Enhancing Kubernetes Security with Feedback-Driven Machine Learning Models

Ghadeer Darwesh

ITMO University, Saint Petersburg, Russia

Abstract. Kubernetes has become the cornerstone of container orchestration in modern cloud computing, offering unmatched scalability and flexibility. However, its growing adoption has introduced critical security challenges, particularly in mitigating Denial-of-Service (DoS) attacks. This study presents an innovative seven-layer framework to enhance Kubernetes security through real-time anomaly detection and feedback-driven machine learning models. The framework integrates two core components: a Feedback Application that captures user input to improve detection precision and a Model Agent for real-time data collection, anomaly detection, and adaptive model retraining. By combining real-time metrics with user feedback, the system dynamically evolves to address emerging threats, ensuring robust protection for Kubernetes environments. Experimental results demonstrate the framework's effectiveness in achieving high anomaly detection accuracy, reducing false positives, and maintaining adaptability in dynamic, cloud-native infrastructures.

Keywords: *kubernetes security, ML models, Feedback-driven leanring, Real-time monitoring, DoS attacks.*

DOI: 10.14357/20790279250108 **EDN: SYYIFM**

Introduction

Kubernetes has revolutionized the deployment and management of containerized applications, providing organizations with unparalleled flexibility and scalability. As the backbone of modern cloud-native architectures, it supports a wide range of applications, from microservices to large-scale enterprise platforms. However, the distributed and dynamic nature of Kubernetes introduces unique security challenges, particularly in detecting and mitigating sophisticated threats such as Denial-of-Service (DoS) attacks [1].

Traditional security mechanisms, which rely heavily on static rules and signature-based detection, have proven insufficient to address the complexities of Kubernetes clusters. These clusters operate in highly dynamic environments where workloads, configurations, and user interactions are constantly evolving. The inability of conventional systems to adapt to such changes leaves Kubernetes environments vulnerable to service disruptions, data breaches, and system instability [1].

To address these limitations, researchers have increasingly turned to machine learning (ML) as a promising solution for real-time anomaly detection. ML models can analyze large datasets to identify patterns indicative of potential threats, providing a proactive approach to securing cloud-native environments. However, existing solutions often lack adaptability, relying on pre-trained models or static rules that fail to account for evolving threat landscapes [2].

This study introduces a novel security framework that combines real-time anomaly detection with feedback-driven machine learning. The proposed framework leverages a Feedback Application and a Model Agent to continuously refine detection capabilities based on user input and real-time metrics. By integrating these components, the system dynamically adapts to emerging threats, providing a robust and adaptive security solution for Kubernetes environments. The following sections outline the framework's design, implementation, and evaluation, highlighting its potential to transform Kubernetes security practices.

1. Background and Related Work

Container security has emerged as a critical area of focus for organizations leveraging Kubernetes to orchestrate their cloud-native applications. The inherent flexibility and scalability of Kubernetes make it highly effective for containerized workloads but also introduce significant security vulnerabilities. This section provides an overview of existing tools, method-

ologies, and gaps in Kubernetes security, setting the stage for the proposed framework.

Several tools have been developed to enhance container security, such as AppArmor, a Linux kernel security module that enforces mandatory access control policies. AppArmor enables administrators to restrict process capabilities using custom security profiles, offering a baseline defense against unauthorized access [4]. Other works, such as Medel et al. [5], explored formal modeling for resource management within Kubernetes clusters, underscoring the importance of robust security measures in dynamic environments.

Monitoring solutions like Prometheus and Grafana are widely adopted in Kubernetes environments. Prometheus excels in scraping and storing time-series metrics, while Grafana is renowned for its powerful data visualization capabilities [6, 7]. Despite their effectiveness, these tools typically rely on static rules and thresholds, which often fail to detect sophisticated or evolving attacks. This limitation highlights the need for integrating machine learning models to enable real-time anomaly detection and adaptive learning [8].

