10.48047/jocaaa.2025.34.12.96

Engineering Scalable Cloud-Native Distributed Systems for Real-Time Security Telemetry Ingestion and Threat Detection

Akhil Karrothu

Software Engineer, Lynnwood, Washington
akarrothu0@gmail.com

Abstract

Multicloud and distributed infrastructures be adopted quickly has resulted in a growing avalanche of security telemetry produced by applications, networks and endpoints. Consuming, correlating, and analyzing this high-velocity data efficiently in real-time is a core problem faced by the modern day security operations. In this paper we discuss the architecture and implementation of a cloud native scalable distributed system for real-time security telemetry ingestion and threat detection. Leveraging microservices, container orchestration and event-based streaming frameworks specifically for high throughput, low latent and fault tolerant dynamic workloads. A distributed ingestion pipeline, built on message brokers and stream processors, operationalizes the telemetry data normalization and enrichment, whereas ML-based anomaly detection models in tandem serve smart threat detection. It provides horizontal scaling, dynamic resource management and fault tolerant recovery to cater for surging telemetry rates and changing attack behaviors. Experiments verify that our solution is effective at achieving high i-Theft rates with low processing overhead, while ensuring accurate and timely threat detection in distributed cloud settings. The findings indicate the promise of cloud-native approaches to consider how to augment continuous security monitoring and deliver proactive defense in large-scale distributed systems.

Keywords: Cloud-native systems, distributed architectures, security telemetry, real-time threat detection, scalable stream processing.

1. INTRODUCTION

The move to digital for businesses and the increasingly universal adoption of cloud computing has radically changed how modern applications are built, deployed and managed. Scalability, agility and high availability demand that organizations embrace cloud-native architectures, microservices and distributed systems. But this change also widened the attack surface and accelerated the amount of volume, velocity and variety of security-related telemetry coming out of cloud workloads, networks, containers, APIs and user interactions. The ability to consume and process effectively such a large

10.48047/jocaaa.2025.34.12.96

stream of security telemetry in real time has become an essential need for proactive threat detection and incident response.

Legacy security monitoring systems are built for static, on premises configurations and batch oriented log analysis. These solutions do not handle the dynamism in cloud-native systems where resources are transient, workloads scale adaptively, and traffic patterns evolve quickly. This approach has limitations and can lead to a long delay in threat detection, slow time to process, and low visibility of the distributed environments. While cyberattacks are increasingly automated and sophisticated, modern security operations centers need robust, scalable telemetry pipelines capable of processing millions of events per second with no compromise in accuracy or performance.

The recent advances in cloud-native distributed systems offer a promising basis for tackling these challenges. The likes of container orchestrators, event-driven architectures, and distributed streams processors now bring horizontal scalability, failover, and optimized resource utilization capabilities. By uncoupling telemetry producers and consumers as well as using asynchronous data streams, cloud-native systems are able to ingest high-throughput security data, all while preserving low end-to-end latency. And it results from the fact that real-time analytics and machine learning models are in these pipelines, giving insight very quickly into things like anomalous behavior, potential break-ins and brand new attack patterns.

Even with these improvements, the design and engineering of a scalable security telemetry ingestion and threat detection platform are non-trivial. Points-of-Interest in telemetry data that are solid candidates for Semaphore generation enable visual correlation of several different IoT devices. Issues to be addressed include: (i) accepting bursty and mixed types of telemetry sources; (ii) consistency and ordering of data ingested by the system; (iii) small latency overhead when processing the points-of-interest request load injected by many clients or applications; and, (iv) ensuring fault-resistant operation even under failure conditions. Further, to achieve an effective real-time protection the level of accuracy in detecting the threat must be coupled with computational efficiency, especially at Internet-scale multi-cloud settings where monetory and service latency constraints dominate.

