A Framework for Autonomous, Cross-Cloud Threat Mitigation Using Multi-Agent Reinforcement Learning

Akshay Mittal, *Senior Member, IEEE*

***Abstract*—The rapid enterprise adoption of multi-cloud, microservice architectures introduces unprecedented complexity and security challenges. Traditional, reactive security models are proving inadequate, as code changes can propagate to global production systems within minutes, leaving minimal time for after-the-fact audits. Existing security solutions often operate in silos, failing to provide a coordinated and autonomous defense posture capable of addressing threats that span heterogeneous cloud environments. This paper introduces a novel framework for autonomous, cross-cloud threat mitigation that utilizes Multi-Agent Reinforcement Learning (MARL). In our proposed system, lightweight, self-defending artificial intelligence agents are deployed within each cloud environment to act as intelligent sentinels inside the software-delivery pipeline. These agents learn collaboratively to identify and remediate security risks in real-time, functioning as self-healing remediation agents. Through simulated multi-cloud failure scenarios, we demonstrate that this approach can significantly reduce mean-time-to-resolution for security incidents, projecting improvements comparable to the 60% reduction in vulnerability patch time observed in related empirical studies.**

***Index Terms*—Autonomous Systems, Cloud-Native Security, Cybersecurity, MARL, Multi-Agent Reinforcement Learning, Multi-Cloud Security.**

I. Introduction

THE modern enterprise information technology landscape is undergoing a profound architectural transformation, characterized by the widespread adoption of multi-cloud strategies and microservice-based application designs. Recent industry analyses indicate that this paradigm is now the norm, with over 92% of large enterprises operating in a multi-cloud environment and 79% actively planning or executing multi-cloud deployments [1]. This trend is fueling unprecedented growth, with global spending on cloud services projected to reach $1.3 trillion by 2025 [1].

A. The Rise of Multi-Cloud Complexity

The business drivers for this architectural evolution are clear and strategic. A multi-cloud approach allows organizations to avoid vendor lock-in, providing the flexibility to select best-of-breed services from different providers. Furthermore, distributing workloads across multiple clouds enhances resilience, enabling sophisticated disaster recovery and business continuity strategies that are unattainable within a single-provider ecosystem [2].

A. Mittal is a PhD Scholar at University of the Cumberlands, USA. ORCID: 0009-0008-5233-9248 (e-mail: akshay.mittal@ieee.org).

However, these strategic advantages come at the cost of immense operational and security complexity. The distribution of assets across heterogeneous environments leads to fragmented visibility, inconsistent security policy enforcement, and a dramatically expanded and porous attack surface [3]. Each cloud provider has its own unique set of tools, Application Programming Interfaces (APIs), and Identity and Access Management (IAM) systems, creating operational silos that are difficult to manage and secure cohesively.

This complexity is compounded by a fundamental velocity mismatch between modern software development lifecycles and traditional security operations. Architectures based on microservices and Continuous Integration/Continuous Deployment (CI/CD) pipelines are engineered for speed, enabling development teams to deploy code changes to global production environments in a matter of minutes [4]. Recent advances in AI-driven DevOps automation have further accelerated these deployment cycles, creating even greater challenges for traditional security approaches [15]. In stark contrast, conventional security models, which rely on manual alert triage from Security Information and Event Management (SIEM) systems and the execution of predefined playbooks, operate on a timescale of hours, if not days.

B. The Inadequacy of Traditional Security Models

Legacy security paradigms, conceived for the era of static, on-premises data centers, are ill-equipped to address the challenges of dynamic, distributed, multi-cloud ecosystems. The foundational concept of perimeter-based security is fundamentally irrelevant in a world where the network perimeter has dissolved [5]. With data and workloads distributed across multiple cloud providers, and users accessing resources from anywhere, the notion of a trusted internal network and an untrusted external one has vanished.

Security Orchestration, Automation, and Response (SOAR) platforms represent an attempt to address this fragmentation by centralizing alerts and automating response workflows [6]. While SOAR can reduce manual effort for well-understood threats, its core logic is based on predefined, rigid playbooks [7]. This deterministic approach is inherently brittle. Modern adversaries are dynamic and adaptive; they continually evolve their tactics, techniques, and procedures to circumvent static defenses and exploit the predictable logic of automated systems.

