ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Volume 13, Issue 04, April 2026

Multi-Cloud Data Pipeline for Honeynet Operations and Cyber Threat Intelligence

[1] Mohammad Ramish Anwar, [2] Waseem Khan

[1] Student, Department of Computer Science and Engineering,
Vivekananda College of Technology and Management, Aligarh, India
[2] Head of Department, Department of Computer Science and Engineering,
Vivekananda College of Technology and Management, Aligarh, India
Corresponding Author's Email: [1] inbox.ramish@gmail.com, [2] waseemk49@gmail.com

***Abstract*— Cloud computing has become the backbone of modern digital infrastructure, with enterprises increasingly adopting multi-cloud strategies to achieve scalability, fault tolerance, redundancy, cost efficiency, and vendor independence. However, this shift introduces major challenges in orchestration, unified monitoring, threat analysis, and maintaining security visibility across heterogeneous cloud providers, where security data often remains siloed. At the same time, cyber threats continue to evolve in scale and sophistication, requiring systems that are scalable, resilient, and capable of detecting and analysing malicious activity in real time.**

Honeynets provide controlled environments for observing real-world cyber-attacks and extracting valuable threat intelligence by capturing attacker behaviour, tactics, and tools. Despite their benefits, most existing honeynet deployments remain limited to single-cloud or on-premise environments, resulting in fragmented intelligence, poor scalability, and difficulty processing large volumes of distributed attack data.

This paper proposes the design and implementation of a cloud-native, multi-cloud honeynet framework integrated with a scalable data pipeline for centralized data ingestion, enrichment, analysis, and visualization. Low-interaction honeypots deployed across Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) capture malicious activity, which is aggregated through an automated pipeline using cloud-native services such as serverless compute, logging systems, and message queues. Collected data is enriched using external cyber threat intelligence platforms including AbuseIPDB, VirusTotal, Shodan, and AlienVault OTX, and further processed through automated attack classification techniques.

Interactive dashboards, time-series analytics, and global attack maps provide real-time visualization of adversary activity, improving situational awareness for defenders. Beyond its cybersecurity application, the proposed framework demonstrates how multi-cloud architectures and data engineering principles can be combined to deliver scalable, resilient, and intelligence-driven monitoring systems. The results contribute both as a proof-of-concept honeynet platform and as a broader model for reliable multi-cloud data analytics.

***Keywords*— Multi-cloud computing, Honeynet, Honeypot, Cyber Threat Intelligence, Cloud Security, Data Pipeline, Threat Detection, Security Analytics, Distributed Systems, Real-time Monitoring.**

I. INTRODUCTION

Cloud computing has become a core component of modern IT infrastructure, enabling organizations to access configurable computing resources on demand with minimal management overhead. Through service models such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), cloud platforms provide elasticity, scalability, and pay-as-you-go pricing while reducing upfront hardware costs. However, reliance on a single provider can create vendor lock-in, limiting portability and long-term flexibility.

To address these limitations, many organizations have adopted multi-cloud strategies that combine providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). Multi-cloud environments improve redundancy, resilience, and cost optimization while helping meet regulatory or data residency requirements. Despite these benefits, distributing workloads across multiple providers significantly increases operational complexity, particularly in orchestration, monitoring, and security management.

Security visibility in multi-cloud environments is often fragmented because each provider offers its own proprietary logging and monitoring tools. This creates siloed security data, inconsistent visibility, and difficulty correlating attack activity across platforms. As cyber threats continue to evolve in scale and sophistication, attackers increasingly target cloud-hosted systems using automated and distributed techniques. Traditional monitoring approaches often fail to deliver actionable intelligence at scale.

Honeypots and honeynets provide an effective mechanism for observing malicious behaviour in controlled environments. A honeypot is a decoy system designed to attract attackers, while a honeynet is a network of such systems. These environments capture valuable telemetry on attacker behaviour, tactics, and tools. Low-interaction honeynets are particularly suitable for cloud deployment due to their lightweight and scalable nature. However, most existing deployments remain limited to single-cloud or

ISSN (Online) 2456 -1304
International Journal of Science, Engineering and Management (IJSEM)
Volume 13, Issue 04, April 2026

on-premise environments, resulting in fragmented data collection and limited scalability.