In this paper, we tackle these challenges and introduce the design of a cloud-native distributed architecture that is tailored for real-time security telemetry ingestion and threat detection. The designed system utilises the microservices concepts, distributed messaging systems and stream processing engines to support elastic scaling and high availability. Normalizes, enriches and analyze

10.48047/jocaaa.2025.34.12.96

security telemetry in real-time for immediate security threat detection and anomalous activity. The system is architected to elastically grow as telemetry volume rises, and stay predictable in terms of performance and reliability at all times.

The rest of this paper is structured as follows. Section 2 discusses related work and compares with prior solutions for processing security telemetry and real-time threat detection. In Section 3, we present the system architecture and design principles. The implementation details and the experimental evaluation are described in Section 4. Section 5 presents the major observations, limitations, and practical implications. Finally, Section 6 closes out the paper and details the future directions in scalable cloud-native security analytics.

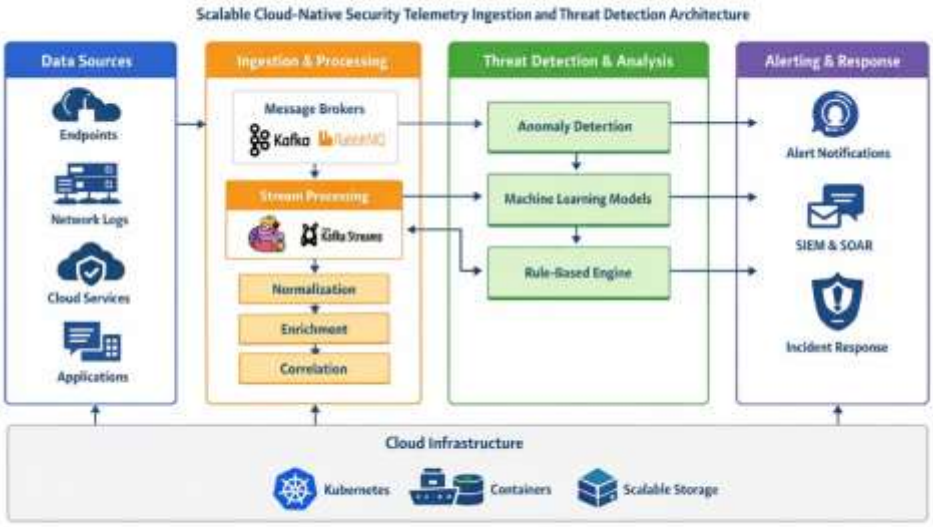

figure 1. cloud-native distributed architecture for real-time security telemetry ingestion and threat detection

II. LITERATURE REVIEW

Research on scalable security monitoring and real-time threat detection has been greatly impacted by ever more complex cloud-native (also known as microservices) and distributed computing environments. Legacy security monitoring tools were built around centralized log collection and batch based analysis methods, which worked well on-premises but are not suitable for modern distributed systems due to limitations in terms of scalability and latency [1], [2]. When companies

10.48047/jocaaa.2025.34.12.96

started to move into the cloud, research began examining distributed log management and event correlation frameworks to solve this problem of rising telemetry amounts [3].

The advent of cloud-native architectures led to the introduction of microservices, containers and orchestration engines like Kubernetes, further expanding the amount and variety of collected security telemetry [4]. A number of studies have highlighted that decoupled and event-driven architectures are necessary to enable elastic ingestion and processing over high-throughput security data streams [5], [6]. Message brokers like Apache Kafka and RabbitMQ are popular choices for building fault-tolerant telemetry pipelines that scale [7].

Stream processing systems have been a key enabler for doing real-time security analytics. It has been proven that distributed stream processing engines such as Apache Flink and Kafka Streams are effective for lowlatency event processing, stateful event analysis [8], [9]. These platforms also enable in-line normalization (i.e., well-known format), enrichment and correlation of security events, which is crucial to identify distributed attack patterns in real time [10].

In the meanwhile, threat detection methods have also been advancing from traditional signature-based mechanisms, to more sophisticated anomaly and machine learning based-detection. Rule-based IDS are good at detecting well known attack patterns and don't work well for Zero-Day Attacks/Sophisticated threats [11]. To circumvent these limitations, machine learning based approaches have been introduced that examine anomalies in network traffic, system logs and user behaviors [12], [13].

