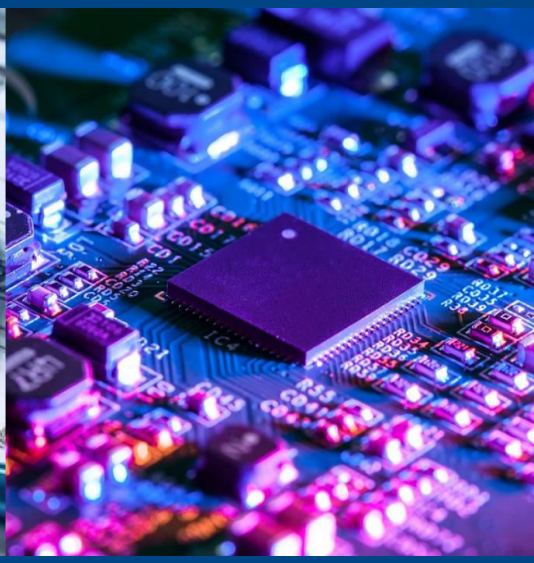
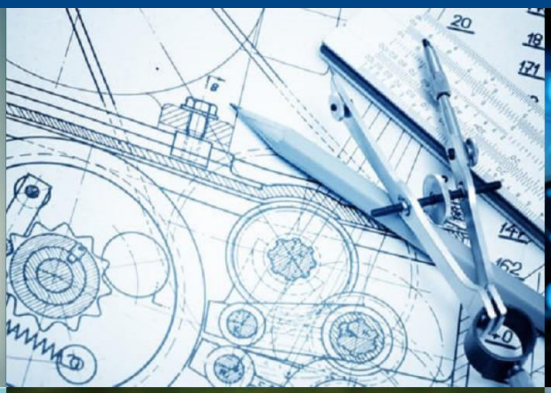
ISSN: 2582-7219

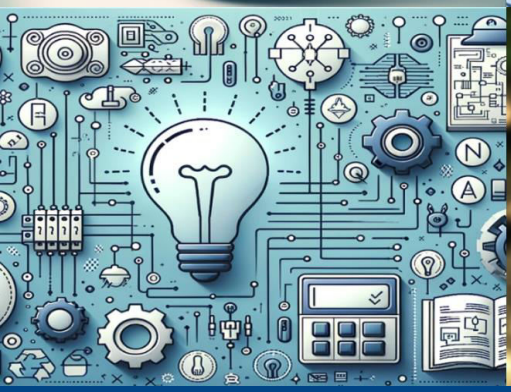




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 9, Issue 3, March 2026**

© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI:10.15680/IJMRSET.2026.0903032

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

AI-Driven Security Automation for Cloud-Native DevSecOps Pipelines

Dr Sudha K, Akshaya R, Agalya S

Department of Computer Science and Business Systems, RMD Engineering College, Chennai, India

ABSTRACT: The paper discusses the transformation of software delivery due to the adoption of cloud-native architectures and automated CI/CD pipelines, which enhance continuous integration and rapid deployment. However, this shift also increases the software supply chain's vulnerability to various risks, including issues such as insecure configurations and identity compromise. Traditional DevSecOps security mechanisms often rely on static scanning tools that produce excessive false positives and lack contextual intelligence.

To address these challenges, the authors propose an Explainable AI-driven Security Risk Scoring Framework designed for cloud-native DevSecOps pipelines. This framework moves beyond reactive security controls by employing a data-driven risk scoring engine that evaluates security posture throughout the pipeline stages. It aggregates security signals from various sources, such as code analysis, dependency metrics, and runtime telemetry from environments like Kubernetes, processing them through a supervised machine learning model to generate a dynamic risk score indicative of deployment threats.

To enhance transparency, the framework includes an Explainable AI (XAI) layer that provides justifications for risk assessments, enabling security teams to understand critical factors influencing pipeline actions. This integration of predictive analytics with interpretability fosters trust, aids in compliance auditing, and strengthens governance within automated DevSecOps workflows.

A prototype demonstrated the framework's effectiveness in a cloud-native pipeline, showcasing improved accuracy in vulnerability prioritization, fewer false positives, and expedited security decision-making processes. The findings suggest that incorporating explainable machine learning into DevSecOps enables a proactive approach to security management, enhancing deployment governance in modern cloud environments.

KEYWORDS: DevSecOps, Cloud-Native Security, AI-Driven Security Automation, CI/CD Pipelines, Multi-Cloud Security, Threat Detection, Zero Trust Architecture, Automated Incident Response.

