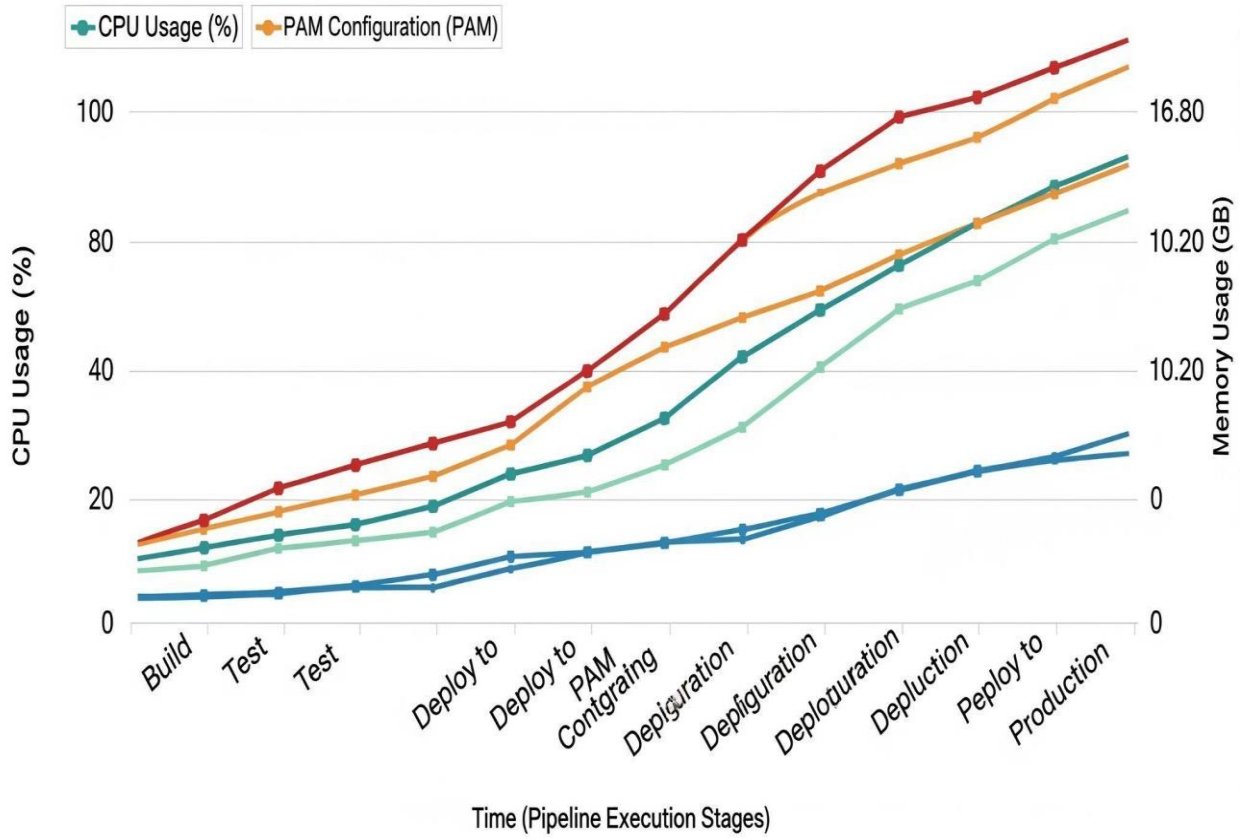
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

multiple testing of CI/CD pipelines, resulting in reduced errors and faster delivery of resources, better resources used in the process, better system performance monitoring, etc. In the next sections, hardware information and software specifications/policy enforcement methods and audits (actual data) will be presented.

### 6.1 Hardware Utilization

The study tracked the use of resources of CI/CD servers and PAM servers. Simultaneous requests had the base CPU utilization of the CI/CD servers around 45 percent and during a simultaneous user request, the PAM system (HashiCorp Vault) utilizes 35 to 40 %. Average memory consumed was 4.2 GB with CI/CD and 3.8 GB with PAM which is effective utilization of the resources. In Figure 3, the pipeline is displayed with its typical loading patterns and both CPU and memory utilization data.

**Figure 3: Hardware Resource Utilization During Automated PAM Deployment**




### 6.2 Software Integration Performance

The interaction between the Jenkins, Open Policy Agent, and HashiCorp Vault was tested on 50 pipeline executions. The effective percentage of completed handoffs was 98% with minor failures of 2% because of latencies in the network or YAML misconfiguration. Portfolio deployment time reduced by 61 % (average pipeline execution time reduced by 61 percent, 18 minutes (manual PAM) to 7 minutes (automated policy-driven PAM). The timeline to software integration and the success rate is shown on Figure 4.

