Issue 18, part 2, 2025

https://doi.org/10.23939/sisn2025.18.2.061

УДК 004.4’2

ON SOME APPROACHES TO INTELLIGENT COUNTERACTING CYBERATTACKS WITHIN MICROSERVICE ARCHITECTURE

Oleksiy Oletsky[1], Vitalii Moholivskyi[2]

[1, 2] National University of Kyiv-Mohyla Academy,
Department of Multimedia Systems, Kyiv, Ukraine
[1] E-mail: oletsky@ukma.edu.ua, ORCID: 0000-0002-0553-5915
[2] E-mail: v.moholivskyi@ukma.edu.ua, ORCID: 0009-0001-2654-7798

An approach to counteracting cyberattacks based on state machines within a microservice architecture is suggested. It focuses on intelligent analysis of actual and possible intrusions. The approach is devised for applications with a microservice architecture deployed on the Kubernetes platform. For purposes of the study, a special dataset has been developed. We have reproduced selected common vulnerabilities and exposures reported in 2024 and collected network traffic of intrusion cyberattacks based on them. A dataset focuses on intrusion attacks targeting software systems deployed in Kubernetes. It contains not only network data captured during attacks but also scripts to reproduce each of the studied attacks, which is particularly helpful for developing and testing intrusion response systems.

Keywords – cybersecurity, cyberattack datasets, network intrusion detection, intrusion response, data mining, microservice architecture, state machines, Kubernetes.

Problem Statement

The problem of effective intrusion detection and counteraction is becoming increasingly urgent. As adversaries adopt more sophisticated attack techniques, defenders must apply intelligent, data-driven methods that analyze network and application telemetry to detect anomalies and select appropriate, timely countermeasures.

The significance of this problem can be illustrated by the following reports. According to IBM’s Cost of a Data Breach report, the global average data breach cost reached 4.88 million dollars in 2024, which is a 10 % increase over the previous year (IBM, 2024). Moreover, it could have been significantly higher without the appliance of automation and artificial intelligence in cyberthreat detection (IBM, 2024). Automated approaches for intrusion detection have become essential in defending against cyberattacks (Goldschmidt & Chudá, 2025).

Applications based on microservice architecture are getting more widespread. This architecture has clear advantages; however, if not designed rigorously enough, it is much more vulnerable than traditional architectures. The issue of supervising and coordinating microservices, which is important even in a normal situation, becomes especially crucial if a cyberattack takes place. On the other hand, if such a microservice-based application is properly designed, supervised, and coordinated, it becomes more stable and resistant to accidental or deliberate faults.

Analysis of Recent Studies and Publications

There are many approaches to detecting intrusions. Traditional ones, such as signature detection (Kwon, Kim, & Lee, 2022) and intelligence sharing (Alaeifar et al., 2024) show a lack of reliability while dealing with unknown attacks, and the problem is more urgent within microservices. So, we think that using AI-based methods

such as behavioral analysis (Palaparthy et al., 2024) and anomaly detection (Palaparthy et al., 2024; Moustafa & Slay, 2016) should be very helpful. Most of these methods are based on Data Mining and reinforcement learning (Buczak & Guven, 2016; Yin et al., 2017). Discussed approaches are compared in Table 1.

Table 2

A comparison of approaches to threat detection

Method	Known threats	Unknown threats
Signature detection	highly detectable	undetectable
Anomaly detection	detectable	highly detectable
Behavioral analysis	detectable	highly detectable
Intelligence sharing	highly detectable	undetectable
Advanced Machine Learning	detectable	highly detectable
Hybrid	highly detectable	highly detectable

The development of a cybersecurity system requires high-quality datasets of known threats to evaluate the obtained results. At the same time, cybersecurity is among the fields with the least amount of publicly available datasets. Some widely acknowledged ones are:

1. CIC-IDS-2017 (Sharafaldin et al., 2018; Engelen et al., 2021)
2. BCCC-CSE-CIC-IDS2018 (Shafi et al., 2025)
3. CIC-BCCC-NRC TabularIoTAttack-2024 (Sasi et al., 2024)
4. The UNSW-NB15 Dataset (Moustafa & Slay, 2016; Moustafa & Slay, 2015; Sarhan et al., 2021)

CIC-IDS-2017 was created by the Canadian Institute for Cybersecurity, University of New Brunswick. It is the first modern dataset of such volume. The dataset consists of realistic traffic that simulates network activity with the relevant labels. It is widely used to evaluate intrusion detection models and algorithms.