The application of machine learning (ML) techniques in anomaly detection has been explored extensively in recent years. ML approaches such as random forests, support vector machines (SVM), and neural networks have demonstrated success in detecting anomalies across network traffic, system logs, and application metrics [9]. For instance, KubAnomaly, a neural network-based system, collects container events using tools like Sysdig and Falco to monitor Kubernetes environments. However, it focuses primarily on log-based events and lacks integration with lightweight metrics collection systems like Prometheus [10].

Recent studies have also addressed anomaly detection in containerized environments through real-time performance analysis. Chang et al. [11] developed a Kubernetes-based monitoring platform to dynamically provision cloud resources, emphasizing the critical role of performance metrics in identifying threats. However, these methods often rely on pretrained models or rigid rule sets, which lack the adaptability to handle new or emerging attack vectors.

Despite these advancements, existing solutions still present several limitations. Most frameworks fail to incorporate user feedback to improve model accuracy, relying instead on static detection mechanisms. Additionally, the complexity and dynamic nature of Kubernetes clusters pose challenges in effectively identifying anomalies across diverse workloads and configurations. This research addresses these gaps by proposing a feedback-driven anomaly detection framework that adapts to new threats through continuous learning. By combining real-time metrics with user feedback, the framework offers a more robust and adaptive approach to securing Kubernetes environments.

2. System Architecture and Methodology

The proposed framework introduces a novel approach to Kubernetes security by integrating real-time anomaly detection with a feedback-driven machine learning model. The framework consists of two primary components: the Feedback Application and the Model Agent. Together, these components form a robust system capable of dynamically adapting to emerging threats in Kubernetes environments. Figure 1 illustrates the overall architecture of our proposed monitoring and detection system.

2.1. Feedback Application

The Feedback Application provides an intuitive interface for administrators and security personnel to contribute valuable insights into detected anomalies. Its primary goal is to enhance detection accuracy through continuous user interaction and feedback. Key features include:

- **User-Friendly Interface**: Simplifies feedback submission with an intuitive web form.
- **Precise Timestamp Input:** Ensures accurate identification of anomaly occurrences.
- **Comprehensive Feedback Options:** Allows users to classify anomalies as true positives, false positives, true negatives, or false negatives, along with additional contextual information.

Built using Flask, a lightweight Python web framework [12], the application follows a streamlined workflow:

1. **Feedback Submission:** Users provide feedback via the web form, including timestamps, anomaly classifications, and other details (Figure 2).

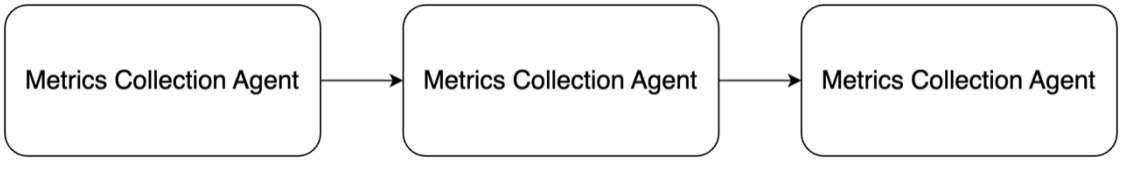

Fig. 1. The architecture of the proposed monitoring and detection system

Fig. 2. The feedback application interface

2. **Data Processing:** The application processes the submitted data and forwards it to the Model Agent.
3. **Model Enhancement:** The Model Agent stores and uses this feedback for retraining the anomaly detection model.

This workflow ensures that the system rapidly adapts to new threats while continuously refining its detection capabilities.

2.2. Model Agent

The Model Agent serves as the core of the framework, responsible for real-time anomaly detection and adaptive learning. It integrates seamlessly with Kubernetes to collect metrics, detect anomalies, and generate alerts. Key functionalities include:

- **Real-Time Detection:** Utilizes pre-trained machine learning models for rapid anomaly identification.
- **Adaptive Learning:** Continuously retrains models based on user feedback.
- **Metrics Integration:** Scrapes data directly from Kubernetes nodes and applications using Prometheus.