This paper proposes a cloud-native, multi-cloud honeynet framework integrated with a scalable data pipeline for centralized ingestion, enrichment, analysis, and visualization. Low-interaction honeypots deployed across AWS, Azure, and GCP capture attack traffic, which is processed through a fault-tolerant pipeline using message queues, logging systems, and serverless services. External threat intelligence sources enrich captured data, while dashboards and global attack maps provide real-time visibility. The framework demonstrates how multi-cloud architectures can support scalable and intelligence-driven cybersecurity systems.

II. LITERATURE REVIEW

Research on honeypots and honeynets has evolved from isolated deception systems to intelligent, cloud-native security platforms. Early foundational work by Spitzner [1], Provos and Holz [2], and the Honeynet Project [3] established honeypots as effective tools for studying attacker behaviour, collecting malware samples, and understanding exploitation techniques. These studies demonstrated the value of deception-based monitoring but were largely designed for traditional on-premise networks, with limited scalability and manual deployment overhead.

With the rise of cloud computing, researchers began adapting honeynet concepts to virtualized and elastic environments. Chin *et al.* [4] proposed a practical cloud-based threat intelligence framework that integrated honeypot telemetry with automated defensive controls and standardized models such as MITRE ATT&CK. Their work validated real-time detection and response in a single-cloud environment, but did not address cross-cloud visibility or multi-provider orchestration. Similarly, Gisolfi *et al.* [5] presented a HoneyNet environment for analysing malicious actors, showing that cloud-hosted deception systems can capture valuable telemetry at scale, although centralized analytics remained limited.

Recent studies have focused on improving the realism and efficiency of honeynets. Möller [6] introduced an adaptive multi-layered honeynet architecture that dynamically escalates attacker interaction levels using reinforcement learning. This approach improves intelligence yield while controlling infrastructure cost. Wang *et al.* [7] proposed HoneyGPT, which uses large language models to generate realistic responses and overcome the traditional trade-off between realism, scalability, and interactivity. While both works enhance honeypot quality, they provide limited treatment of large-scale data pipelines or multi-cloud deployment.

Another important research direction concerns automated analysis of honeypot data. Lanka *et al.* [8] explored AI-driven interpretation of honeypot logs using machine learning and large language models for summarization and threat detection. Karaarslan *et al.* [9] applied AI agents to Cowrie logs, automatically generating structured summaries of attacker behaviour. These studies show the growing role of intelligent analytics, but depend on clean, centralized datasets generated through robust ingestion pipelines.

Cloud-native data engineering has therefore become central to modern cyber defence systems. Balasubramanian *et al.* [10] introduced a cognitive platform for cyber threat intelligence collection using microservices, streaming pipelines, and NLP models. Babu *et al.* [11] proposed a hybrid cloud management plane for orchestrating distributed data pipelines across heterogeneous environments using Kubernetes-based control planes. Both works validate modular and scalable architectures, though neither focuses specifically on honeynet telemetry.

Threat intelligence enrichment has also received significant attention. Maliga *et al.* [12] addressed rate-limited malware analysis pipelines by reducing redundant API queries and optimizing enrichment workflows. Their findings are directly relevant to systems relying on external services such as VirusTotal, AbuseIPDB, or AlienVault OTX. Such optimization is essential for scalable, cost-aware security analytics.

Security monitoring in containerized cloud environments has been studied by Tan *et al.* [13], who proposed real-time container file monitoring through virtual machine introspection. Their work highlights the importance of resilient monitoring beyond guest operating systems, particularly relevant for containerized honeypot deployments. Madireddy [14] further demonstrated that graph neural networks can detect anomalies in cloud IAM logs, indicating how enriched cloud telemetry may support advanced analytics in future honeynet systems.

Visualization remains a key requirement for operational usability. Studies on dashboards, heatmaps, and time-series security analytics [15] show that visual interfaces significantly improve situational awareness, yet most implementations target single environments and lack multi-cloud scope.