BCCC-CSE-CIC-IDS2018 and CIC-BCCC-NRC TabularIoTAttack-2024 were created by Behaviour-Centric Cybersecurity Center, York University. The BCCC-CSE-CIC-IDS2018 dataset is an enhanced version of CSE-CIC-IDS2018, which fixed a lot of issues present in CIC-IDS-2017 (Engelen et al., 2021) and went through a few cycles of improvement already.

The raw network packets in the UNSW-NB15 dataset were generated using the IXIA PerfectStorm tool at the Cyber Range Lab of UNSW Canberra, creating a combination of realistic, modern normal activities and simulated contemporary attack behaviors.

Those datasets are highly valuable for training intrusion detection systems. However, they are not quite convenient for developing and testing intrusion response systems as they do not contain specific steps required to reproduce attacks. In this context, the dataset “The Kubernetes dataset for misuse detection” (Sever & Dogan, 2023) stands out with its approach tailored specifically for microservices architectures and Kubernetes workloads. It consists of network activity data collected from a Kubernetes-based environment, containing realistic benign behaviors and labeled examples of container-based cyberattacks. However, it also does not include an automated programmable way to reproduce attacks, only a general description, which is not quite convenient for research related to intrusion response. The main problem with existing datasets is that they are weakly oriented on behavioral aspects, e.g., examples of attacks are not reproducible.

Concerning intrusion response systems, they mostly use network-level firewall rules, shutdown of compromised systems, or even only passive responses consisting of monitoring and alerting (Inayat et al., 2016; Wang & Stolfo, 2004; Stakhanova et al., 2007). Some studies focus heavily on intrusion prevention rather than response (Kaul, 2025). In (Savchenko et al., 2025), the technological and technical processes affecting the development and cybersecurity of Digital Twins in the dairy industry are investigated. The research focuses on creating a comprehensive monitoring system that enables early detection of deviations and potential threats in production, which helps maintain product quality and, at the same time, reduces cybersecurity risks. Neglecting cyberthreat detection and response in Digital Twins may lead to devastating consequences in physical systems; thus, it requires special attention. Methods not only for detecting

intrusions but those for planning reasonable responses and for intelligent counteracting are not highly developed currently. An interesting approach to adaptive cost-based intrusion response has been suggested in (Kourki Nejat & Kabiri, 2017), but it can be entrenched by more sophisticated application-aware response techniques. In this paper, we are developing an approach aiming at solving such problems.

Formulation of the Article's Objective

In (Oletsky & Moholivskyi, 2024a), we suggested an approach to supervising and coordinating microservices involving their declarative descriptions based on so-called state machines (Sipser, 2012; "XState documentation," n.d.). In particular, this approach enables applying mathematical methods for the analysis and optimization of application structure based on such a declarative description. A software demonstrative prototype implementing such an approach has been developed; it functions in Kubernetes environment. This article objectives are developing and studying a prototype of a state machine-driven intrusion detection and response system that helps to manage the growing number and complexity of emerging cyberthreats. We aim to provide solid means of supervision and coordination that are necessary both during the cyberthreat detection and the response stages.

This study is focused on exploring possibilities related to detecting possible cyberattacks and counteracting them. There is always a lack of recent cyberattack datasets that include both the necessary data and scripts to reproduce such attacks, specifically within a Kubernetes environment. So, we have created a new dataset for this investigation. Only selected recent CVEs reported in 2024 were included in this dataset; thus, these vulnerabilities are not present in any existing datasets. It makes this dataset not only useful for our research but also for the cybersecurity community, as it accommodates a constant need for datasets with recent vulnerabilities. This need is especially relevant for 2024, as according to the CVE Program, the number of reported CVE records has increased by 38 % in 2024 (Common Vulnerabilities and Exposures Program, n.d.).

Main Results

Integrating state machine-based supervision and coordination into the Intrusion Detection and Response (IDR) System for Kubernetes enhances its effectiveness in several ways. In (Oletsky & Moholivskyi, 2024b), we described the application of state machines for supervising and coordinating microservices in web applications. It discusses both orchestration and choreography approaches, introduces a prototype library based on state machines, and highlights benefits such as centralized control, declarative system descriptions, enhanced visualization, and improved monitoring of complex distributed workflows.