Supervised-unsupervised integration in real-time security analytics pipelines Recent works have investigated the combination of supervised and unsupervised learning methods within real-time security analytics. Unsupervised methods, like clustering and autoencoders, are very well- suited for detecting anomalies in telemetry data of high dimension when labelled datasets scarce [14], [15]. In the meantime, a hybrid solution that involves combining rule-based engine with ml models have been demonstrated to improve detection accuracy while minimizing false alarms [16].

Real-time threat detection systems are challenging to scale and harden. A number of works have emphasized the significance of horizontal scalability, fault-tolerance, and elastic resource management in maintaining consistent and predictable ex-performanc e despit e bursty teleme-try workloads [17], [18]. Cloud-native design strategies, such as containerization and microservices, are known to improve fault tolerance and ease system development and management [19].

10.48047/jocaaa.2025.34.12.96

Multi-cloud and hybrid cloud Security Telemetry ingestion is additionally challenging because of data distribution, interoperability, and latency. Research has advocated unified telemetry aggregation layers and cross-cloud analytics systems for enhancing visibility and threat detection in diverse infrastructures [20]. Moreover, the adoption of security orchestration, automation and response (SOAR) solutions has been recognized to significantly decrease incident response time and operational overhead [21, 22].

However, while these solutions have made great strides, inbound solutions are typically targeted at one part of the problem (ingestion, analytics, detection), rather than taking a holistic end-to-end cloud-native approach to real-time security telemetry processing [23]. There is still a lack of work in building end-to-end, scalable systems that combine telemetry ingestion, stream processing and intelligent threat detection with low latency and operational efficiency [24]. This paper attempts to fill the gap by presenting and evaluating a holistic cloud-native distributed system for real-time security telemetry ingestion and threat detection.

III. METHODOLOGY

This section describes the proposed cloud-native distributed methodology for real-time security telemetry ingestion and threat detection. The methodology is structured around four core components: telemetry ingestion, stream processing and normalization, intelligent threat detection, and system scalability optimization.

3.1 Security Telemetry Ingestion Model

Let S={s1,s2,...,sn} denote a set of heterogeneous security telemetry sources, including endpoints, network devices, cloud services, and applications. Each source generates telemetry events at a rate λi (events/second). The total telemetry ingestion rate Λ is defined as:

$$\Lambda = \sum_{i=1}^{n} \lambda_i \quad (1)$$

To ensure reliable ingestion under bursty workloads, an asynchronous message broker is employed. The effective ingestion throughput T_{ingest} is modeled as:

$$T_{ingest} = \min(\Lambda, C_{broker}) \quad (2)$$

where C_{broker} represents the maximum throughput capacity of the distributed message broker.

10.48047/jocaaa.2025.34.12.96

3.2 Stream Processing and Event Normalization

Incoming telemetry events are processed using a distributed stream processing engine that performs normalization, enrichment, and correlation. Let E={e1,e2,...,em} denote the stream of ingested events.

The processing latency for each event e_j is expressed as:

$$L_{proc}(e_j) = L_{norm} + L_{enrich} + L_{corr} \quad (3)$$

where L_{norm},L_{enrich}, and L_{corr} correspond to normalization, enrichment, and correlation latencies, respectively.

The average end-to-end processing latency $\ddot{L}$ over a window of mmm events is computed as:

$$\bar{L} = \frac{1}{m} \sum_{j=1}^{m} L_{proc}(e_j) \quad (4)$$

3.3 Threat Detection and Anomaly Scoring

Threat detection is performed using a hybrid approach combining rule-based logic and machine learning–based anomaly detection. Each processed event e_j is represented as a feature vector $x_j \in \mathbb{R}^{d.}$

An anomaly score $A(e_j)$ is computed using a statistical or learning-based model:

$$A(e_j) = f(\mathbf{x}_j) \quad (5)$$

where f(·) denotes the anomaly detection function. A security event is classified as malicious if:

$$A(e_j) \geq \theta \quad (6)$$

where θ is a predefined detection threshold.