In summary, prior work has advanced deception systems, analytics, cloud orchestration, and CTI pipelines independently. However, an integrated framework combining multi-cloud honeynet deployment, centralized ingestion, enrichment, automated analysis, and real-time visualization remains insufficiently explored. This gap motivates the proposed research.

III. PROBLEM STATEMENT

As organizations increasingly adopt multi-cloud strategies, they benefit from improved flexibility, resilience, cost optimization, and reduced vendor dependence. However, these advantages are accompanied by significant challenges in monitoring, data integration, and security visibility. Each cloud provider offers its own proprietary logging and

ISSN (Online) 2456 -1304
International Journal of Science, Engineering and Management (IJSEM)
Volume 13, Issue 04, April 2026

monitoring services, resulting in fragmented telemetry, inconsistent formats, and limited cross-platform correlation. Consequently, security teams often lack a unified view of attack activity across heterogeneous cloud environments.

These challenges become more severe for security-focused workloads such as honeynets. Deploying, scaling, and maintaining honeypot infrastructure across multiple cloud platforms requires extensive manual configuration, platform-specific expertise, and continuous operational effort. In addition, attack data collected from distributed honeypots is often decentralized, noisy, and difficult to interpret without substantial preprocessing and contextual analysis. The absence of standardized collection and management mechanisms reduces the operational value of captured threat intelligence.

Most existing honeynet implementations in both academic and industrial settings remain limited to single-cloud or on-premise environments. While such systems provide useful insights into attacker behaviour, their scope is restricted by infrastructure boundaries and limited geographic diversity. They also lack interoperability and fail to exploit the scalability and redundancy offered by multi-cloud architectures. As a result, current approaches are insufficient for modern distributed environments where threats operate across multiple platforms and regions.

Another major limitation is the lack of end-to-end automation. Without an integrated pipeline for data ingestion, normalization, enrichment, classification, storage, and visualization, much of the collected telemetry remains underutilized. Raw logs alone provide limited value unless transformed into actionable intelligence through contextual enrichment using external cyber threat intelligence sources and intuitive analytics interfaces.

Therefore, the core research gap lies in the absence of a cloud-native, scalable, and unified framework capable of deploying honeynet workloads across multiple cloud providers while centralizing their outputs for real-time analytics. Addressing this gap requires a solution that combines multi-cloud orchestration with automated threat intelligence workflows. Such a system would simplify deployment, improve scalability, and significantly enhance the quality, usability, and timeliness of security insights for both research and operational defence contexts.

IV. PROPOSED ARCHITECTURE

The proposed system follows a cloud-native, modular architecture designed to collect, process, enrich, and visualize honeynet telemetry across multiple cloud providers. As illustrated in the high-level architecture, the framework integrates distributed honeypot deployments with a centralized data pipeline to provide unified threat intelligence and real-time analytics.

A. Multi-Cloud Honeynet Layer

The first layer consists of low-interaction honeypots deployed across Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). These honeypots emulate commonly targeted services such as SSH and HTTP in order to attract malicious traffic and capture attacker interactions. Deploying honeypots across multiple providers improves geographic coverage, increases infrastructure diversity, and reduces dependence on a single vendor. It also enables comparative observation of attack patterns across different cloud environments.

B. Log Collection and Forwarding Layer

Each honeypot instance generates logs containing connection attempts, authentication events, commands, payloads, and metadata such as source IP addresses and timestamps. Local log forwarders, implemented using Fluentd or Logstash, are deployed near the honeypot nodes to collect and normalize these logs before transmission. This layer ensures reliable shipping of telemetry while reducing inconsistencies in raw log formats produced by different honeypot instances.

C. Streaming and Message Queue Layer

Normalized events are transmitted to a centralized messaging backbone using technologies such as Apache Kafka or AWS Kinesis. This layer decouples data producers from downstream consumers, enabling asynchronous communication, buffering, and fault tolerance. It also allows the system to scale horizontally as the number of honeypots or event volume increases. The queue-based design prevents data loss during temporary outages and supports real-time stream processing.