State machines provide a clear visual representation of the system's states and transitions, offering centralized control over the IDR system's workflows. This visualization aids in understanding and managing the system's behavior, especially during complex intrusion scenarios. By defining workflows declaratively, state machines facilitate the specification of intrusion detection and response processes. This approach simplifies the design, implementation, and modification of detection rules and response strategies within the Kubernetes environment. State machines enable real-time monitoring of the IDR system's current state, assisting in tracking ongoing processes and identifying anomalies. State machine-based coordination supports the dynamic scaling of microservices, accommodating varying loads and evolving security requirements. This flexibility is essential for adapting to new threats and integrating additional detection mechanisms as needed.

Fig. 1 shows the proposed system design. The figure pictures three Kubernetes namespaces: the arbitrary application namespace, the kube-system Kubernetes system namespace, and the namespace where the designed IDR system is deployed.

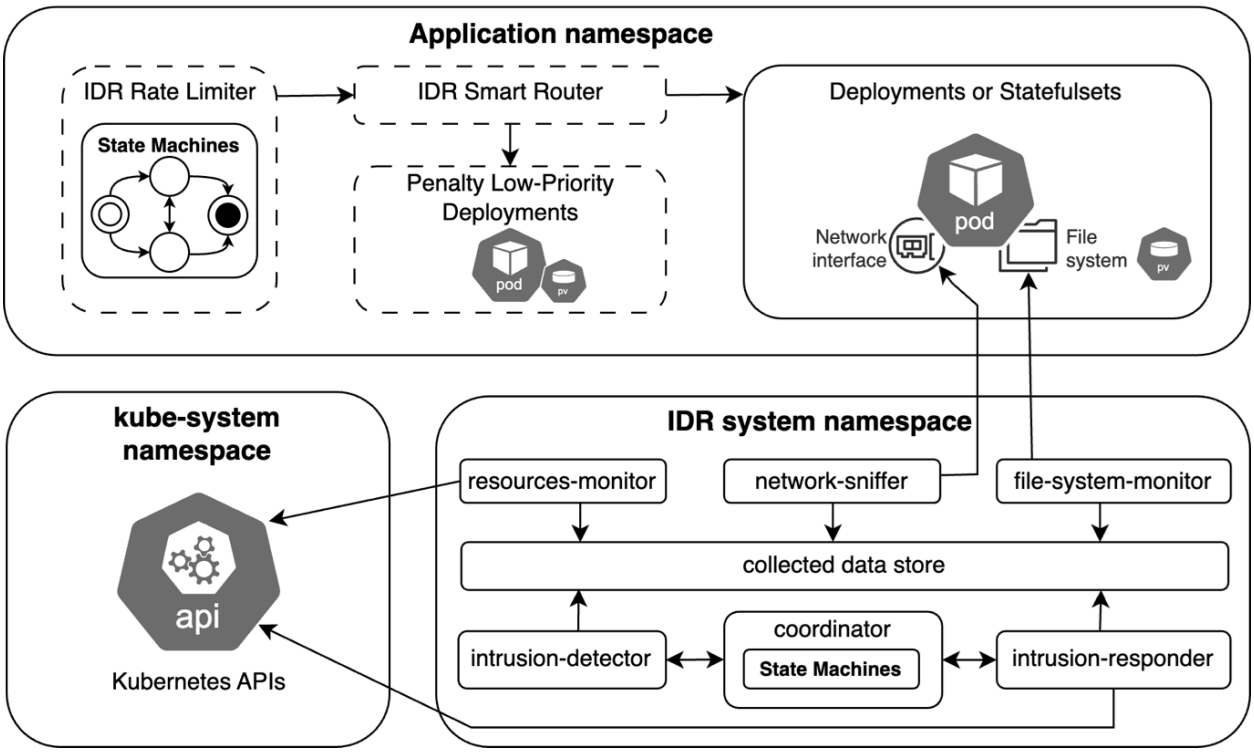

Fig. 1. The architecture of the Intrusion detection and response system for Kubernetes

Within the application namespace, standard user applications requiring protection are deployed using Kubernetes abstractions such as Deployments, StatefulSets, and others. These workloads operate within pods that interact with networking interfaces and shared or persistent file systems. To ensure security-aware traffic routing, the architecture introduces an IDR Rate Limiter and IDR Smart Router.

The IDR Rate Limiter can throttle or isolate potentially harmful traffic patterns. The logic of the rate limiter is governed by a set of formalized state machines, which define system responses under different security states.

The IDR Smart Router controls the flow of external and internal requests directed to application workloads. It dynamically manages request flow based on the system's security state. When the system detects potential threats or suspicious activity, the smart router can reconfigure traffic paths to reassign suspicious traffic to penalty low-priority deployments, a form of workload degradation that reduces privyleges or computational priority until further investigation is completed. This enables proactive containment without halting all application functionality, ensuring continued availability under partial compromise.