The Model Agent workflow is divided into four critical stages (Figure 3):

1. **Metrics Collection:** Gathers performance data from Kubernetes clusters, such as CPU usage, memory utilization, and network traffic.
2. **Anomaly Detection:** Pre-processed metrics are analyzed using machine learning models to detect anomalies in real time.
3. **Alert Generation:** Alerts are triggered via Prometheus Alertmanager, notifying administrators of detected anomalies.
4. **Feedback Integration:** User feedback from the Feedback Application is incorporated to retrain and refine the ML models.

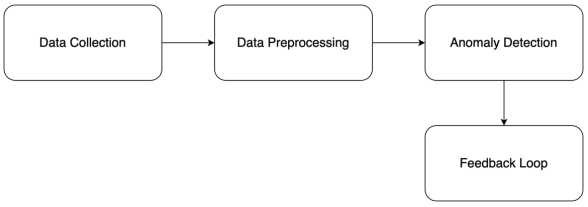

Fig. 3. The workflow of real-time anomaly detection, from data collection to feedback integration

This iterative process allows the framework to evolve continuously, maintaining its relevance against new and emerging threats.

2.3. Metrics Collection, Anomaly Detection, and Alert Generation

The Model Agent collects metrics at both node and application levels. Table 1 outlines the key metrics monitored:

Collected metrics are preprocessed and analyzed in real-time by the ML models. Upon detecting anomalies, the Model Agent generates detailed alerts containing:

- Specific anomaly details.
- Relevant metrics that triggered the alert.
- Precise timestamps of detection.

These alerts are forwarded to Prometheus Alertmanager, which initiates response protocols to mitigate potential threats.

2.4. Feedback Integration and Model Retraining

User feedback is a cornerstone of the framework's adaptability. The feedback includes detailed classifications (true positives, false positives, etc.) that are instrumental in retraining the ML models. The retraining process involves:

1. **Incorporating Feedback:** Adding user-provided data to the training dataset.

Tab. 1

Key metrics used for monitoring

Metric	Description
CPU Usage	Measures the CPU utilization of the node and applications.
Memory Usage	Measures the memory consumption of the node and applications.
Network Traffic	Monitors the amount of data being transmitted and received.
Disk I/O	Tracks the input/output operations on the disk.
HTTP Requests	Counts the number of HTTP requests handled by the applications.

2. **Model Refinement:** Updating the ML model to improve accuracy and reduce false positives.
3. **Deployment:** Deploying the retrained model back into the Model Agent for real-time detection.

This feedback loop ensures the framework remains resilient and effective against evolving security threats in Kubernetes environments.

2.5. Evaluation Metrics

To evaluate the effectiveness of the proposed framework, several metrics were used to measure its accuracy, adaptability, and overall performance:

1. **Detection Accuracy:** The percentage of correctly detected anomalies compared to the total number of anomalies identified.
2. **False Positive Rate:** The proportion of false alarms generated by the system relative to the total number of detections.
3. **Retraining Impact:** The degree of improvement in model performance after incorporating user feedback and retraining the model.

These metrics provide a comprehensive understanding of the framework's ability to detect threats and adapt over time.

2.6. Experimental Setup and Simulated Attacks

The framework was implemented in a controlled Kubernetes environment to validate its effectiveness. The experimental setup consisted of the following components:

- **Kubernetes Cluster:** Multiple nodes running containerized applications, simulating a dynamic cloud-native infrastructure.
- **Prometheus and Alertmanager:** Used for metrics collection and alert generation.
- **Feedback Application:** Enabled administrators to provide detailed feedback on detected anomalies.
- **Simulated DoS Attacks:** A range of attack scenarios were created to test the framework's detection and adaptation capabilities.

To thoroughly evaluate the framework, the following types of Denial-of-Service (DoS) attacks were simulated:

1. **Network-Based Attacks:** Targeted the network layer to congest traffic and deny service access to legitimate users.
2. **Application-Based Attacks:** Generated abnormal CPU and memory utilization, overloading system resources.
3. **Mixed Attacks:** Combined network and application-level attack vectors to mimic sophisticated real-world scenarios.

These simulations allowed for comprehensive testing of the framework's ability to identify and respond to a variety of security threats.