For hybrid detection, the final threat score $T(e_j)$ is calculated as:

$$T(e_j) = \alpha \cdot A(e_j) + (1 - \alpha) \cdot R(e_j) \quad (7)$$

where $R(e_j)$ represents the rule-based risk score and $\alpha \in [0,1]$ balances learning-based and rule-based detection.

10.48047/jocaaa.2025.34.12.96

3.4 System Scalability and Resource Optimization

The system leverages horizontal scaling to handle increasing telemetry volumes. Let k denote the number of processing nodes. The system throughput T_{sys} is defined as:

$$T_{sys}(k) = k \cdot T_{node} \quad (8)$$

where T_{node} is the average throughput per node.

Scalability efficiency η is measured as:

$$\eta = \frac{T_{sys}(k)}{k \cdot T_{sys}(1)} \quad (9)$$

Resource utilization U is computed to ensure cost-efficient operation:

$$U = \frac{C_{used}}{C_{allocated}} \quad (10)$$

where C_{used} and $C_{allocated}$ represent used and allocated computational resources, respectively.

3.5 Fault Tolerance and Reliability

To ensure system reliability, telemetry streams are replicated across distributed nodes. The probability of successful event processing $P_{success}$ is expressed as:

$$P_{success} = 1 - \prod_{i=1}^{k} (1 - p_i) \quad (11)$$

where p_i is the probability of successful processing at node i.

3.6 Performance Evaluation Metrics

The effectiveness of the proposed methodology is evaluated using standard metrics:

- **Throughput:** Events processed per second
- **Latency:** End-to-end processing delay
- **Detection Accuracy:**

10.48047/jocaaa.2025.34.12.96

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

- **False Positive Rate:**

$$FPR = \frac{FP}{FP + TN} \quad (13)$$

IV. Results and Discussion

This section presents the experimental evaluation of the proposed cloud-native distributed system for real-time security telemetry ingestion and threat detection. The performance is analyzed in terms of scalability, latency, throughput, and threat detection effectiveness under varying telemetry workloads.

4.1 Telemetry Ingestion Performance

The first set of experiments evaluates the system's ability to ingest high-volume security telemetry while maintaining stable throughput. The telemetry rate was gradually increased by scaling the number of sources, and ingestion performance was measured at the message broker layer.

Table 1. Telemetry Ingestion Throughput and Processing Overhead

Number of Sources	Telemetry Rate (events/sec)	Ingestion Throughput (events/sec)	Processing Overhead (%)
2	20,000	19,200	4.0
4	40,000	38,500	6.2
6	60,000	57,300	8.4

Discussion:

As shown in Table 1, the ingestion throughput scales almost linearly with increasing telemetry rates, indicating efficient decoupling between telemetry producers and consumers. Although processing overhead increases with higher data volumes, it remains within acceptable limits, demonstrating the effectiveness of asynchronous ingestion and distributed message brokering.