The IDR system namespace forms the defensive intelligence core of the architecture. It contains several monitoring components, including the resources monitor, network sniffer, and file system monitor. These components are tasked with collecting telemetry on computational resource usage, packet-level network activity, and file system interactions, respectively. These streams of data are consolidated in a unified data store, which forms the analytical base for detecting suspected intrusions.

The intrusion detector constantly analyzes collected data, applying rule-based heuristics, signature detection, and machine learning techniques to identify potential threats. Upon detecting a confirmed intrusion, the system escalates the event to the coordinator, which is powered by state machines. The coordinator determines the appropriate course of action based on predefined security workflows, ensuring a structured transition between states, for example, normal, suspicious, mitigation, and recovery. By following a declarative approach to security automation, the state machine framework allows for transparent and predictable intrusion-handling processes, reducing response time and minimizing manual intervention.

Following the coordinator's decision, the intrusion responder can execute mitigation strategies in accordance with the detected threat and the importance scores of microservices (Oletsky & Moholivskyi, 2024b). A declarative description of the system based on the state machine allows for evaluating measures of importance across microservices mathematically; one approach of such a sort based on Page Rank-like methods has been suggested in (Oletsky & Moholivskyi, 2024b). For example, if a certain Kubernetes container is compromised, the responder might isolate it from the network, revoke access permissions, or even terminate the deployment entirely to prevent further threat propagation. By integrating directly with Kubernetes APIs, the IDR system can enforce these security measures in real-time, adjusting network policies, scaling down risky deployments, or redirecting malicious traffic as needed. This proactive response mechanism helps maintain system stability while limiting the potential damage caused by cyberthreats.

Fig. 2 illustrates how a state machine defines an intrusion response workflow. It pictures an intrusion response state machine, detailing the transition processes and decision-making criteria involved when handling security incidents. Initially, the system starts in the "scheduled" state, awaiting for "STATE_MACHINE_START" event. The primary task at this stage is to initiate an alert through the "raiseAlert" process. This is represented by invoking the "raiseAlert" action, transitioning the machine to the "choosingResponseStrategy" state based on a triggered event.