3. Result and Discussion

The experiments conducted in a simulated Kubernetes environment demonstrated the effectiveness of the proposed framework in addressing key security challenges. By integrating real-time anomaly detection with feedback-driven machine learning, the framework achieved high accuracy and adaptability in detecting and mitigating threats (Fig. 4).

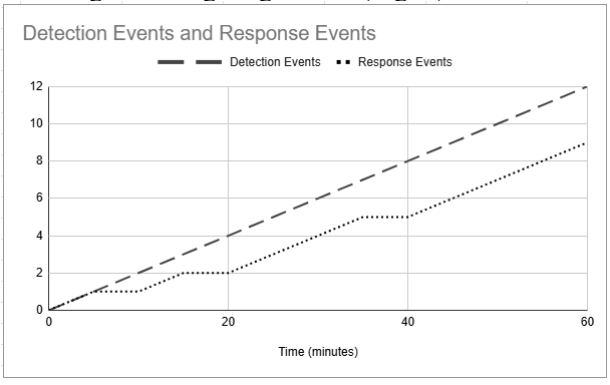

Fig. 4. The timeline of real-time detection and response, showcasing the framework's effectiveness

Key Findings:

1. **High Detection Accuracy:** The machine learning models achieved a significant improvement in anomaly detection after incorporating user feedback. The detection accuracy increased from an initial 85% to 95% post-retraining, showcasing the

framework's ability to adapt to new and emerging threats.

2. **Reduced False Positives:** The false positive rate decreased from 10% to 3%, a critical improvement that minimizes unnecessary alerts and enhances the system's reliability.
3. **Adaptive Learning:** User feedback was instrumental in refining the ML models, resulting in a continuous feedback loop that improved detection accuracy and reduced the occurrence of false negatives. Each retraining cycle incorporated new insights, ensuring the system remained robust against evolving attack patterns.

Impact of Feedback on Model Performance

The feedback mechanism played a pivotal role in enhancing the framework's performance. User-provided feedback allowed the system to:

- Distinguish between true and false anomalies more effectively.
- Reduce the rate of false alarms by learning from contextual data.
- Adapt to the peculiarities of specific Kubernetes environments, ensuring tailored security measures.

For example, feedback on false positives helped refine the model's threshold settings, while data on previously undetected threats enabled the system to adjust its anomaly detection parameters. This iterative improvement process ensured that the framework stayed relevant and efficient in dynamic cloud-native infrastructures.

Performance Metrics

The framework's performance metrics highlight its operational success:

- Initial Detection Accuracy: 85%
- Post-Retraining Detection Accuracy: 95%
- False Positive Rate: Reduced from 10% to 3%
- Retraining Interval: Weekly

These metrics validate the effectiveness of the proposed system in maintaining high accuracy while adapting to the ever-changing nature of Kubernetes environments.

Comparison with Existing Solutions

Unlike traditional anomaly detection methods that rely on static rules or pre-trained models, the proposed framework:

- Incorporates real-time feedback to dynamically adapt to new threats.
- Utilizes lightweight metrics collection tools, such as Prometheus, to ensure seamless integration with Kubernetes environments.
- Reduces false positives more effectively, providing more actionable insights for administrators.
- These advancements position the framework as a robust and scalable solution for Kubernetes security.

Limitations and Areas for Improvement

While the proposed framework demonstrates significant advantages, some limitations remain:

- Dependency on Feedback Quality: The system's performance heavily relies on the accuracy and timeliness of user feedback. Inconsistent or delayed feedback can impact retraining effectiveness.
- Resource Intensity: Retraining machine learning models at frequent intervals can be resource-intensive, potentially affecting system performance in high-load environments.
- Scalability Challenges: As Kubernetes clusters grow in size and complexity, the framework may require additional optimization to maintain its performance across distributed nodes.

These limitations present opportunities for further research and refinement, as outlined in the future work section.