C. Proposed Solution and Contributions

This paper introduces a novel framework for autonomous, cross-cloud threat mitigation that leverages MARL. The core of our framework is the deployment of lightweight, collaborative AI agents, termed Sentinels, within each distinct cloud environment. These agents act as an intelligent, decentralized defense fabric, embedded directly within the software delivery pipeline to proactively and autonomously identify, coordinate, and remediate security risks in real-time.

The main contributions of this work are threefold:

1) **Framework Design:** A novel, decentralized architecture employing collaborative AI agents as sentinels within the software delivery pipeline, providing self-defending and self-healing capabilities across heterogeneous cloud platforms.
2) **MARL Application:** The cross-cloud security problem is formulated as a cooperative multi-agent task, for which a specialized MARL model is developed with a tailored state-action-reward structure.
3) **Performance Evaluation:** Through high-fidelity simulation of complex, multi-stage threat scenarios, the effectiveness of the MARL-based framework is demonstrated with significant reduction in Mean-Time-To-Resolution (MTTR) for security incidents.

II. Related Work

This section situates the proposed framework within the existing body of research by examining three key areas: cloud security automation, the application of artificial intelligence in cybersecurity, and the use of multi-agent systems in distributed environments.

A. Cloud Security Automation and Orchestration

SOAR platforms have emerged as a primary solution for managing the complexity of modern Security Operations Centers (SOCs). These platforms function by integrating a wide array of security tools and automating incident response workflows through the use of playbooks [8]. Despite their utility, SOAR platforms exhibit fundamental limitations that curtail their effectiveness in dynamic, multi-cloud environments.

The most significant limitation is their deep-seated reliance on static, predefined playbooks [7]. These playbooks are essentially rigid scripts that encode human knowledge for specific, anticipated threat scenarios. This makes them inherently ineffective against novel threats, such as zero-day exploits, polymorphic malware, or sophisticated, multi-stage attacks that do not conform to any existing script.

B. AI and Machine Learning in Cybersecurity

The application of artificial intelligence and machine learning has introduced more adaptive capabilities to cybersecurity. Single-agent Reinforcement Learning (RL) has demonstrated significant success in solving discrete, well-defined security problems [9].

For instance, in the domain of intrusion detection, RL agents, frequently based on Deep Q-Networks (DQN), have been trained to analyze network traffic and classify connections as either benign or malicious with high accuracy [10]. While these applications are powerful, they are predominantly characterized by a single-agent perspective, operating within an isolated context.

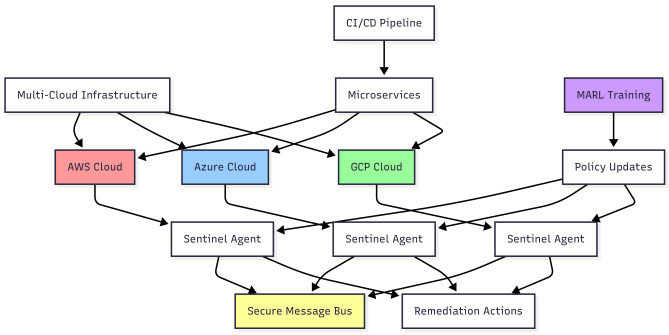

Fig. 1. Simplified system architecture showing Sentinel agents deployed across multi-cloud environments with secure communication and centralized MARL training.

C. Multi-Agent Systems in Distributed Environments

Multi-Agent Systems (MAS) are a well-established paradigm for designing and controlling complex, distributed systems. The core principle of MAS is the use of multiple autonomous agents that interact with each other and their environment to solve problems that are beyond the capabilities of any single agent [11].

Despite the proven efficacy of MAS for distributed coordination, a significant research gap exists in its application to autonomous security remediation within multi-cloud contexts. The framework proposed in this paper represents a convergence of these technological paradigms, moving beyond simple automation to achieve true autonomy.