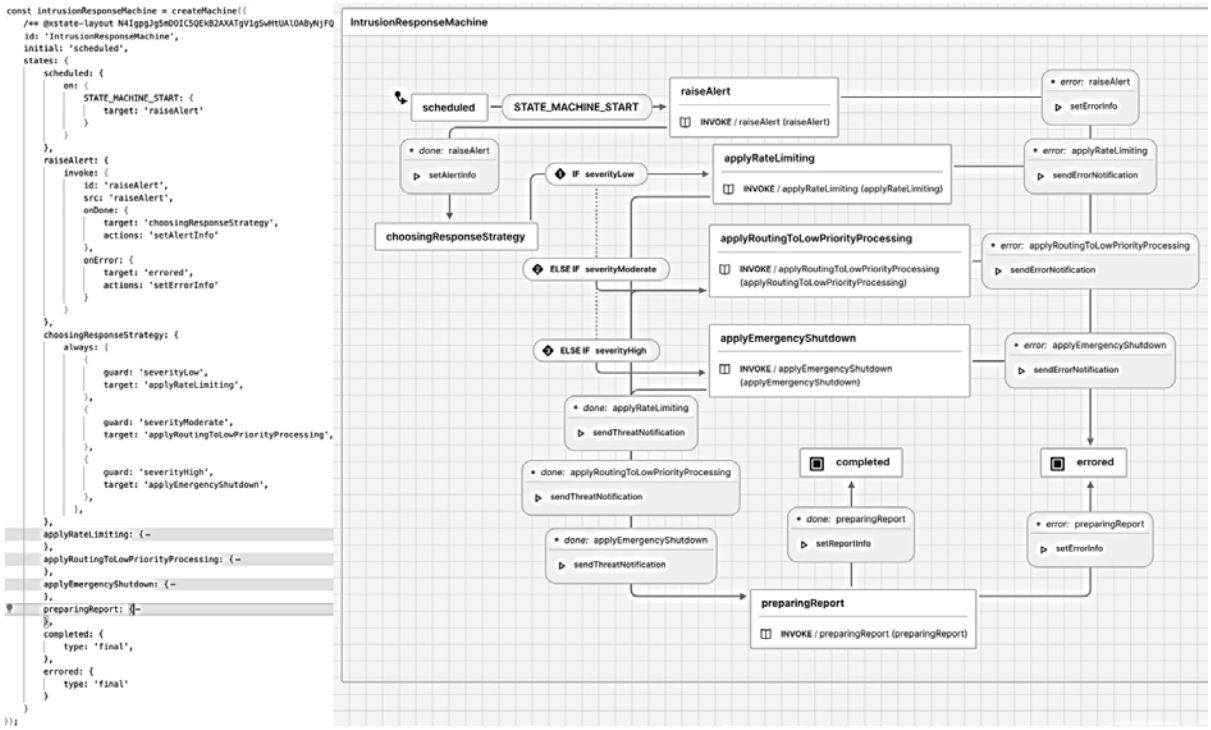

Fig. 2. The example of a state machine used for intrusion response and its visualization created using XState Visualizer

Upon entering the "choosingResponseStrategy," the state machine evaluates the severity of the intrusion. This decision is governed by three conditional paths: "severityLow," "severityModerate," and "severityHigh." If the intrusion is assessed as low severity, the system proceeds to "applyRateLimiting." For moderate severity, it initiates "applyRoutingToLowPriorityProcessing," and for high severity, it does "applyEmergencyShutdown". Each of these actions triggers subsequent states or notifications, such as "sendThreatNotification," indicating real-time communication of threats.

The final stages involve completing the response actions and transitioning to report preparation. Successful completion leads to a "completed" state where a report is prepared, whereas errors during processing lead to an "errored" state, causing error-specific notifications.

This state machine shows well-structured, easily visualizable, and automated handling of security incidents.

The proposed IDR system leverages state machines to coordinate detection, analysis, and response actions in a structured manner. The use of multi-dimensional monitoring covering resources, network activity, and filesystem ensures comprehensive security coverage. By automating enforcement and dynamically adjusting security policies through Kubernetes APIs, the system enhances resilience against intrusions while allowing legitimate workloads to continue operating smoothly. This approach provides a scalable and efficient solution for securing Kubernetes environments against a wide range of evolving threats.

To properly evaluate the suggested solution, it's required to reproduce attacks within the testing environment. For this purpose, we have created a dataset that not only includes data for training detection models but also steps to reproduce attacks in a controlled environment to develop and evaluate suggested response strategies.

A custom dataset of few selected common vulnerabilities and exposures of 2024 was collected by the authors of this paper. The repository containing received data and reproduction instructions is available on GitHub (Moholivskyi, 2025). The repository documents the creation of a dataset containing network traffic captures associated with specific Common Vulnerabilities and Exposures (CVEs). It provides resources and instructions for setting up a local Kubernetes cluster to replicate and analyze both benign and malicious traffic patterns related to these vulnerabilities.

A local deployment of Kubernetes is used as an environment for the dataset collection. The use of Kubernetes for this task serves two main purposes. It enhances the reproducibility of the research and simulates the conditions of commercial production workloads. According to the CNCF 2023 annual survey, 84 % of cloud service providers and consumers use or evaluate Kubernetes (Cloud Native Computing Foundation, 2023). Thus, we ensure that our intrusion cyberattack dataset is collected in an environment that closely matches the conditions in which our intrusion detection system will operate in a production environment. Network traffic is captured as package capture files on a pod container network interface level. These files are then processed by NTLFlowLyzer (Shafi et al., 2025) to extract network layer features for further preliminary analysis. We have used the Random Forest machine learning algorithm for the initial analysis of the collected dataset.

Despite existing datasets with sound data, having one's own dataset is very helpful during cybersecurity research. Moreover, a dataset of data related to recent exposures is highly valuable to the community. While reproducing recent vulnerabilities is complex and time-consuming, having such data to additionally evaluate new developments is always useful. Kubernetes is used as an environment to reproduce attacks and collect data. Utilizing Kubernetes for this task not only ensures that research results are reproducible but also brings us as close as possible to the conditions in which we want to detect attacks.

CVEs to reproduce for dataset collection were selected based on the following criteria:

- they should be recent, meaning reported in 2024;
- they should have a network trace as only network activity is recorded for the dataset;
- it should be possible to reproduce them in the Kubernetes.
- Based on defined criteria CVEs listed in Table 2 have been selected.