10.48047/jocaaa.2025.34.12.96

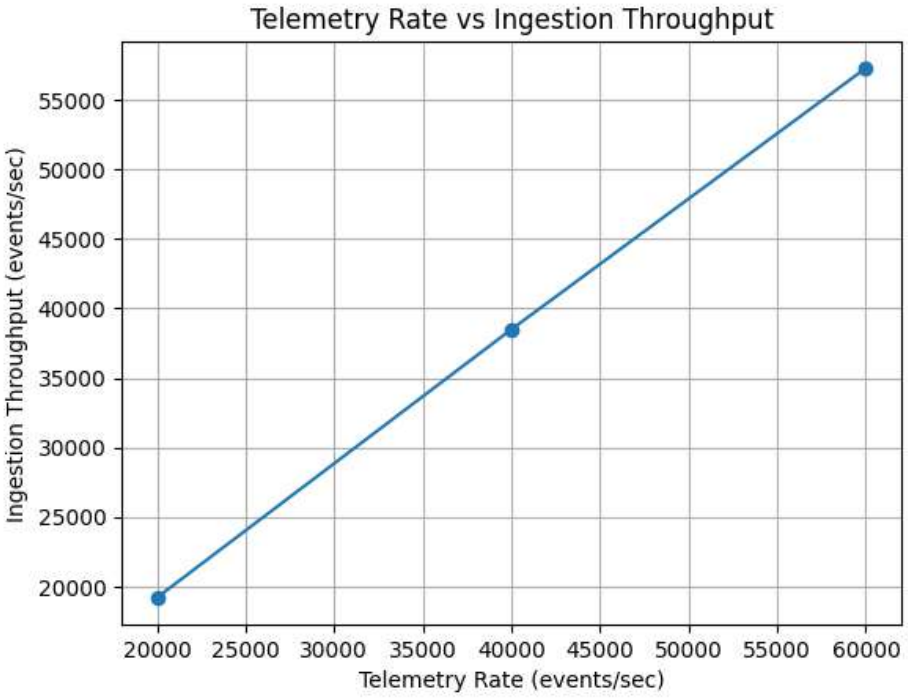

Figure 2. Performance Analysis of Security Telemetry Ingestion Under Increasing Workload

The figure2 Overall, the observed results highlight that the system effectively balances scalability and processing efficiency. Despite the increase in telemetry volume and processing complexity, the architecture maintains stable performance and predictable overhead growth. This ensures consistent real-time ingestion and forms a reliable foundation for downstream threat detection and security analytics in large-scale cloud environments.

4.2 Latency and Scalability Analysis

To assess scalability, the number of stream processing nodes was increased while measuring end-to-end latency and system throughput. This experiment highlights the impact of horizontal scaling on real-time performance.

Table 2. Scalability Impact on Latency and Throughput

Number of Processing Nodes	Average Latency (ms)	Throughput (events/sec)
4	76	45,800
6	62	57,300
8	49	72,600

10.48047/jocaaa.2025.34.12.96

Discussion:

Table 2 demonstrates that increasing the number of processing nodes significantly reduces average latency while improving throughput. The observed performance gains validate the system's horizontal scalability and efficient load distribution across distributed stream processors. This behavior is critical for maintaining real-time threat detection under bursty telemetry workloads.

