International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET2411159

Strengthening DevOps Security with Multi-Agent Deep Reinforcement Learning Models

Phani Monogya Katikireddi
Independent Researcher, USA

Article Info
Volume 9, Issue 2

Publication Issue :
March-April-2022

Article History
Accepted : 06 April 2022
Published: 14 April 2022

Page No : 497-502

ABSTRACT

DevOps practices have significantly changed software development and delivery processes by promoting integration and deployment. However, it comes with many security issues through the increased speed, dynamic, and automation of the developed DevOps pipelines; these are misconfigurations, errors within the dependencies, and various external threats. In this paper, we discuss using Multi-Agent Deep Reinforcement Learning (MADRL) models to enhance security in DevOps environments. In other words, through agent-to-agent cooperation, they learn in the environment to adapt to new threats proactively and provide vulnerability control and real-time response. This research focuses on the significant strengths of MADRL, such as the model's ability to scale naturally to many agents, its ease of envisioning different threat levels, and its flexibility in adapting to most threat scenarios. Using simulation results, it can be proved that the proposed MADRL models can be employed to learn security policies, discover unfamiliar patterns, and control risks to achieve effective DevOps security automation. This work demonstrates the extent to which MADRL can help transform the complexity of the new security problems we encounter in delivering software pipelines in today's sophisticated environment.

Keywords : DevOps, Security, Multi-Agent Systems, Deep Reinforcement Learning, Vulnerability Management, Proactive Threat Detection, Automation, Scalability.

Introduction

DevOps has become an innovative approach to software delivery and operation, which leads to a fully automated process with the integration of multiple teams. DevOps is, therefore, seen to assist in the quick and efficient delivery of the software by eradicating barriers put in place by the development and operational departments. On the same note, the dynamic and automated nature of the DevOps pipelines is a leading cause of increased security risks. The six lack correct configurations, uncovered susceptibilities, flawed code references or dependencies, and high speeds at which drifts or perils spread across systems. That is why more conventional security methods fail to sufficiently address these issues, as they are either reactive or rely on extensive use of manual processes.

In this respect, organizations need to adopt smart and elastic form of security necessary to grow ensuring the pipelines against the existing threats with high response time. This issue can be addressed using an MADRL – a novel concept. Thus, through cooperation between independent learners, MADRL can permanently protect, detect, and prevent security threats in DevOps. These agents mimic transactions happening in the System and bring contextuality to the policies, as well as ameliorating threat handling without oversight.

Simulation report

A simulated world using DevOps cloud-native pipelines was created to compare the effects of Multi-Agent Deep Reinforcement Learning for DevOps security. This scenario mimicked the traditional DevOps features, such as CI/CD/CD, and integrated security exercises, such as misconfiguration, dependency vulnerability, and external intrusion.

Simulation Setup

The simulated setup was a private real multi-cloud engaged with several microservices, including fundamental small-scale structures from the theory of Mfula and Nurminen (2018), for encouraging fault tolerance and scalability. The MADRL model, working with the System, incorporated multiple agents dedicated to addressing given security event occurrences within such services. The environment was provided with the possibility to detect anomalies and to simulate real-time threats based on the Symbolic Reasoning done by Calegari et al. (2018).

Process and Implementation

Training in Artificial Intelligence was performed using deep Q-learning techniques based on exercises analyzed by Stadelmann et al. (2018) on deep learning frameworks. Initially, no reward signals were provided for the particular security policies; however, in the subsequent transactions, the agent independently acquired the security policies by interacting with the System and updating the policy according to the reward signals it received. Threats encompassed synthetic injection, unauthored access, and resource exploitation, according to the survey by da Rosa Righi et al. (2019).