Table 2

Reproduced vulnerabilities

Vulnerability	Category	CWE	Known threats	Unknown threats	CVSS Score
CVE-2024-27983	Denial of Service	CWE-362	highly detectable	undetectable	8.2
CVE-2024-31449	Overflow, Execute code	CWE-20, CWE-121	detectable	highly detectable	7.0
CVE-2024-21538	Denial of service	CWE-1333	detectable	highly detectable	7.5
CVE-2024-21534	Execute code	CWE-94	highly detectable	undetectable	9.8

The table contains the CVE security vulnerability database identifier for each entry. Each entry can have one or many categories according to the CVE database and one or many CWE. Finally, the Common Vulnerability Scoring System (CVSS) Score is provided for each row.

CVE-2024-27983 is a Node.js HTTP/2 vulnerability that leads to the denial of service state of the web server. It is a flaw in HTTP/2 implementation in Node.js. An attacker can exploit it by sending a specific sequence of HTTP/2 frames. This triggers a race condition in the Http2Session destructor, which causes the server to crash.

CVE-2024-31449 is a Redis Lua scripting stack buffer overflow issue. This vulnerability can potentially lead to remote code execution.

CVE-2024-21538 is a regular expression denial of service (ReDoS) in the cross-spawn package. Due to broken input validation, an attacker can craft a specially designed, large string that results in high CPU usage and possible application failure.

CVE-2024-21534 is a Remote Code Execution (RCE) vulnerability in the *jsonpath-plus* package. The issue arises from insufficient input validation and the insecure default use of the *vm* module in Node.js, which allows attackers to execute arbitrary code on the affected system.

A local Kubernetes cluster was deployed using minikube to simulate enterprise network conditions. Required configurations were applied, including enabling necessary addons and deploying predefined services. A combination of benign and malicious network traffic was generated. Targeted CVEs were exploited in a controlled manner to replicate attack scenarios.

Network traffic was recorded using packet capture files. Additional flow-based representations were created to facilitate analysis using NTLFlowLyzerFlow (Shafi et al., 2025). The collected traffic was organized into structured datasets, divided into benign and malicious traffic.

The initial dataset analysis involved a structured approach to preprocessing, training, and evaluating a machine learning model for detecting malicious network activity. The dataset was first loaded and preprocessed by replacing infinite values with not a number and imputing missing values using the mean. Non-relevant columns, such as IP addresses, timestamps, and flow identifiers, were removed to ensure the model focused on meaningful network traffic features. The dataset was then split into training (60 %) and testing (40 %) subsets, followed by standardization to normalize feature values. A Random Forest model was trained with different max depths. Performance metrics such as accuracy, precision, recall, and F1-score were computed to evaluate the model's ability to detect malicious activity.

Table 3 shows the performance of a Random Forest model at different tree depths. As the collected dataset was insignificant in size, even depth 1 provides satisfactory performance. On a larger dataset, the results would not have been as positive. However, the objective is only to get an initial insight into the collected dataset.

Table 3

The performance of the Random Forest on the collected dataset

Depth	Accuracy, %	Precision, %	Recall, %	F1	False Positive	True Positive	False Negative	True Negative
1	91.15	79.37	75.37	77.32	105	404	132	2037
2	95.37	99.52	77.24	86.97	2	414	122	2140
3	95.41	99.76	77.24	87.07	1	414	122	2141
5	99.74	98.71	100	99.35	7	536	0	2135
10	99.70	98.71	99.81	99.26	7	535	1	2135

At depth 1, the model achieves an accuracy of 91.15 %, but precision (79.37 %) and recall (75.37 %) are relatively low, indicating misclassifications. The false negative count (132) shows that many malicious samples were missed. At depths 2 and 3, accuracy improves to 95.37 % and 95.41 %, and precision rises significantly to 99.52 % and 99.76 %, meaning fewer benign samples were misclassified. However, recall

remains at 77.24 %, implying that some attacks are still undetected. The false negative count (122) supports this observation. At depth 5, the model achieves near-perfect recall (100 %) with 99.74 % accuracy. There are zero false negatives, meaning all malicious samples were correctly identified. However, there are 7 false positives, meaning a few benign samples were incorrectly flagged as malicious. At depth 10, performance remains nearly the same, suggesting depth 5 is optimal for the dataset.

This dataset is tailored specifically to test our state machine-based intrusion response strategies and does not act as the standalone basis for an intrusion detection system. At the same time, despite its modest size, it is a valuable addition to other datasets as it contains recent vulnerability data.