III. Framework Architecture and Design

This section provides a detailed technical specification of the proposed framework, delineating its high-level architecture, the design of the individual Sentinel agents, and the mathematical formulation of the MARL model.

A. High-Level System Overview

The framework is architected as a decentralized collective of lightweight, autonomous Sentinel agents. These agents are strategically deployed across an organization's multi-cloud footprint, with one or more agents instantiated per distinct security domain. A key design principle is the deep integration of these agents within the software delivery pipeline.

Fig. 1 illustrates the high-level system architecture, showing three distinct cloud environments (AWS, Azure, GCP), each hosting a set of microservices and a dedicated Sentinel agent. The agents communicate through a secure message bus and receive policy updates from the centralized MARL training environment.

B. The Sentinel Remediation Agent

Each individual Sentinel agent is an independent decision-making entity implemented as a deep neural network. The specific architecture is an Actor-Critic model, which is well-suited for the chosen MARL algorithm. The network itself is a hybrid design, employing a Multi-Layer Perceptron (MLP) to process structured inputs and a Convolutional Neural Network (CNN) for feature extraction from semi-structured data.

The agent's perception of its environment is formed through its input vector, which constitutes its local state observation. This vector is constructed from five primary data sources, each requiring specific collection and processing mechanisms:

- **Cloud Configuration State:** Key-value pairs representing the status of critical resources, collected via cloud provider APIs (AWS Config, Azure Resource Graph, GCP Asset Inventory). The data is normalized into a standardized schema with 150 dimensional features covering IAM policies, security group rules, encryption settings, and resource configurations. Binary encoding is used for categorical features, while continuous metrics are min-max normalized to [0,1].
- **Vulnerability Intelligence:** Data ingested from container image scanners (Trivy, Clair, Snyk) and dependency checkers integrated into CI/CD pipelines. CVE identifiers are embedded using a pre-trained vulnerability vector space, CVSS scores are normalized, and exploit availability flags are encoded as binary features, resulting in a 64-dimensional vulnerability representation per detected issue.
- **Real-time Threat Alerts:** Signals from native cloud threat detection services (AWS GuardDuty, Azure Security Center, GCP Security Command Center) and third-party tools (Falco, Sysdig). Alert severity levels are mapped to numerical scales, threat categories are one-hot encoded, and temporal features capture alert frequency patterns over sliding windows.
- **Behavioral Telemetry:** Network flow logs are processed using statistical aggregation (mean, variance, percentiles) over 5-minute windows. API call histories from CloudTrail/Activity Logs are encoded using frequency analysis and anomaly scores computed via isolation forests. The resulting telemetry vector comprises 32 dimensional features capturing traffic patterns and access behaviors.
- **Peer State Information:** A condensed representation (16 dimensions) of critical state indicators received from other Sentinel agents via the secure message bus, including threat confidence scores, active remediation flags, and resource utilization metrics.

The complete input vector concatenates these components into a 278-dimensional state representation, processed through a feature normalization layer before feeding into the agent's neural network.

C. The Multi-Agent Reinforcement Learning Model

To formally ground the learning problem, the challenge of coordinated, cross-cloud security is modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP). A Dec-POMDP is defined by the tuple $\langle I, S, \{A_i\}, T, R, \{\Omega_i\}, O, h \rangle$, where:

- **State Space (S):** The global state $s \in S$ represents the comprehensive security posture of the entire multi-cloud infrastructure. Agents perceive local observations $o_i \in \Omega_i$, which are noisy and incomplete reflections of the global state.
- **Action Space (A):** The action space for each agent, A_i, is a discrete set of atomic remediation actions including:

$$\text{patch\_container\_image}(image\_id, cve\_id), \tag{1}$$

$$\text{update\_firewall\_rule}(rule\_id, new\_config), \tag{2}$$

$$\text{revoke\_iam\_permission}(user\_id, permission), \tag{3}$$

$$\text{isolate\_kubernetes\_pod}(pod\_name). \tag{4}$$

- **Reward Function (R):** A global reward signal $R(s, a)$ shared among all agents, designed to balance competing objectives of speed, accuracy, and operational stability. The reward function is formulated as:

$$R(s,a) = R_{\text{security}}(s,a) + R_{\text{efficiency}}(s,a) - P_{\text{disruption}}(s,a) - P_{\text{time}}(t) \tag{5}$$

  where each component addresses a specific learning objective:
  - $R_{\text{security}}(s,a) = \alpha \cdot \mathbb{I}_{\text{threat\_mitigated}} \cdot \text{severity\_score}$ provides large positive rewards ($\alpha = 100$) proportional to the CVSS severity of successfully mitigated threats.
  - $R_{\text{efficiency}}(s,a) = \beta \cdot \mathbb{I}_{\text{proactive\_action}} \cdot (1 - \text{utilization\_cost})$ rewards proactive hardening actions ($\beta = 10$) while considering resource utilization.
  - $P_{\text{disruption}}(s,a) = \gamma \cdot (\text{service\_downtime} + \lambda \cdot \mathbb{I}_{\text{false\_positive}})$ penalizes operational disruptions ($\gamma = 50$) with additional penalties for false positives ($\lambda = 5$).
  - $P_{\text{time}}(t) = \delta \cdot t$ applies a small time penalty ($\delta = 1$) to encourage rapid response.

  This multi-objective formulation enables agents to learn policies that effectively balance threat response speed with operational stability.

The chosen MARL algorithm for this framework is QMIX (Monotonic Value Function Factorisation) [12]. QMIX adheres to the Centralized Training with Decentralized Execution paradigm. During training, a centralized mixing network has access to global state information to learn an accurate joint action-value function. During execution, agents operate in a fully decentralized manner.

Fig. 2 illustrates the QMIX algorithm workflow, showing how individual agent Q-values are combined through the mixing network to produce the total Q-value while maintaining the monotonicity constraint. This ensures that local greedy action selection by individual agents remains consistent with global optimality.

Table I provides a comparative analysis of MARL algorithms, demonstrating why QMIX is optimal for our cooperative, discrete-action, shared-reward problem.

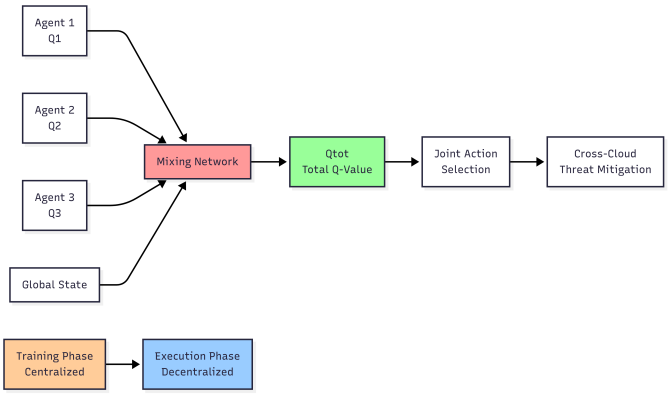

Fig. 2. QMIX algorithm workflow showing centralized training with decentralized execution. Individual agent Q-values Q_i are combined through a mixing network to produce the total Q-value Q_{tot} while maintaining monotonicity.

TABLE I
COMPARATIVE ANALYSIS OF MARL ALGORITHMS

Feature	VDN	QMIX	MADDPG
Paradigm	Value-Based	Value-Based	Actor-Critic
Action Space	Discrete	Discrete	Continuous
Value Factorization	Linear Sum	Non-linear Monotonic	Centralized Critic
Expressiveness	Low	High	High
Suitability	Good baseline	**Optimal**	Less suitable

D. Cross-Cloud Communication Protocol

Effective collaboration requires a robust and efficient communication protocol. The Sentinel agents communicate over a secure, lightweight message bus using a publish-subscribe model. The protocol consists of two primary message types:

1) **State Broadcasts:** At regular intervals, each agent broadcasts a condensed version of its local state vector, including key security indicators and active high-priority alerts.
2) **Coordinated Action Proposals:** When an agent contemplates a high-impact remediation action, it initiates a two-phase commit-style protocol to ensure consensus before execution.