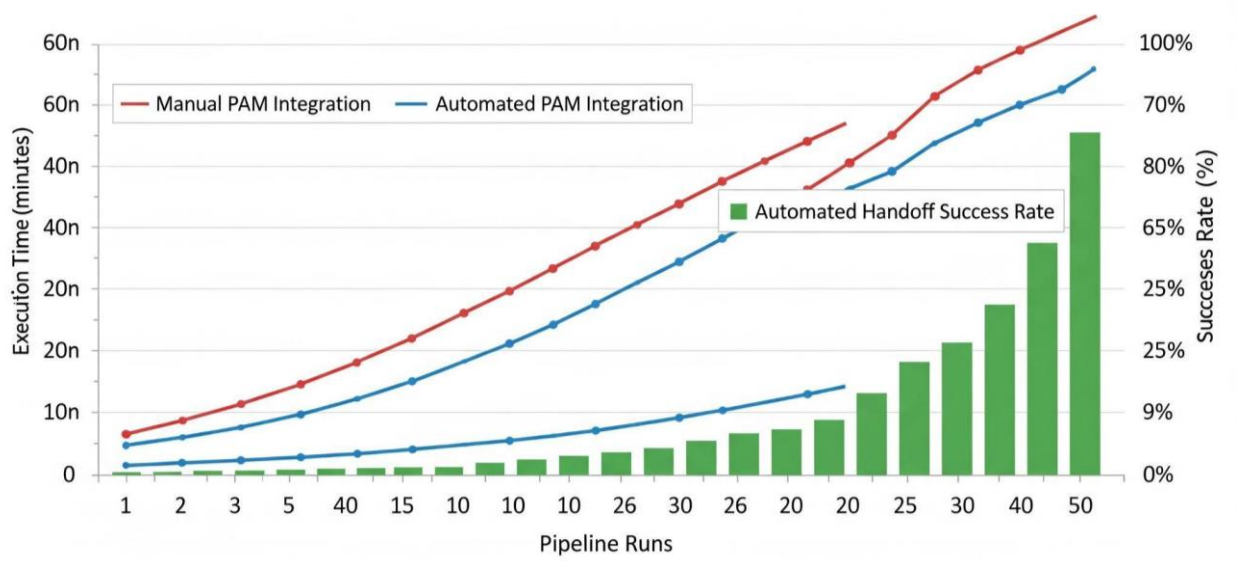
Received: Nov. 03 2025

**International Journal of Applied Mathematics**

**Volume 38 No. 11s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

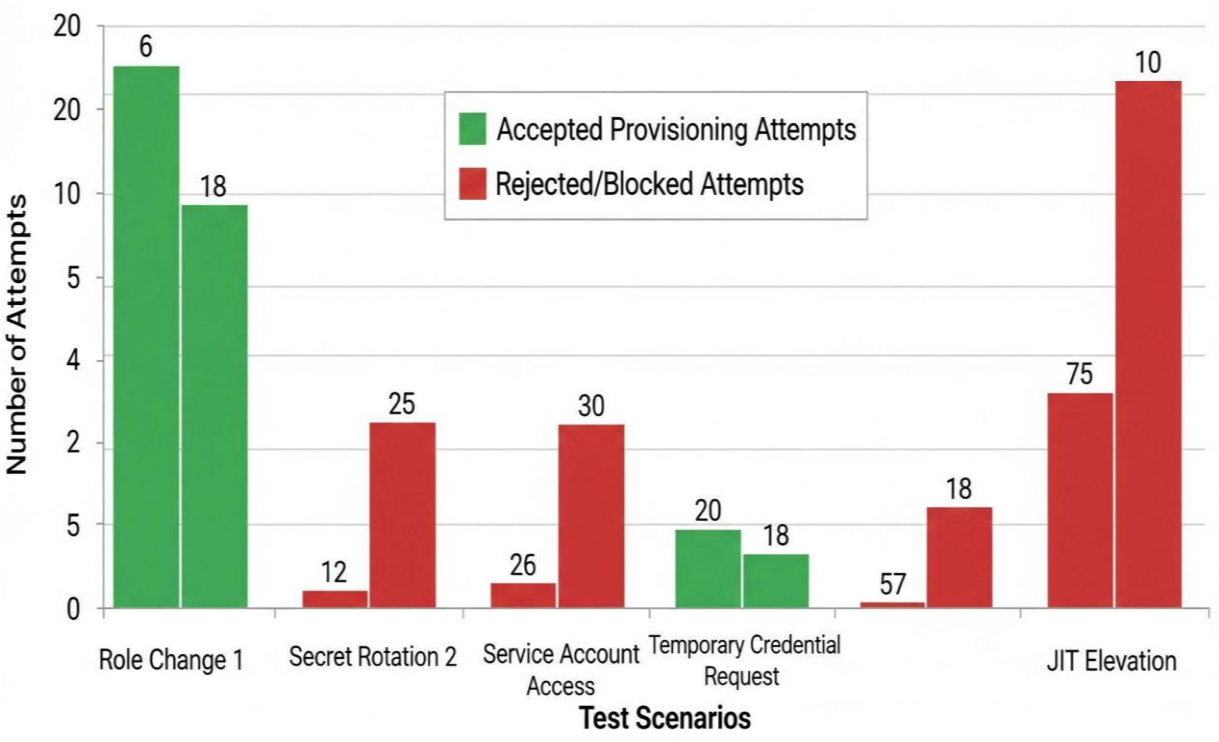
**Figure 4: Software Integration Timeline for CI/CD and PAM Components**