Conclusions

An approach to counteracting cyberattacks on the base of state machines offers significant enhancements in managing complex cyberthreats within microservice architecture. Utilizing state machine-based supervision and coordination for the Kubernetes Intrusion Detection and Response (IDR) system provides a logical and visual framework that simplifies the declaration, monitoring, visualization, and support of detection and response workflows. By defining them declaratively, the system becomes more transparent and predictable. The suggested approach ensures efficient handling of intrusion scenarios and structural execution of the appropriate response strategies. State machines are also used for granular control over network traffic by dynamically adjusting throttle or isolation logic based on detected threats. The proactive containment strategy aids in preserving operational continuity even when partial system compromises occur, for example, by rerouting suspicious traffic to low-priority deployments. This form of penalty workload degradation reduces privileges and computational priority, allowing further investigation to be conducted without significant disruption. The proposed system represents a scalable and efficient solution for securing Kubernetes environments. It establishes a solid foundation for the development of advanced cybersecurity mechanisms, which can provide robust protection against a variety of potential threats.

Another valuable finding in this paper is a specialized dataset of cyberattacks. We have replicated certain common vulnerabilities and exposures reported in 2024 and gathered network traffic data during cyberattacks based on these vulnerabilities. This dataset concentrates on intrusion attacks aimed at software systems deployed in Kubernetes environments. It includes both network data captured during the attacks and scripts to automatically reproduce each of the analyzed attacks. It is especially valuable when developing and testing intrusion response systems. The use of Kubernetes to replicate network conditions ensures that the dataset accurately represents real-world scenarios, enhancing its utility for intrusion detection research. The selection process for CVEs, based on recency and reproducibility in Kubernetes, guarantees that the dataset remains relevant and applicable to modern environments. Moreover, a dataset of data related specifically to recent exposures is highly valuable to the community.

REFERENCES

Alaeifar, P., Pal, S., Jadidi, Z., Hussain, M., & Foo, E. (2024). Current approaches and future directions for cyber threat intelligence sharing: A survey. Journal of Information Security and Applications, 83, 103786. doi:10.1016/j.jisa.2024.103786

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176. doi:10.1109/comst.2015.2494502

Cloud Native Computing Foundation. (2023). CNCF annual survey 2023. Retrieved from https://www.cncf.io/reports/cncf-annual-survey-2023

Common Vulnerabilities and Exposures Program. (n.d.). CVE metrics. Retrieved from https://www.cve.org/about/Metrics

Engelen, G., Rimmer, V., & Joosen, W. (2021). Troubleshooting an intrusion detection dataset: The CICIDS2017 case study. In 2021 IEEE Security and Privacy Workshops (SPW). IEEE. doi:10.1109/spw53761.2021.00009

Goldschmidt, P., & Chudá, D. (2025). Network intrusion datasets: A survey, limitations, and recommendations. arXiv. doi:10.48550/arXiv.2502.06688

IBM. (2024). Cost of a data breach 2024. Retrieved from https://www.ibm.com/reports/data-breach

Inayat, Z., Gani, A., Anuar, N. B., Khan, M. K., & Anwar, S. (2016). Intrusion response systems: Foundations, design, and challenges. Journal of Network and Computer Applications, 62, 53–74. doi:10.1016/j.jnca.2015.12.006

Kaul, D. (2025). Blockchain-powered cyber-resilient microservices: AI-driven intrusion prevention with zero-trust policy enforcement. SSRN Electronic Journal. doi:10.2139/ssrn.5096255

Kourki Nejat, S., & Kabiri, P. (2017). An adaptive and cost-based intrusion response system. Cybernetics and Systems, 48(6–7), 495–509. doi:10.1080/01969722.2017.1319693

Kwon, H.-Y., Kim, T., & Lee, M.-K. (2022). Advanced intrusion detection combining signature-based and behavior-based detection methods. Electronics, 11(6), 867. doi:10.3390/electronics11060867

Moholivskyi, V. (2025). Selected CVE dataset 2024. GitHub. Retrieved from https://github.com/vitalii-moholivskyi/selected-cve-dataset-2024

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. In 2015 Military Communications and Information Systems Conference (MilCIS). IEEE. doi:10.1109/milcis.2015.7348942

Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal, 25(1–3), 18–31. doi:10.1080/19393555.2015.1125974

Oletsky, O., & Moholivskyi, V. (2024a). Coordination of microservices using state machines. NaUKMA Research Papers. Computer Science, National University of Kyiv-Mohyla Academy, 7, 4–10. doi:10.18523/2617-3808.2024.7.4-10