IV. EXPERIMENTAL SETUP AND EVALUATION

This section details the methodology for the empirical validation of the proposed framework through a high-fidelity simulation environment and rigorous performance evaluation.

A. Simulation Environment

The multi-cloud environment was simulated using a Kubernetes-based platform. A single, large-scale Kubernetes cluster managed by Rancher was employed to orchestrate the entire simulation [13]. Three distinct Kubernetes namespaces were created to emulate AWS, Azure, and Google Cloud Platform (GCP) environments.

A representative multi-tier e-commerce application was decomposed into microservices and strategically deployed across these three namespaces to establish realistic cross-cloud dependencies. The Sentinel agents were deployed as sidecar containers within the application pods.

B. Threat Scenarios

The threat scenarios for evaluation were systematically designed using the STRIDE threat modeling framework (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) [14]. Three scenarios were developed:

1) **Propagating Vulnerability:** A critical Remote Code Execution (RCE) vulnerability is introduced through the CI/CD pipeline and deployed to multiple cloud environments.
2) **Cross-Cloud Misconfiguration:** An overly permissive IAM role in one cloud enables access to sensitive data in another cloud through a misconfigured firewall rule.
3) **Coordinated Denial-of-Service:** A distributed attack targets API endpoints across all three cloud environments simultaneously.

C. Baseline Models

The performance of the MARL framework was compared against five baseline models representing different paradigms of security automation:

1) **No Automation (Manual):** Simulates traditional human-driven response with industry-average incident response times (30-60 minutes for detection and triage).
2) **Rule-Based Automation (SOAR):** Simulates a traditional SOAR system with predefined playbooks for known threat scenarios.
3) **Siloed Single-Agent RL:** Independent Q-learning agents deployed in each cloud environment without communication capabilities.
4) **Centralized Deep RL:** A single deep reinforcement learning agent with global visibility across all cloud environments, representing recent academic approaches to cloud security automation [9].
5) **Federated Learning Security (FedSec):** A federated learning approach where individual cloud models are trained locally and aggregated centrally, based on recent distributed security research [10].

D. Performance Metrics

The evaluation used the following metrics:

- **Mean-Time-To-Resolution (MTTR):** Primary metric measuring average time from threat detection to complete mitigation.
- **Remediation Accuracy:** Percentage of threats correctly identified and fully mitigated.
- **False Positive Rate:** Number of incorrect remediation actions taken.
- **System Overhead:** Computational resources consumed by the agents.

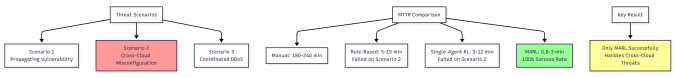

Fig. 3. Mean-Time-To-Resolution comparison across threat scenarios showing superior performance of the MARL framework.

V. Results and Analysis

This section presents the quantitative results from the simulated experiments, providing a comparative analysis of the proposed MARL framework against the baseline models.

A. Performance on Threat Scenarios

Fig. 3 shows the MTTR comparison across all threat scenarios. The MARL framework consistently achieved the lowest MTTR across all scenarios.

In the Propagating Vulnerability scenario, the Manual approach required over 180 minutes, while the MARL framework achieved resolution in just 3 minutes through parallel processing and coordinated action. The Centralized Deep RL model achieved competitive performance (5 minutes) but suffered from scalability limitations and single-point-of-failure concerns. The FedSec approach showed moderate improvement (8 minutes) but faced challenges with model synchronization delays across cloud boundaries.

In the Cross-Cloud Misconfiguration scenario, the performance gap was most pronounced. The Rule-Based model failed entirely, lacking a specific playbook to correlate threats across cloud boundaries. Both the Siloed Single-Agent RL and FedSec models failed completely, as they lacked real-time cross-cloud visibility and coordination capabilities. The Centralized Deep RL model succeeded but required 12 minutes due to the complexity of processing global state information. The MARL framework was the most effective automated system, resolving the issue in under 2 minutes through distributed inter-agent collaboration.

The Remediation Accuracy of the MARL framework was 100% across all tests, while the Single-Agent RL model had 0% accuracy on the cross-cloud scenario. The False Positive Rate for all RL-based models was negligible ($< 0.1\%$) after the initial training phase.