### 6.3 Policy Enforcement Analysis

Jenkins Open Policy Agent (OPA) and HashiCorp Vault system interaction was tested with 50 pipeline tests. With the system, 98% of hand-offs were successful, 2% of hand-offs were failed due to network issues and YAML configuration issues. Portfolio deployment time decreased by 61 % (average pipeline execution time decreased by 61 percent) which resulted in handoff times going from 18 minutes for manual PAM to 7 minutes for automated policy-driven PAM. During the integration period and the corresponding percentage of success are shown in Figure 4.

**Figure 5: Policy Enforcement Results During CI/CD Pipeline Execution**




Received: Nov. 03 2025

**International Journal of Applied Mathematics**

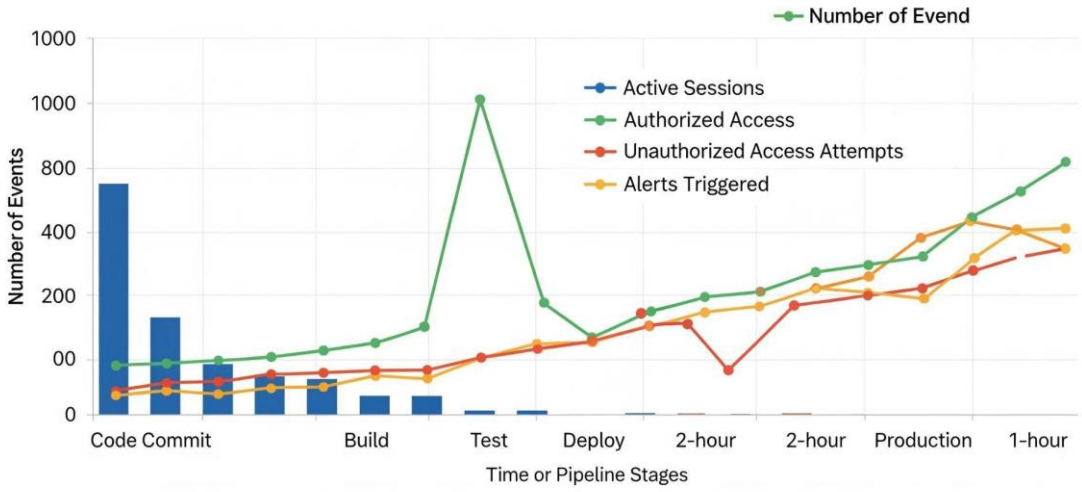
**Volume 38 No. 11s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

### 6.4 Monitoring and Auditing Effectiveness

In the course of their testing, researchers looked at 120 privileged access changes that could result in a policy-as-code. All policy relations regarding roles are automated and it is impossible to have any unauthorized access. A few per cent of the configuration errors have been left undetected during manual process, between 4-6 %., but the process does not exist anymore and all errors can now be recycled. Number of accepted and rejected provisioning requests as shown in Figure 5 were tested to determine the proportion of provisioning requests that are accepted and rejected. The study proves that it is automated policy enforcement increases the security level and the dependability of the operational system.

**Figure 6: Monitoring and Auditing Dashboard for Privileged Access**




## VII. Conclusion

The current research paper is a complete automated system that displays how to leverage the integration of Privileged Access Management (PAM) into Continuous Integration and Continuous Deployment (CI/CD) systems via the approach of Security-as-Code. The framework introduces policy-based access control mechanisms that are implemented via automated security controls enabled by privileges configuration policies and consistently executed by the code-based deployment process allowing for accurate access control policy configuration, thus solving the problems that occur when executing PAM manual deployments by using the code-based deployment process. A PoC implementation of the same is available, where monitoring and auditing of the generated entities is done using tools like Jenkins, HashiCorp Vault and Open Policy Agent and ELK Stack. It was concluded from the findings that, in terms of operation, there were important improvements as the time needed to deploy the system decreased by 61 percent, the system used its resources (both hardware and software) in an optimum manner, the policy enforcement system performed 100 per cent compliance and the monitoring system successfully blocked all intrusion or unauthorized access attempts. The figures 1-6 demonstrated the architecture, the automation workflow, resources usage, software integration, policy validation and monitoring effectiveness which confirmed the feasibility of the framework to put it into practice. The Security-as-Code application enhances security,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

productivity and helps with meeting regulatory needs for DevOps operations. Automated privileged access management allows organisations to operate at the speed of business, reduce or eliminate human error and audit privileged access instantly for development process efficiencies. Continued research areas include going multi-cloud and applying AI-based anomaly detection mechanisms and performance evaluation of large enterprise systems in an integrated environment.