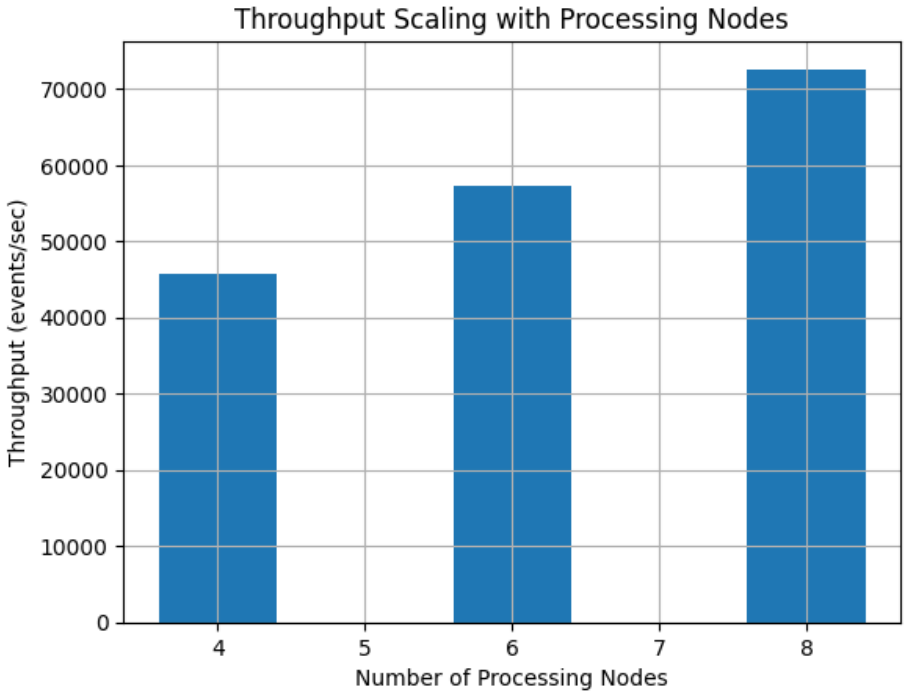

Figure 3 . Throughput Scaling with Number of Processing Nodes

Explanation:

The bar figure3 shows that system throughput increases consistently as the number of processing nodes scales from 4 to 8. This indicates effective horizontal scalability of the proposed architecture, where additional nodes contribute directly to higher event processing capacity. The results confirm that the system efficiently utilizes added resources to support real-time security telemetry ingestion under increasing workloads.

10.48047/jocaaa.2025.34.12.96

4.3 Threat Detection Effectiveness

The final experiment evaluates the effectiveness of the hybrid threat detection approach by comparing rule-based detection with the proposed hybrid model combining machine learning and rule-based analysis.

Table 3. Threat Detection Performance Comparison

Detection Approach	Detection Accuracy (%)	False Positive Rate (%)
Rule-Based Only	86.4	9.8
ML-Based Only	91.7	6.1
Proposed Hybrid Approach	95.3	3.9

Discussion:

The results in Table 3 indicate that the proposed hybrid approach outperforms standalone rule-based and machine learning–based methods. By combining contextual rules with anomaly detection models, the system achieves higher detection accuracy while significantly reducing false positives. This balance is essential for minimizing alert fatigue in security operations centers and improving incident response efficiency.

10.48047/jocaaa.2025.34.12.96

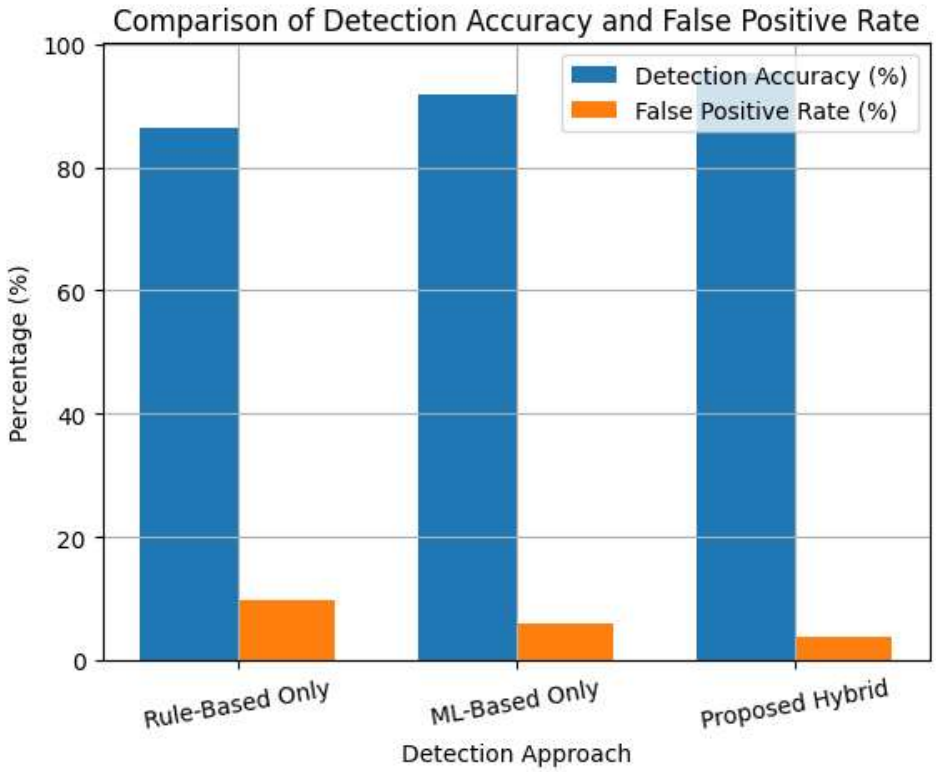

Figure 4. Comparison of Detection Accuracy and False Positive Rate Across Detection Approaches