Results

The practical application of the integrated security model with an implementation of the MADRL solved the problem and improved the results in threat recognition and security improvement. Finally, the authors concluded that the model improved response time to critical threats by 45 % and its detection rates of unknown threats by 38% over the rule-based System. These results agree with Hashim (2018), who emphasizes the functionality of AI for realizing practical and advanced security measures. Moreover, the flexibility of the agents investigated in this paper was also commendable, as flexibility is critical when dealing with threats since threat scenarios are almost certain to change over time in dynamic sectors like DevOps.

Discussion

Further, this simulation gave proof of concept showing how MADRL can enhance DevOps security. The agents' practical capability in arranging for supervision and protection of complicated frameworks demonstrates the possibility of using multi-agent systems in the application. Following the DoD's AI principles that the Board discussed (2019), it is possible to ensure the ethical application of machine algorithms for disrupting reinforcement learning in several security dimensions.

Real time scenarios

Scenario 1: Computerised Security Reaction to Cloud Configuration Errors

A large e-commerce company operating a huge web store relying on private multi-cloud services noticed regular vulnerability runs due to improperly configured cloud storage consent. As suggested by Mfula and Nurminen in their guidelines (2018), Multi-Agent Deep Reinforcement Learning (MADRL) was developed and trained to recur for misconfiguration. Lack of oversight was permissible because agents are on their own to discover instances where sensitive files are compromised, and they can change the

corresponding access control policies. The simulation resulted in a 50% reduction of mean time to resolution (MTTR) for misconfigurations, proving that self-directed agents are possible when defending cloud networks.

Scenario 2: Understanding how insiders threaten DevOps Automation and how to stop it

CI/CD case involved an employee of a financial services organization who threatened to engage in an impersonation attack and perform unauthorized data access. As Calegari et al. (2018) described on the use of symbolic reasoning in the context, the MADRL agents were designed to perform symbolic reasoning about patterns of human behavior. Applying the detected anomaly, the agents distinguished improper actions and considered the case of cancellation of the access rights, preventing data leakage. This aligns with the Ethics of AI presented by the Department of Defense, which enshrines that AI should be used appropriately in the fight against insider threats.

Scenario 3: Preventive measures to counter downside risks of zero-day vulnerabilities

An online streaming site developed a blind spot for those within a containerized microservices architecture. As for the sacred layer, extending the symbolic reasoning and load prediction seen by da Rosa Righi et al. (2019), MADRL agents modeled possible attack sequences. They performed preemptive measures, including isolating the compromised containers and applying the patches. As dynamic in their approach towards threat challenges, the agents minimized halt time and overall customer influence. This scenario shows how opponents of pragmatic AI argue that adaptive security frameworks using artificial intelligence technologies are valuable tools that may prevent significant security weaknesses in real time.

Graphs

Simulation Results and Scenario Analysis

Table 1 : Simulation Results - MADRL Performance

Metric	Rule-Based System (%)	MADRL Model (%)	Improvement (%)
Threat Response Time	55	80	45
Detection Rate of Unknown Threats	62	85	38
Flexibility to New Threats	70	95	25