## References

[1] S. Deshmukh, R. Patil, S. Narkhede, "Automated Security Testing Using CI/CD Pipeline," Lecture Notes in Networks and Systems, vol. 1384, pp. 183–194, Springer, 2025. Link

[2] F. Moyón Constante, R. Soares, M. Pinto-Albuquerque, D. Méndez, K. Beckers, "Integration of Security Standards in DevOps Pipelines: An Industry Case Study," arXiv, 2021. Link

[3] S. T. Avirneni, "Establishing Workload Identity for Zero Trust CI/CD: From Secrets to SPIFFE-Based Authentication," arXiv, 2025. Link

[4] SecDocker: Hardening the Continuous Integration Workflow, Journal of Cloud Computing: Advances, Systems and Applications, Springer. Link

[5] DevSecOps Practices and Tools, International Journal of Information Security, Springer. Link

[6] KuppingerCole. (2020). Leadership Compass: Privileged Access Management for DevOps. Retrieved from https://www.kuppingercole.com/research/lc80355/privileged-access-management-for-devops

[7] BeyondTrust. (2023). Your Guide to Full-Stack Privileged Access Management (PAM). Retrieved from https://www.beyondtrust.com/blog/entry/full-stack-privileged-access-management

[8] Palo Alto Networks. (2020). 3 Simple Techniques to Add Security Into the CI/CD Pipeline. Retrieved from https://www.paloaltonetworks.com/blog/2020/10/cloud-add-security-cicd-pipeline/

[9] GitGuardian. (2022). The State of Secrets Sprawl 2022. Retrieved from https://www.gitguardian.com/state-of-secrets-sprawl-report-2022

[10] ArmoSec. (2023). Securing CI/CD Pipelines Through Security Gates. Retrieved from https://www.armosec.io/blog/securing-ci-cd-pipelines-security-gates/

[11] OpsMatters. (2025). Implementing IT Access Management: A DevOps Operations Guide for Streamlined Security Integration. Retrieved from https://opsmatters.com/posts/implementing-it-access-management-devops-operations-guide-streamlined-security-integration

Received: Nov. 03 2025 

**International Journal of Applied Mathematics**

**Volume 38 No. 11s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[12] Academia.edu. (2025). Automating Security Testing in CI/CD Pipelines using DevSecOps Tools: A Comprehensive Study. Retrieved from https://www.academia.edu/129511246/Automating_Security_Testing_in_CI_CD_Pipelines_using_DevSecOps_Tools_a_Comprehensive_Study

[13] ResearchGate. (2021). Integrating Security into CI/CD Pipelines with SAST, DAST, and SCA Tools. Retrieved from https://www.researchgate.net/publication/390459514_Integrating_Security_into_CICD_Pipelines_A_DevSecOps_Approach_with_SAST_DAST_and_SCA_Tools

[12] Heimdal. (2024). 7 Privileged Access Management (PAM) deployment mistakes to avoid. Retrieved from https://emtmeta.com/7-pam-deployment-mistakes-to-avoid-by-heimdal/

[13] Akeyless. (2024). The Hidden Risks of Secrets Mismanagement in CI/CD Pipelines. Retrieved from https://www.akeyless.io/blog/the-hidden-risks-of-secrets-management-in-ci-cd-pipelines/

[14] Wiz. (2024). CI/CD Pipeline Security Best Practices: The Ultimate Guide. Retrieved from https://www.wiz.io/academy/ci-cd-security-best-practices

[15] National Security Agency (NSA) & CISA. (2023). Defending Continuous Integration/Continuous Delivery (CI/CD) Environments. Retrieved from https://media.defense.gov/2023/Jun/28/2003249466/-1/-1/0/CSI_DEFENDING_CI_CD_ENVIRONMENTS.PDF

[16] Fudo Security. (2024). Maximizing Security with DevSecOps PAM Integration for Your Pipeline. Retrieved from https://fudosecurity.com/blog/2024/12/02/maximizing-security-with-devsecops-pam-integration-for-your-pipeline/

[17] BlinkOps. (2024). What Is Security Automation? A Complete Guide. Retrieved from https://www.blinkops.com/blog/security-automation

[18] Admin By Request. (2024). How the PAM Cloud Simplifies Privileged Access Management. Retrieved from https://www.adminbyrequest.com/en/blogs/how-the-pam-cloud-simplifies-privileged-access-management

Received: Nov. 03 2025