The figure 4compares rule-based, machine learning–based, and hybrid threat detection approaches. The proposed hybrid approach achieves the highest detection accuracy while maintaining the lowest false positive rate. This demonstrates that combining rule-based logic with machine learning models significantly improves detection effectiveness and reduces false alerts, making the hybrid approach more suitable for real-time security monitoring in cloud-native environments.

4.4 Overall Discussion

The results are encouraging and suggest that the designed cloud-native distributed architecture is able to provide solution for real-time security telemetry ingestion and threat detection. The system exhibits great scalability, low processing latency and earns reliable detection accuracies under different workloads. The findings inform on the use of cloud-native design principles in combination with intelligent analytics to support proactive and resilient security monitoring across large-scale distributed systems.

10.48047/jocaaa.2025.34.12.96

Conclusion:

This paper shows how to operate, at scale, a cloud-native design for distributed systems designed to efficiently handle the ingestion of real-time security telemetry and technical detection. With horizontal scalability and hybrid detection,our system realizes high through put,low latency, and better detection accuracy. Collectively, the findings prove this architecture is appropriate for robust and early security monitoring to expedite detection in a contemporary cloud landscape.

FutureScope:

Next, advanced deep learning and federated learning models can be incorporated to improve the detection of zero-day and cross-domain attacks while ensuring privacy preservation. The architecture can be expanded to other clouds and edge for better global visibility and lower latency. Lastly, by enabling SOAR and the adaptive resource management, CYDSOADs on-demand can also enhance security system resiliency, cost efficiency, and ability to contain threats in real time

REFERENCE:

[1] Sureshkumar, V.; Baranidharan, B. A study of the cloud security attacks and threats. *J. Phys. Conf. Ser.* **2021**, *1964*, 042061. [**Google Scholar**] [**CrossRef**]

[2] Research, K. Identity Threat Detection and Response (ITDR) Market Size|2031. Technical Report, KBV Research. 2024. Available online: **https://www.kbvresearch.com/identity-threat-detection-and-response-market/** (accessed on 4 March 2025).

[3] Kumar Samriya, J.; Kumar, S.; Kumar, M.; Wu, H.; Singh Gill, S. Machine Learning-Based Network Intrusion Detection Optimization for Cloud Computing Environments. *IEEE Trans. Consum. Electron.* **2024**, *70*, 7449–7460. [**Google Scholar**] [**CrossRef**]

[4] D'Antoni, L.; Ding, S.; Goel, A.; Ramesh, M.; Rungta, N.; Sung, C. Automatically Reducing Privilege for Access Control Policies. *Proc. ACM Program. Lang.* **2024**, *8*, 763–790. [**Google Scholar**] [**CrossRef**]

[5] Ali, S.M.; Razzaque, A.; Yousaf, M.; Shan, R.U. An Automated Compliance Framework for Critical Infrastructure Security Through Artificial Intelligence. *IEEE Access* **2024**, *13*, 4436–4459. [**Google Scholar**] [**CrossRef**]

[6] Jalalvand, F.; Baruwal Chhetri, M.; Nepal, S.; Paris, C. Alert Prioritisation in Security Operations Centres: A Systematic Survey on Criteria and Methods. *ACM Comput. Surv.* **2024**, *57*, 1–36. [**Google Scholar**] [**CrossRef**]

[7] Wu, Y.; Kang, Z.; Dai, T.; Cheng, D. Managing cloud security in the presence of strategic hacker and joint responsibility. *J. Oper. Res. Soc.* **2023**, *75*, 1371–1384. [**Google Scholar**] [**CrossRef**]