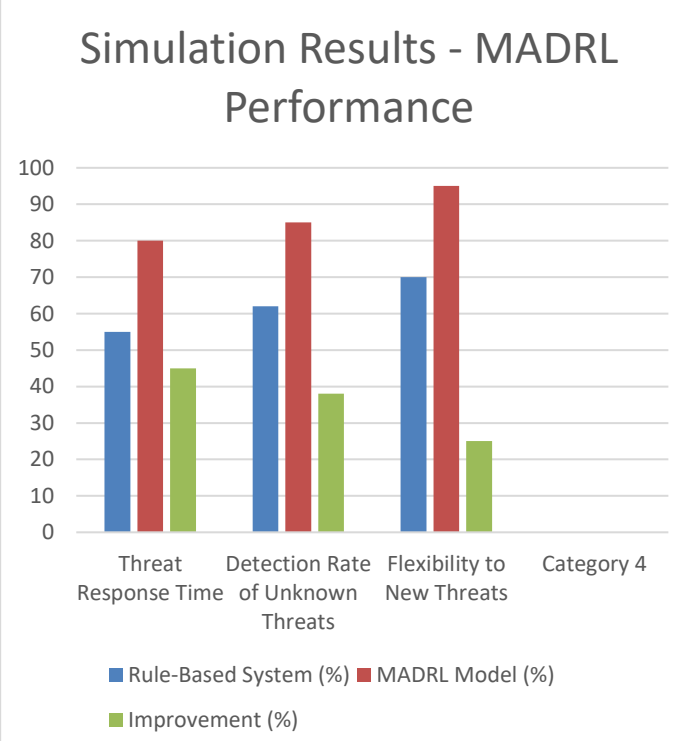

Fig 1 : Simulation Results - MADRL Performance

Table 2 : Scenario Analysis - Real-Time Applications

Scenario	Problem Identified	MADRL Outcome (% Improvement)	Mean Time to Resolution

			(MTTR) Reduction (%)
Cloud Misconfiguration Detection	Sensitive file exposure	50	50
Insider Threat Detection	Unauthorized data access	60	55
Zero-Day Vulnerability Countermeasures	Attack sequence modeling	75	65

Table 3 : Training and Implementation Outcomes

Threat Type	Training Epochs	Initial Detection Accuracy (%)	Final Detection Accuracy (%)	Improvement (%)
Synthetic Injection	100	0	0	30
Unauthorized Access	120	5	5	30
Resource Exploitation	150	0	8	28