D. Threat Intelligence Enrichment Layer

Events from the message queue are processed by Python-based enrichment microservices. These services query external cyber threat intelligence platforms such as AbuseIPDB, VirusTotal, AlienVault OTX, and Shodan to enrich captured data with contextual metadata. Examples include IP reputation, geolocation, malware associations, open ports, and known indicators of compromise. This step transforms raw logs into actionable intelligence suitable for deeper analysis.

E. Storage and Data Management Layer

Enriched events are stored in scalable storage backends such as Elasticsearch, Amazon S3, or Google BigQuery. Elasticsearch supports fast indexing and search for real-time dashboards, object storage platforms such as S3 provide cost-efficient archival storage, and analytical warehouses such as BigQuery enable large-scale historical queries. The use of multiple storage options allows the architecture to balance performance, retention, and cost requirements.

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

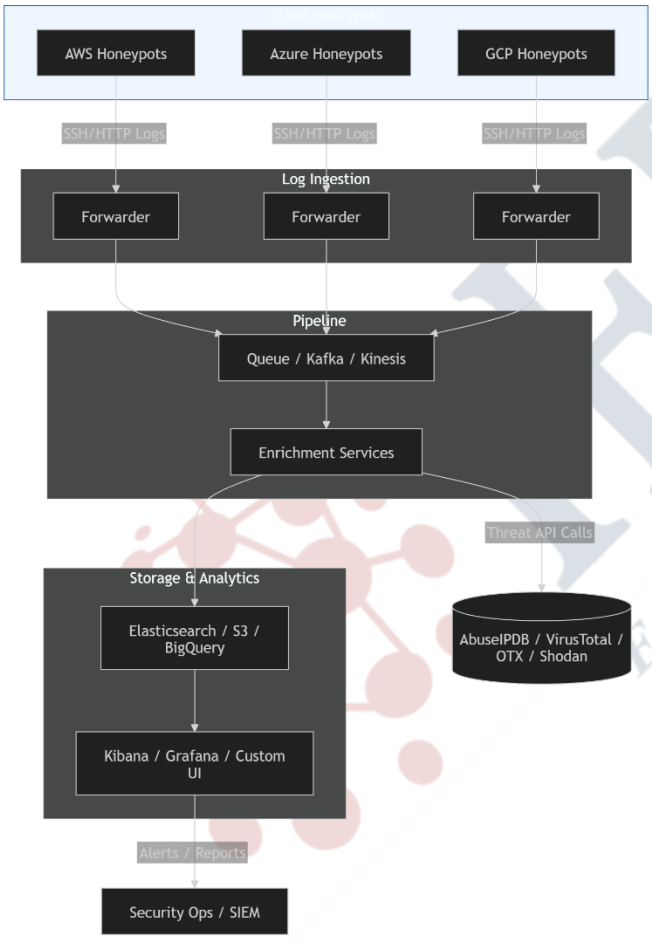
Volume 13, Issue 04, April 2026

F. Visualization and Analytics Layer

The processed data is presented through visualization platforms such as Kibana, Grafana, or a custom web interface. Dashboards display metrics including attack frequency, top targeted services, geographic origin of attacks, time-series trends, and threat intelligence summaries. Interactive global maps and filtering tools improve situational awareness and allow analysts to investigate patterns in real time.

G. Security Operations Integration Layer

The final layer provides optional integration with Security Operations Centre (SOC) workflows, SIEM platforms, and automated reporting systems. Alerts and reports generated from the analytics layer can be forwarded to operational teams for investigation, correlation with enterprise telemetry, or incident response.



H. Design Characteristics

The architecture is designed around key engineering principles:

- Scalability: Additional honeypots or processing nodes can be added without redesign.
- Fault Tolerance: Message queues and distributed services improve resilience to failures.
- Portability: Cloud-agnostic components reduce vendor lock-in.
- Modularity: Each layer can be upgraded or replaced independently.
- Real-Time Visibility: Streaming pipelines and dashboards support near real-time monitoring.