I. INTRODUCTION

The paper addresses the transformation of software delivery through the adoption of cloud-native architectures and automated CI/CD pipelines, which has increased the software supply chain attack surface and introduced various security risks. Existing DevSecOps security mechanisms often rely on static scanning and rule-based policy enforcement, leading to excessive false positives and limited transparency. The authors propose an Explainable AI-driven Security Risk Scoring Framework designed to provide a unified, data-driven assessment of security posture throughout CI/CD workflows. This framework integrates various security signals and employs a supervised machine learning model to generate a dynamic risk score, enhancing transparency through an XAI layer that explains risk decisions. A prototype implementation demonstrated improved accuracy in vulnerability prioritization and faster decision-making compared to traditional methods, thereby enabling a shift towards proactive, risk-informed security governance in cloud-native environments. This paper proposes an Explainable AI-driven Security Risk Scoring Framework for cloud-native DevSecOps pipelines. The proposed system introduces a unified risk aggregation engine that consolidates multidimensional security signals across development, build, and runtime stages into a dynamic deployment risk score. A supervised machine learning model evaluates contextual features derived from code artifacts, container configurations, dependency graphs, pipeline execution metadata, and runtime telemetry. An integrated explainability layer provides interpretable justifications for each risk prediction, enabling transparent deployment gating decisions and enhancing governance, compliance, and operational trust.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

By combining predictive risk modeling with explainable decision support, the proposed framework shifts DevSecOps security from isolated vulnerability detection toward holistic, risk-aware deployment governance. Experimental validation demonstrates that integrating explainable machine learning into CI/CD pipelines improves prioritization accuracy, reduces false positives, and strengthens decision accountability in modern cloud-native environments.

II. MOTIVATION & RESEARCH GAPS

The integration of security into DevOps pipelines has rapidly evolved with the adoption of DevSecOps practices within Continuous Integration/Continuous Deployment (CI/CD) workflows. These workflows generate ongoing security findings through various tools including Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST). However, this creates a fragmented security landscape, where alerts lack comprehensive contextual interpretation. Security decisions in CI/CD tend to be binary and rule-driven, overlooking important contextual factors such as the likelihood of exploitation and historical attack patterns.

The rise of automation and machine learning models in deployment governance raises trust issues, as black-box AI systems create opaque reasoning paths that hinder validation by security analysts. Critical gaps remain in cloud-native CI/CD frameworks due to isolated security tools, leading to uncorrelated risks across pipeline stages and a lack of unified risk models. Furthermore, the reliance on deterministic scoring systems for deployment gating, such as CVSS ratings, fails to adapt to contextual threats, leading to excessive false positives. The regulatory demand for traceability and accountability complicates matters when existing DevSecOps frameworks offer inadequate auditing capabilities, particularly in multi-cloud environments with diverse security controls.

III.CONTRIBUTIONS OF THIS PAPER

This paper makes the following key contributions to the field of cloud-native DevSecOps security:

A. **AI-Driven Autonomous Security Framework**

This work proposes a novel Explainable AI-driven security risk scoring framework that consolidates heterogeneous security signals across CI/CD pipeline stages into a dynamic and interpretable deployment risk score. Unlike traditional rule-based or severity-threshold mechanisms, the proposed framework models contextual relationships among vulnerabilities, configuration states, and runtime indicators to generate adaptive and data-driven risk assessments.

B. **Multi-Agent Security Architecture**

The paper introduces a structured feature aggregation mechanism that integrates security metrics from static analysis, dependency vulnerability scanning, container assessments, infrastructure-as-code validation, identity access events, and runtime telemetry collected from orchestrated environments such as Kubernetes. This multidimensional feature modeling enables holistic risk evaluation rather than isolated vulnerability inspection.

C. **Adaptive Threat Modeling for DevSecOps Pipelines:**

A supervised machine learning model is designed to compute contextualized deployment risk scores using aggregated security indicators. The framework learns nonlinear relationships between correlated security features and historical incident patterns, enabling predictive estimation of pipeline risk levels prior to deployment approval. This contribution moves DevSecOps security from reactive detection to proactive governance.

D. **Integration of Zero Trust Principles into CI/CD Workflows:**

To address the limitations of black-box AI systems, this paper incorporates Explainable AI techniques to provide feature attribution and interpretable reasoning behind each risk prediction. The explainability layer enables security teams to understand which factors most significantly influenced deployment blocking, approval, or escalation decisions. This enhances operational trust, regulatory compliance, and governance transparency within automated CI/CD environments.

E. **AI-Based Vulnerability Triage and Predictive Threat Analysis:**

The framework introduces a dynamic deployment gating mechanism that leverages computed risk scores to automate CI/CD decision workflows within platforms such as Jenkins. Instead of binary severity thresholds, the proposed system supports graded decision policies, including automatic approval, conditional review, or deployment blocking based on contextual risk levels.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

F. Automated Incident Response and Self-Healing Pipelines:

By modeling feature interactions and contextual dependencies, the proposed approach improves vulnerability prioritization accuracy and significantly reduces false-positive alerts compared to conventional static scoring mechanisms. This contribution mitigates alert fatigue and improves operational efficiency in security operations teams.