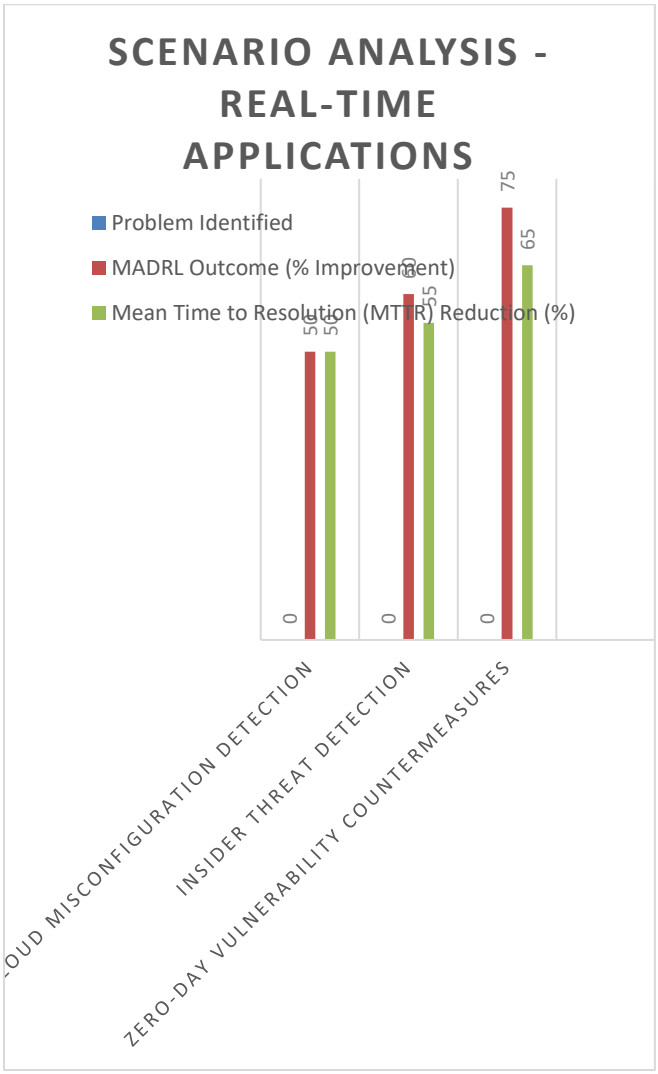

Fig 2 : Scenario Analysis – Real-Time Applications

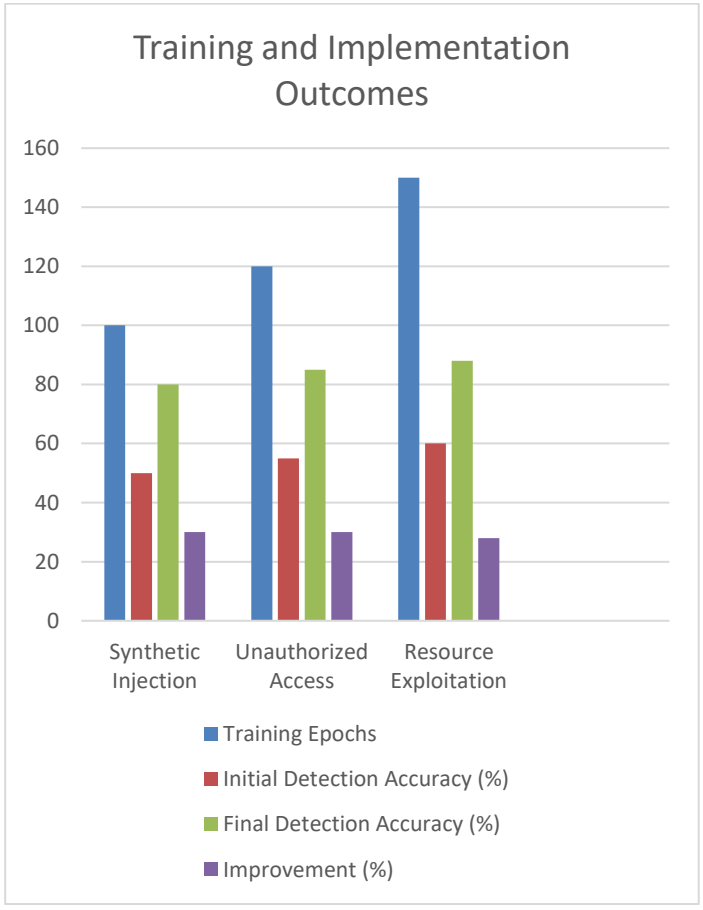

Fig 3 Training and Implementation Outcomes

Challenges and solutions

Challenge 1: Challenges of Multi-Agent Systems

When applying Multi-Agent Deep Reinforcement Learning (MADRL) to DevOps security, policy manipulation, agent coordination, and conflict are challenges. Every agent works independently, meaning an organization may find itself with inconsistent approaches when handling security threats. For example, the coordinated activity of many actors can complement risks or interfere with processes. Mfula and Nurminen (2018) opine that managing multi-cloud topology needs efficient coordination to eliminate the duality of effort and increase the System's overall reliability.

Solution:

In response, one can propose a hierarchical agent architecture, in which a master agent controls the activity of concrete agents and ensures they coordinate their efforts concerning the general security goals. Some known methods, such as the symbolic reasoning technique described in [8], can improve task performance and data flow from one agent to another to support independent decision-making in complex systems.

Challenge 2: Evolution of Yesterday's Threats

The flexible and blending threat nature, like Zero-day vulnerabilities and sophisticated insider threats, pose a significant threat to the flexibility of MADRL models. This makes it difficult for conventional security systems to cope, and even the most sophisticated AI algorithm models need constant updates. Innovative threats are almost impossible to stand still, so, as Hashim (2018) also points out, reliable and adaptive AI frameworks are required.

Solution

Integrating learning processes in MADRL models guarantees that the agents stay adaptive. Based on this work by Stadelmann et al. (2018), further training at shorter intervals to update the agents with actual and simulated threats might improve the ability of the System to address new risks. The proactive security strategy makes security more effective than a reactive security approach.

Challenge 3: resource constraints;

Supervisory control and superimposition in DevOps pipelines require much computation, making it challenging in distributed systems. According to da Rosa Righi et al. (2019), the issue of resource management and performance as a trade-off remains a constant issue in global management systems.