Oletsky, O., & Moholivskyi, V. (2024b, November 20–21). On supervising and coordinating microservices within web applications on the basis of state machines. *In Selected Papers of the XI International Scientific Conference "Information Technology and Implementation" (IT&I 2024)*, Kyiv, Ukraine (pp. 442–454). CEUR Workshop Proceedings. Retrieved from https://ceur-ws.org/Vol-3909/Paper_35.pdf

Palaparthy, K., Reddy, Y. M., Paul, J. V., & Raju, S. (2024). Enhancing insider threat detection through integrated behavioral, signature, and anomaly based detection methods. International Journal of Scientific Research in Engineering and Management, 8(12), 1–6. doi:10.55041/ijsrem39835

Sarhan, M., Layeghy, S., Moustafa, N., & Portmann, M. (2021). NetFlow datasets for machine learning-based network intrusion detection systems. In Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (pp. 117–135). Cham, Switzerland: Springer. doi:10.1007/978-3-030-72802-1_9

Sasi, T., Lashkari, A. H., Lu, R., Xiong, P., & Iqbal, S. (2024). An efficient self attention-based 1D-CNN-LSTM network for IoT attack detection and identification using network traffic. Journal of Information Intelligence. doi:10.1016/j.jiixd.2024.09.001

Savchenko, T., Lutska, N., Vlasenko, L., Sashnova, M., Zahorulko, A., Minenko, S., Ibaiev, E., & Tytarenko, N. (2025). Risk analysis and cybersecurity enhancement of Digital Twins in dairy production. Technology Audit and Production Reserves, 2(2(82)), 37–49. https://doi.org/10.15587/2706-5448.2025.325422

Sever, Y., & Dogan, A. H. (2023). A Kubernetes dataset for misuse detection. ITU Journal of Future and Evolving Technologies, 4(2), 383–388. doi:10.52953/fplr8631

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy. SCITEPRESS. doi:10.5220/0006639801080116

Shafi, M., Lashkari, A. H., & Roudsari, A. H. (2025). NTLFlowLyzer: Towards generating an intrusion detection dataset and intruders' behavior profiling through network and transport layers traffic analysis and pattern extraction. Computers & Security, 148, 104160. doi:10.1016/j.cose.2024.104160

Sipser, M. (2012). Introduction to the theory of computation. Boston, MA: Thomson South-Western.

Stakhanova, N., Basu, S., & Wong, J. (2007). A taxonomy of intrusion response systems. International Journal of Information and Computer Security, 1(1–2), 169–184. doi:10.1504/ijics.2007.012248

Wang, K., & Stolfo, S. J. (2004). Anomalous payload-based network intrusion detection. In E. Jonsson, A. Valdes, & M. Almgren (Eds.), Recent advances in intrusion detection (pp. 203–222). Berlin, Germany: Springer. doi:10.1007/978-3-540-30143-1_11

XState. (n.d.). XState documentation. Retrieved from https://xstate.js.org/docs/

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5, 21954–21961. doi:10.1109/access.2017.2762418

ПРО ДЕЯКІ ПІДХОДИ ДО ІНТЕЛЕКТУАЛЬНОЇ ПРОТИДІЇ КІБЕРАТАКАМ В РАМКАХ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ

Олексій Олецький[1], Віталій Моголівський[2]

[1, 2] Національний університет «Києво-Могилянська академія»,
Кафедра мультимедійних систем, Київ, Україна
[1] E-mail: oletsky@ukma.edu.ua, ORCID: 0000-0002-0553-5915
[2] E-mail: v.moholivskyi@ukma.edu.ua, ORCID: 0009-0001-2654-7798

Запропоновано підхід для протидії кібератакам у рамках мікросервісної архітектури з використанням моделей на основі машин станів. Створене рішення орієнтоване на інтелектуальний аналіз поточних та потенційних мережевих вторгнень. Метод розроблено для застосувань, що функціонують у середовищі мікросервісної архітектури, розгорнутої на платформі Kubernetes. У рамках дослідження було зібрано спеціалізований набір даних. Для цього було відтворено низку поширених вразливостей зареєстрованих у 2024 році, та зібрано відповідний мережевий трафік кібератак. Зібраний набір даних зосереджується на атаках, спрямованих проти програмних систем, розгорнутих у Kubernetes. Він містить мережеві дані, зафіксовані під час атак, та скрипти для відтворення кожної з досліджених атак, що є важливим для подальшої розробки та тестування систем виявлення й реагування на вторгнення.

Ключові слова – кібербезпека, набори даних кібератак, виявлення мережевих вторгнень, реагування на вторгнення, інтелектуальний аналіз даних, мікросервісна архітектура, машини станів, Kubernetes.