IV. ZERO TRUST INTEGRATION AND POLICY ENFORCEMENT

This section describes how Zero Trust principles are systematically integrated into the proposed AI-driven DevSecOps security framework. Unlike traditional perimeter-based security models, Zero Trust assumes no implicit trust between users, services, or pipeline components and enforces continuous verification throughout the software delivery lifecycle.

A. Zero Trust Design Principles in DevSecOps

The proposed framework adopts core Zero Trust principles, including continuous identity verification, least-privilege access, and explicit policy enforcement at every pipeline stage. Each interaction within the DevSecOps pipeline—such as code commits, build execution, artifact access, deployment actions, and runtime service communication—is treated as a potential trust boundary.

Trust decisions are dynamically evaluated based on contextual factors such as identity attributes, execution environment, risk score, and observed behavior. This approach ensures that trust is not statically assumed based on network location or prior authentication but is continuously reassessed throughout pipeline execution.

B. Identity-Aware Access Control

Identity and access management mechanisms are embedded across the pipeline to regulate access to repositories, build agents, container registries, and cloud resources. Authentication and authorization decisions are enforced using strong identity verification mechanisms, including token-based authentication and mutual authentication between services.

Access permissions are dynamically adjusted based on assessed risk levels. For example, elevated privileges required for deployment or infrastructure modification are granted only when risk scores remain below predefined thresholds. This dynamic privilege management reduces the attack surface and limits the impact of compromised credentials or malicious pipeline modifications.

C. Policy-as-Code Enforcement

Security policies are defined and enforced using policy-as-code mechanisms, enabling consistent and auditable policy evaluation across environments. Policies govern actions such as artifact promotion, infrastructure provisioning, network exposure, and runtime behavior.

The policy enforcement engine evaluates policies continuously during pipeline execution and runtime operation. Detected policy violations are correlated with AI-derived risk assessments to determine appropriate enforcement actions. This integration allows the framework to distinguish between benign deviations and high-risk violations, enabling proportional and context-aware responses.

V. IMPLEMENTATION DETAILS AND TECHNOLOGY STACK

This section describes the implementation of the proposed AI-driven security automation framework, detailing the DevSecOps pipeline configuration, technology stack, and system integration. The implementation is designed to demonstrate feasibility in real-world cloud-native environments while maintaining modularity and extensibility.

A. Cloud-Native DevSecOps Pipeline Setup

The prototype implementation is deployed in a multi-cloud environment leveraging public cloud platforms to reflect realistic operational conditions. CI/CD pipelines are configured using industry-standard automation tools to orchestrate code integration, build, test, and deployment stages. Source code repositories serve as the entry point for pipeline execution, triggering automated workflows upon code commits or configuration changes.

Containerization is employed to package application components and security tools, ensuring consistent execution across environments. Container images are built and stored in secure registries, while orchestration platforms manage deployment, scaling, and runtime operations. Infrastructure provisioning and configuration are defined declaratively using infrastructure-as-code templates, enabling version-controlled and repeatable infrastructure deployment.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

B. **Security Tool Integration**

Multiple security scanning tools are integrated into the pipeline to analyze different artifact types. Static analysis tools inspect source code for insecure patterns, dependency scanners identify vulnerable third-party libraries, and configuration scanners validate infrastructure-as-code templates against security best practices. Container image scanners assess vulnerabilities in base images and application layers before deployment.

These tools are executed as pipeline stages or sidecar services, generating structured security reports that are forwarded to the AI intelligence layer. Tool outputs are normalized into a common schema to facilitate consistent analysis and correlation across heterogeneous security signals.

C. AI and Automation Components

The AI intelligence layer is implemented using a modular architecture that supports independent deployment of machine learning models and reasoning agents. Machine learning components are developed using Python-based frameworks and are responsible for classification, anomaly detection, and risk scoring. Models are trained and evaluated using historical security data and continuously updated based on feedback from incident response outcomes.

Large language model components are integrated through controlled interfaces to support vulnerability triage, alert summarization, and policy interpretation. Automation logic is implemented using event-driven scripts and webhooks that translate AI decisions into actionable responses within the CI/CD pipeline and runtime environment.

D. **Design Considerations**

The implementation emphasizes scalability, minimal pipeline overhead, and interoperability with existing DevSecOps tooling. Components are loosely coupled to allow independent updates and future extension. The framework is designed to integrate seamlessly into existing pipelines without requiring major architectural changes, facilitating adoption in practical enterprise environments.