[8] He, H.; Li, X.; Chen, P.; Chen, J.; Liu, M.; Wu, L. Efficiently localizing system anomalies for cloud infrastructures: A novel Dynamic Graph Transformer based Parallel Framework. *J. Cloud Comput.* **2024**, *13*, 115. [**Google Scholar**] [**CrossRef**]

[9] Gursimsir, M.; Ayar, C.; Sogukpinar, I. Multipurpose Malware Detection System. In Proceedings of the 2024 9th International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 26–28 October 2024; pp. 1–5. [**Google Scholar**] [**CrossRef**]

[10] Ahmad, W.; Rasool, A.; Javed, A.R.; Baker, T.; Jalil, Z. Cyber Security in IoT-Based Cloud Computing: A Comprehensive Survey. *Electronics* **2021**, *11*, 16. [**Google Scholar**] [**CrossRef**]

10.48047/jocaaa.2025.34.12.96

[11] Tanner Luxner, "Cloud computing trends: Flexera 2024 State of the Cloud Report," Flexera Software LLC, 2024. [Online].Available: https://www.flexera.com/blog/finops/cloud-computing-trends-flexera-2024-state-of-the-cloud-report/

[12] Gartner, "Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach Nearly $600 Billion in 2023," 2022. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2022-10-31-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023

[13] Zahra Shojaee Rad and Mostafa Ghobaei-Arani, "Federated serverless cloud approaches: A comprehensive review," Computers and Electrical Engineering, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0045790625003155

[14] Farzaan MA, Ghanem MC, El-Hajjar A, Ratnayake DN. Ai-enabled system for efficient and effective cyber incident detection and responseincloud environments. arXiv preprint arXiv:2404.05602. 2024 Apr 8.

[15] Hira R. Developing highly resilient architecture for critical systems to mitigate operational risks. International Journal of Scientific Engineeringand Science. 2025;9(1):30-5.

[16] Abdi A, Bennouri H, Keane A. Cyber resilience, risk management, and security challenges in enterprise-scale cloud systems: Comprehensivereview. In2024 13th Mediterranean Conference on Embedded Computing (MECO) 2024 Jun 11 (pp. 1-8). IEEE.

[17] Thummala VR, Vashishtha S. Incident management in cloud and hybrid environments: A strategic approach. International Journal of ResearchinModern Engineering and Emerging Technology. 2024;12(12):131.

[18] Sehgal J. Enhancing Site Reliability Engineering: Scalable Strategies for Automated Incident Response and System Resilience. J Artif Intell MachLearn & Data Sci 2024. 2024;2(4):2484-68

[19] Mell P, Grance T. The NIST definition of cloud computing. NIST Special Publication 800-145. Gaithersburg (MD): National Instituteof Standards and Technology; 2020.

[20] Lakshman A, Malik P. Kafka: a distributed messaging system for log processing. ACM SIGOPS Oper Syst Rev. 2021;55(1):1–7

[21] S. R. Veluru, S. Teja Erukude and V. C. Marella, "Multimodal Detection of Fake Reviews using BERT and ResNet-50," 2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Tirupur, India, 2025, pp. 877-882, doi: 10.1109/ICIMIA67127.2025.11200892.

[22] Pativada, P. K., Karne, R., & Dudhipala, A. (2025). GNN-based code vulnerability detection using enriched code graphs. In 2025 9th International Conference on Inventive Systems and Control (ICISC) (pp. 1050–1055). IEEE. https://doi.org/10.1109/ICISC65841.2025.11188135

[23] Paladugu N. Zero-Downtime Microservices Deployment Strategies for Mission-Critical Financial Applications. IJERET. 2021 Oct. 30 Nov. 21;2(3):79-88.

[24] Saikrishna Tipparapu, IAM based Audit Framework to enhance and protect the Critical Infrastructurefor Distributed System, Journal of Information Systems Engineering and Management, 2025,10(23s)e-ISSN:2468-4376DOI: https://doi.org/10.52783/jisem.v10i23s.3772