Conclusion

This research introduces a novel framework to enhance Kubernetes security through real-time anomaly detection and feedback-driven machine learning. By integrating a Feedback Application and a Model Agent, the framework achieves high detection accuracy, reduces false positives, and dynamically adapts to evolving threats. Unlike traditional static approaches, the proposed system incorporates user feedback to refine its machine learning models, ensuring continuous improvement in detecting and mitigating security anomalies. The experimental results validate the framework's effectiveness in addressing Kubernetes-specific security challenges. Detection accuracy improved from 85% to 95% following model retraining, while the false positive rate was significantly reduced from 10% to 3%. These results highlight the potential of combining real-time monitoring with adaptive learning to safeguard cloud-native environments. The proposed framework offers a practical and scalable solution for Kubernetes security, with real-world applicability demonstrated through its deployment in a simulated environment. By leveraging lightweight tools like Prometheus and integrating user feedback, the system provides robust, actionable insights to administrators. This research lays the groundwork for future advancements in securing dynamic and distributed Kubernetes clusters.

References

1. *Darwesh G., Hammoud J. and Vorobeva A.* "Security in kubernetes: Best practices and security analysis," vol. 2, pp. 63–69, 06 2022.
2. *Shah J. & Dubaria D.* (2019, January). Building modern clouds: using docker, kubernetes & Google cloud platform. In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0184-0189). IEEE.
3. *Takahashi K., Aida K., Tanjo T. & Sun J.* (2018, January). A portable load balancer for kubernetes cluster. In Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region (pp. 222-231).
4. *Sultan S., Ahmad I. & Dimitriou T.* (2019). Container security: Issues, challenges, and the road ahead. IEEE access, 7, 52976-52996.
5. *Medel V., Rana O., Bañares J.Á.* & Arronategui U. (2016, December). Modelling performance & resource management in kubernetes. In Proceedings of the 9th International Conference on Utility and Cloud Computing (pp. 257-262).
6. Prometheus. (n.d.). Getting started | Prometheus. Retrieved August 1, 2024, from https://prometheus.io/docs/prometheus/latest/getting_started/
7. Technical documentation | Grafana Labs. (n.d.). Grafana Labs. Retrieved August 1, 2024, from https://grafana.com/docs/
8. *Darwesh G., Hammoud J. & Vorobeva A.A.* (2023). A novel approach to feature collection for anomaly detection in Kubernetes environment and agent for metrics collection from Kubernetes nodes. Научно-технический вестник информационных технологий, механики и оптики, 23(3), 538-546.
9. *Cao C., Blaise A., Verwer S. & Rebecchi F.* (2022, August). Learning state machines to monitor and detect anomalies on a kubernetes cluster. In Proceedings of the 17th International Conference on Availability, Reliability and Security (pp. 1-9).
10. *Tien C.W., Huang T.Y., Tien C.W., Huang T.C. & Kuo S.Y.* (2019). KubAnomaly: Anomaly detection for the Docker orchestration platform with neural network approaches. Engineering reports, 1(5), e12080.
11. *Chang C.C., Yang S.R., Yeh E.H., Lin P. & Jeng J.Y.* (2017, December). A kubernetes-based monitoring platform for dynamic cloud resource provisioning. In GLOBECOM 2017-2017 IEEE Global Communications Conference (pp. 1-6). IEEE.
12. Welcome to Flask – Flask Documentation (3.0.X). (n.d.). Retrieved August 1, 2024, from https://flask.palletsprojects.com/en/3.0.x/
13. Prometheus. (n.d.-a). AlertManager | Prometheus. Retrieved August 1, 2024, from https://prometheus.io/docs/alerting/latest/alertmanager/

Ghadeer Darwesh. ITMO University, Saint Petersburg, Russian Federation, PhD Student, https://orcid.org/0000-0003-1116-9410, ghadeerdarwesh32@gmail.com

Улучшение безопасности Kubernetes с использованием моделей машинного обучения с обратной связью