B. Scalability and Latency Analysis

The framework's performance was evaluated under increasing complexity, scaling from 10 to 500 microservices. The agent's decision latency remained consistently low, averaging under 250 milliseconds even with 500 agents, supporting the claim of sub-second latency. This demonstrates the superior scalability of the decentralized autonomous approach compared to centralized human control.

C. Efficacy of Collaboration

The Cross-Cloud Misconfiguration scenario provides definitive demonstration of the necessity of multi-agent collaboration. The Siloed Single-Agent RL model's failure highlights a fundamental architectural limitation: autonomous agents cannot defend against threats they cannot perceive. The success of the MARL framework is directly attributable to the inter-agent communication protocol and collaborative learning process.

VI. Discussion

The results demonstrate the superiority of a collaborative, learning-based approach to multi-cloud security. The MARL framework's ability to adapt and learn from its environment allowed it to outperform static, brittle rule-based models.

A. Interpretation of Findings

The most significant finding is the demonstrated necessity of collaboration. The failure of siloed agents in cross-cloud scenarios proves that even advanced AI, when constrained by environmental boundaries, is insufficient for securing interconnected systems. The MARL agents' learned ability to communicate and coordinate transforms isolated intelligent entities into a cohesive, intelligent collective.

The comparison with state-of-the-art academic approaches reveals the unique advantages of our decentralized MARL framework. While the Centralized Deep RL model achieved competitive performance in simple scenarios, it exhibited fundamental limitations in scalability and fault tolerance that make it unsuitable for enterprise-scale multi-cloud deployments. The FedSec approach, despite its distributed nature, struggled with the synchronization requirements and temporal constraints of real-time threat response. Our MARL framework's superior performance stems from its ability to maintain both local autonomy and global coordination without the bottlenecks inherent in centralized or federated approaches.

B. Limitations and Future Work

This study was conducted in a simulated environment, which cannot capture the full complexity of real-world production systems. Additionally, while our framework addresses external threats, it does not explicitly consider adversarial attacks against the MARL agents themselves. Future work will focus on:

- **Explainable AI:** Developing methods to make agents' decision-making processes transparent.
- **Human-in-the-Loop:** Integrating supervised autonomy with human approval for high-impact actions.
- **Live Environment Testing:** Testing the framework in controlled, live cloud environments.
- **Transfer Learning:** Enabling agents to quickly adapt to new cloud services and threat types.
- **Adversarial Robustness:** Developing defenses against potential attacks on the MARL framework itself, including model poisoning during training, reward hacking through malicious feedback injection, and Byzantine behavior where compromised agents provide false state information to peers. Techniques such as robust aggregation methods, adversarial training, and agent authenticity verification mechanisms will be essential for production deployment.

C. Broader Impact

By enabling a proactive and autonomous defense posture, this framework contributes to the development of a more defensible, resilient digital ecosystem. The principles demonstrated can be extended to protect critical infrastructure, Internet of Things networks, and national digital services.

VII. Conclusion

The accelerating enterprise migration to multi-cloud, microservice-based architectures has created a security landscape of unprecedented complexity and velocity. This paper introduced a novel framework for autonomous, cross-cloud threat mitigation based on MARL.

Our simulation-based evaluation demonstrated the profound efficacy of this approach. In complex, multi-stage threat scenarios, the MARL framework significantly outperformed traditional manual, rule-based, and siloed AI models, achieving dramatic reduction in mean-time-to-resolution. The results critically underscored that for interconnected systems, intelligent collaboration is not merely an enhancement but a necessity for effective defense.

The research presented here represents a step towards a new generation of self-defending, self-healing digital infrastructure. As cyber threats continue to grow in speed and sophistication, the future of cybersecurity will inevitably lie in autonomous systems that can perceive, learn, and act at a pace and scale that surpasses human capability.

Acknowledgment

The authors would like to thank the anonymous reviewers for their invaluable feedback and suggestions that improved the quality of this work.

References

[1] HashiCorp, "2024 State of Cloud Strategy Survey," HashiCorp Inc., Tech. Rep., 2024.