VI. RESULTS AND PERFORMANCE EVALUATION

This section presents the experimental results obtained from evaluating the proposed AI-driven security automation framework and compares its performance against a baseline DevSecOps pipeline without AI-assisted analysis and automated response.

A. Threat Detection Performance

The proposed framework demonstrates a notable improvement in threat detection accuracy compared to the baseline configuration. Across all evaluated scenarios, the AI-driven framework achieved higher precision in identifying security issues originating from code vulnerabilities, infrastructure misconfigurations, and runtime anomalies.

In particular, the integration of anomaly detection models enabled the identification of abnormal behaviors that were not detected by traditional rule-based scanners. These results indicate that the framework is effective not only in detecting known vulnerabilities but also in identifying previously unseen or behavior-based threats.

B. False Positive Reduction

One of the key challenges in existing DevSecOps pipelines is the high volume of false-positive alerts generated by independent security tools. Experimental results show that the proposed framework significantly reduces false positives by correlating findings across multiple pipeline layers and applying contextual risk scoring.

Compared to the baseline setup, the AI-driven framework reduced the false-positive rate by a substantial margin. This reduction directly improved alert quality and

reduced manual analysis effort for developers and security teams.

C. Detection and Response Time

The event-driven design and continuous analysis capabilities of the framework resulted in improved detection and response times. The mean time to detect (MTTD) security incidents was reduced due to real-time processing of security signals and anomaly detection mechanisms.

Similarly, the mean time to respond (MTTR) was significantly lower in the proposed framework, as automated mitigation actions were executed immediately after risk thresholds were exceeded. These results demonstrate the effectiveness of automation in minimizing exposure time and limiting the potential impact of security incidents.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VII. CONCLUSION AND FUTURE WORK

The integration of security into DevOps has led to the rise of DevSecOps practices, with continuous security findings generated through various testing and monitoring tools like SAST, DAST, and container scanning. However, this has resulted in a fragmented security landscape where alerts lack cohesive contextual interpretation. Security decisions in CI/CD pipelines are typically binary, based on threshold scores that ignore contextual factors like vulnerability relationships and behavioral anomalies. The introduction of machine learning in automation creates trust issues, as black-box AI predictions cannot be easily validated by security professionals. Research gaps persist in cloud-native CI/CD environments, particularly due to the isolation of security tools and the reliance on deterministic severity-based mechanisms, which fail to incorporate contextual risk factors, leading to potential oversight of significant threats.

REFERENCES

[1] N. Sundari, P. Manja, P. Mathur, and P. B. Honnavalli, “Extensive Review of Threat Models for DevSecOps,” IEEE Access, vol. 13, pp. 45252–45278, 2025, doi: 10.1109/ACCESS.2025.3547932.
[2] T. Rangnau, R. van Buijtenen, F. Fransen, and F. Turkmen, “Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines,” in Proc. IEEE 24th Int. Enterprise Distributed Object Computing Conf. (EDOC), 2020, pp. 145–154.
[3] N. Tomas, J. Li, and H. Huang, “An Empirical Study on Culture, Automation, Measurement, and Sharing of DevSecOps,” in Proc.IEEE Int. Conf. Software Architecture Companion (ICSA-C), 2019,
[4] pp. 1–8.
[5] A. Agung and H. Kabetta, “Integrating Security into DevOps Pipeline Using Static and Dynamic Security Testing,” in Proc. Int. Conf. Information Technology Systems and Innovation (ICITSI), 2018, pp. 1–6.
[6] M. Bajpai and G. Lewis, “Managing Misconfigurations in Cloud-Native DevSecOps Pipelines,” IEEE Software, vol. 37, no. 5, pp. 72–79, 2020.
[8] Yasar, “Security Analysis Across DevOps Pipelines: Challenges and Opportunities,” IEEE Security & Privacy, vol. 18, no. 4, pp. 42–49, 2020.
[9] S. Ibrahim, R. A. Khan, and S. U. Khan, “Infrastructure as Code Security: A Systematic Review,” IEEE Access, vol. 8, pp. 117273–117294, 2020.
[10] J. Pecka and N. Stephen, “Security Threats in CI/CD Pipelines: An Empirical Analysis of DevSecOps Environments,” in Proc. IEEE Int. Conf. Cloud Engineering (IC2E), 2021, pp. 1–10.
[11] Carutan and R. Goya, “Financial and Security Challenges of Adopting DevSecOps in Cloud-Native Platforms,” IEEE Cloud Computing, vol. 8, no. 2, pp. 56–64, 2021.
[12] Cankar, M. Zoric, and A. Pavlic, “AI-Assisted Static and Dynamic Security Analysis of Infrastructure as Code,” IEEE Access, vol. 9, pp. 134221–134235, 2021.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com