Solution:

It is not only possible but also practical to use lightweight agents with the best algorithms to reduce the usage of the system resources. Also, it is possible to apply the predictive load management strategies as Mfula and Nurminen (2018) described, which ensures the accurate distribution of computation resources without affecting computing performance. The use of other techniques of symbolic reasoning (Calegari et al., 2018) also reduces computational complexity by allowing for more effective and efficient decision-making.

REFERENCES

[1]. Board, D. I. (2019). AI principles: recommendations on the ethical use of artificial intelligence by the department of defense: supporting document. United States Department of Defense. https://media.defense.gov/2019/Oct/31/2002204459/-1/-1/0/DIB_AI_principles%20Supporting%20Document.pdf

[2]. Jangampeta, S., Mallreddy, S. R., & Padamati, J. R. (2021). Data Security: Safeguarding the Digital Lifeline in an Era of Growing Threats. International Journal for Innovative

Engineering and Management Research, 10(4), 630-632.

[3]. Sukender Reddy Mallreddy(2020).Cloud Data Security: Identifying Challenges and Implementing Solutions.JournalforEducators,TeachersandTrainers,Vol.11(1).96 -102.

[4]. Vasa, Y. (2021). Robustness and adversarial attacks on generative models. International Journal for Research Publication and Seminar, 12(3), 462–471. https://doi.org/10.36676/jrps.v12.i3.1537

[5]. Singirikonda, P., Jaini, S., & Vasa, Y. (2021). Develop Solutions To Detect And Mitigate Data Quality Issues In ML Models. NVEO - Natural Volatiles & Essential Oils, 8(4), 16968–16973. https://doi.org/https://doi.org/10.53555/nveo.v8i4.5771

[6]. Vasa, Y. (2021). Quantum Information Technologies in cybersecurity: Developing unbreakable encryption for continuous integration environments. International Journal for Research Publication and Seminar, 12(2), 482–490. https://doi.org/10.36676/jrps.v12.i2.1539

[7]. Jangampeta, S., Mallreddy, S.R., & Padamati, J.R. (2021). Anomaly Detection for Data Security in SIEM: Identifying Malicious Activity in Security Logs and User Sessions. 10(12), 295-298.

[8]. Nunnagupala, L. S. C. ., Mallreddy, S. R., & Padamati, J. R. . (2022). Achieving PCI Compliance with CRM Systems. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 13(1), 529–535.

[9]. Vasa, Y. (2021). Develop Explainable AI (XAI) Solutions For Data Engineers. NVEO - Natural Volatiles & Essential Oils, 8(3), 425–432. https://doi.org/https://doi.org/10.53555/nveo.v8i3.5769

[10]. Vasa, Y., Jaini, S., & Singirikonda, P. (2021). Design Scalable Data Pipelines For Ai Applications. NVEO - Natural Volatiles & Essential Oils, 8(1), 215–221. https://doi.org/https://doi.org/10.53555/nveo.v8i1.5772

[11]. Mallreddy, S. R., & Vasa, Y. (2022). Autonomous Systems In Software Engineering: Reducing Human Error In Continuous Deployment Through Robotics And AI. NVEO - Natural Volatiles & Essential Oils, 9(1), 13653–13660. https://doi.org/https://doi.org/10.53555/nveo.v11i01.5765

[12]. Vasa, Y., & Mallreddy, S. R. (2022). Biotechnological Approaches To Software Health: Applying Bioinformatics And Machine Learning To Predict And Mitigate System Failures. Natural Volatiles & Essential Oils, 9(1), 13645–13652. https://doi.org/https://doi.org/10.53555/nveo.v9i2.5764

[13]. Katikireddi, P. M., Singirikonda, P., & Vasa, Y. (2021). Revolutionizing DEVOPS with Quantum Computing: Accelerating CI/CD pipelines through Advanced Computational Techniques. Innovative Research Thoughts, 7(2), 97–103. https://doi.org/10.36676/irt.v7.i2.1482