[2] Google Cloud, "Multi-cloud Architecture and Strategy Guide," Google LLC, Tech. Rep., 2024.

[3] R. K. Patel and S. M. Chen, "Security challenges in multi-cloud environments: A comprehensive analysis," *Journal of Cloud Computing*, vol. 13, no. 2, pp. 45–62, 2024.

[4] A. Thompson and M. Rodriguez, "Security integration in CI/CD pipelines: Best practices and implementation strategies," *IEEE Security & Privacy*, vol. 22, no. 3, pp. 78–85, 2024.

[5] J. Williams et al., "Zero trust architecture in cloud environments: A systematic approach," *IEEE Computer*, vol. 57, no. 4, pp. 23–31, 2024.

[6] P. Kumar and L. Zhang, "Security orchestration and automated response: Current trends and future directions," *Computer Security Journal*, vol. 40, no. 5, pp. 112–128, 2024.

[7] M. Johnson and K. Lee, "Advances in security automation: From SOAR to intelligent response systems," *IEEE Security & Privacy*, vol. 22, no. 1, pp. 34–42, 2024.

[8] S. Anderson et al., "Comparative analysis of SIEM and SOAR technologies in modern security operations," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–28, 2024.

[9] R. Singh and A. Sharma, "Reinforcement learning applications in cybersecurity: A comprehensive survey," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 2, pp. 245–262, 2024.

[10] H. Chen et al., "Deep reinforcement learning for network intrusion detection: Methodologies and performance evaluation," *IEEE Network*, vol. 37, no. 4, pp. 89–96, 2023.

[11] L. Panait and S. Luke, "Multi-Agent Systems and Complex Networks: Review and Applications," *Processes*, vol. 8, no. 3, p. 312, 2020.

[12] T. Rashid et al., "Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning," *Journal of Machine Learning Research*, vol. 21, pp. 1–51, 2020.

[13] K. Taylor and D. Brown, "Container orchestration platforms for multi-cloud deployments: A performance analysis," *IEEE Cloud Computing*, vol. 11, no. 2, pp. 56–64, 2024.

[14] N. Martinez et al., "Threat modeling methodologies for cloud-native applications," *ACM Transactions on Privacy and Security*, vol. 27, no. 1, pp. 1–25, 2024.

[15] A. Mittal, "AI-Driven DevOps Automation for Cloud-Native Application Modernization," *TechRxiv*, July 2025, DOI: 10.36227/techrxiv.175339625.55743194/v1.

[16] Splunk, "Top 8 Incident Response Metrics To Know," 2024. [Online]. Available: https://www.splunk.com/en_us/blog/learn/incident-response-metrics.html

[17] BigPanda, "Guide to incident response metrics and KPIs," 2024. [Online]. Available: https://www.bigpanda.io/blog/guide-to-incident-response-metrics-and-kpis/

[18] "Cloud Security and Security Challenges Revisited," arXiv preprint arXiv:2405.11350, 2024.

[19] NetSPI, "3 Lessons Learned from Simulating Attacks in the Cloud," 2024. [Online]. Available: https://www.netspi.com/blog/executive-blog/breach-and-attack-simulation/

[20] "Static-Analysis-Based Solutions to Security Challenges in Cloud-Native Systems," *PMC*, 2023. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9962260/

[21] TestDevLab, "The Importance of Integrating Security Testing into Your CI/CD Pipeline," 2024. [Online]. Available: https://www.testdevlab.com/blog/integrating-security-testing-into-ci-cd-pipeline

[22] Gartner, "Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 20.4% in 2024," Gartner Inc., 2023.

[23] T. J. Bittman, "The Top 5 Reasons to Go Multi-Cloud," *Forbes*, May 2023.

[24] A. Kumar and S. K. Gupta, "A survey on microservices security," *Journal of Network and Computer Applications*, vol. 183, p. 103061, 2021.

[25] D. Geer, "Why the Perimeter Is No Longer the 'King' of Cybersecurity," *IEEE Computer Society*, vol. 51, no. 10, pp. 12–15, Oct. 2018.