Гадир Дарвиш
Университет ИТМО, Санкт-Петербург, Россия

Аннотация. Kubernetes стал основой оркестрации контейнеров в современной облачной среде, обеспечивая непревзойденную масштабируемость и гибкость. Однако его растущая популярность привела к появлению серьезных проблем с безопасностью, особенно в предотвращении атак типа «отказ в обслуживании» (DoS). В данном исследовании представлен инновационный семислойный фреймворк для улучшения безопасности Kubernetes за счет использования моделей машинного обучения для обнаружения аномалий в реальном времени и с обратной связью. Фреймворк включает два ключевых компонента: приложение для обратной связи, которое фиксирует пользовательский ввод для повышения точности обнаружения, и модельный агент, отвечающий за сбор данных в реальном времени, обнаружение аномалий и адаптивное переобучение моделей. Объединяя метрики в реальном времени с пользовательской обратной связью, система динамически адаптируется к возникающим угрозам, обеспечивая надежную защиту Kubernetes. Экспериментальные результаты демонстрируют эффективность фреймворка в достижении высокой точности обнаружения аномалий, снижении числа ложных срабатываний и поддержании адаптивности в динамичной облачной инфраструктуре.

Ключевые слова: безопасность *Kubernetes, модели машинного обучения, обучение с обратной связью, мониторинг в реальном времени, атаки DoS.*
DOI: 10.14357/20790279250108 **EDN: SYYIFM**

Литература

1. *Darwesh G., Hammoud J., Vorobeva A.A.* Безопасность в Kubernetes: лучшие практики и анализ безопасности // Научно-технический вестник информационных технологий, механики и оптики. 2022. Т. 2. С. 63–69.
2. *Shah J., Dubaria D.* Построение современных облаков: использование Docker, Kubernetes и Google Cloud Platform // Материалы IEEE 9-й ежегодной конференции по вычислениям и коммуникациям (CCWC). 2019. С. 0184–0189.
3. *Takahashi K., Aida K., Tanjo T., Sun J.* Переносимый балансировщик нагрузки для кластера Kubernetes // Материалы Международной конференции по высокопроизводительным вычислениям в Азиатско-Тихоокеанском регионе. 2018. С. 222–231.
4. *Sultan S., Ahmad I., Dimitriou T.* Безопасность контейнеров: проблемы, вызовы и пути решения // IEEE Access. 2019. Т. 7. С. 52976–52996.
5. *Medel V., Rana O., Bañares J.Á., Arronategui U.* Моделирование производительности и управления ресурсами в Kubernetes // Материалы 9-й Международной конференции по использованию и облачным вычислениям. 2016. С. 257–262.
6. Prometheus. Основы работы | Prometheus. Дата обращения: 1 августа 2024 г. URL: https://prometheus.io/docs/prometheus/latest/getting_started/
7. Technical documentation | Grafana Labs. Дата обращения: 1 августа 2024 г. URL: https://grafana.com/docs/
8. *Darwesh G., Hammoud J., Vorobeva A.A.* Новый подход к сбору признаков для обнаружения аномалий в Kubernetes и агент для сбора метрик с узлов Kubernetes // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 3. С. 538–546.
9. *Cao C., Blaise A., Verwer S., Rebecchi F.* Изучение автоматов состояний для мониторинга и обнаружения аномалий в кластере Kubernetes // Материалы 17-й Международной конференции по доступности, надежности и безопасности. 2022. С. 1–9.
10. *Tien C.W., Huang T.Y., Tien C.W., Huang T.C., Kuo S.Y.* KubAnomaly: обнаружение аномалий для платформы оркестрации Docker с использованием нейронных сетей // Engineering Reports. 2019. Т. 1, № 5. С. e12080.
11. *Chang C.C., Yang S.R., Yeh E.H., Lin P., Jeng J.Y.* Мониторинговая платформа на базе Kubernetes для динамического обеспечения облачных ресурсов // Материалы GLOBECOM 2017 - Глобальная конференция по коммуникациям IEEE. 2017. С. 1–6.
12. Welcome to Flask – Flask Documentation (3.0.X). Дата обращения: 1 августа 2024 г. URL: https://flask.palletsprojects.com/en/3.0.x/
13. Prometheus. AlertManager | Prometheus. Дата обращения: 1 августа 2024 г. URL: https://prometheus.io/docs/alerting/latest/alertmanager/

Гадир Дарвиш. Университет ИТМО, г. Санкт-Петербург, Россия. Аспирант. Область научных интересов: информационная безопасность. E-mail: ghadeerdarwesh32@gmail.com