Overall, the proposed architecture demonstrates how multi-cloud infrastructure and cloud-native data engineering can be combined to create a scalable, resilient, and intelligence-driven honeynet monitoring platform.

V. METHODOLOGY

The proposed methodology follows a systematic approach to designing, implementing, and validating a multi-cloud scalable data pipeline tailored for honeynet operations and cyber threat intelligence. The project is executed in sequential phases to ensure modularity, reproducibility, and controlled development.

Phase 1: Requirements Analysis and Design

The first phase focuses on defining the functional and non-functional requirements of the proposed system. Key architectural requirements include scalability, fault tolerance, security, portability, and ease of deployment. Based on these requirements, a high-level architecture is designed to describe the interaction between honeypot workloads, data ingestion mechanisms, storage backends, enrichment services, and visualization dashboards. Appropriate tools and services from AWS, Microsoft Azure, and Google Cloud Platform are mapped to each layer of the architecture.

Phase 2: Multi-Cloud Honeynet Deployment

Low- to medium-interaction honeypots are deployed across multiple cloud environments to capture malicious traffic. Cowrie is used for SSH/Telnet emulation, while Dionaea is used for malware collection and service emulation. Deployment is automated using Infrastructure-as-Code tools such as Terraform, and configuration management is handled through Ansible. Where required, Docker or Kubernetes is used to containerize workloads, improving portability, consistency, and horizontal scalability.

Phase 3: Data Ingestion and Aggregation

Logs and network activity generated by distributed honeypots are forwarded to a centralized ingestion layer. Apache Kafka or AWS Kinesis acts as the real-time streaming backbone for handling continuous event flow. Logstash or Fluentd is used to parse, normalize, and standardize raw log formats before downstream processing. Aggregated data is then stored in scalable platforms such as Elasticsearch, BigQuery, or DynamoDB for indexing, querying, and long-term analysis.

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)

Volume 13, Issue 04, April 2026

Phase 4: Threat Intelligence Enrichment

To increase the analytical value of captured events, the pipeline integrates external cyber threat intelligence APIs including AbuseIPDB, VirusTotal, Shodan, and AlienVault OTX. A modular enrichment layer built with Python microservices performs automated lookups and appends metadata such as IP reputation, malware associations, geolocation, and known indicators of compromise.

Phase 5: Automated Attack Classification

Basic automated classification methods are implemented to categorize malicious activity by protocol, targeted port, attack frequency, and behavioural patterns. Lightweight rule-based heuristics, such as port signatures and frequency thresholds, are prioritized to maintain practicality and avoid unnecessary complexity.

Phase 6: Visualization and Monitoring

A real-time monitoring dashboard is developed using Kibana, Grafana, or a custom React.js/D3.js interface. The dashboard presents attack trends, geographic source maps, time-series metrics, targeted services, and enriched event summaries. It connects directly to the processed datastore for live analytics and investigation.

Phase 7: Scalability and Evaluation

The final phase evaluates the system under varying workloads and deployment scales. Experiments include increasing the number of honeypots, measuring ingestion throughput, and testing resilience under service or node failures. Metrics such as latency, data loss, uptime, and enrichment success rate are analysed to validate reliability and scalability.

This methodology ensures that the primary contribution lies in cloud-native data pipeline design, while honeynet deployment and threat intelligence enrichment serve as practical demonstrations of its effectiveness.

VI. EXPECTED OUTCOMES

The expected outcome of this research is the successful development of a functional, cloud-native, multi-cloud data pipeline that supports honeynet operations and cyber threat intelligence generation. The proposed system is intended to demonstrate that distributed security workloads can be deployed, managed, and analysed efficiently across heterogeneous cloud platforms such as AWS, Microsoft Azure, and Google Cloud Platform.

One primary outcome is the automated deployment of low- to medium-interaction honeypots across multiple cloud providers using Infrastructure-as-Code principles. This will validate the practicality of repeatable, scalable, and vendor-agnostic provisioning for security-focused workloads. The use of automation is also expected to reduce manual configuration effort, minimize deployment inconsistencies, and simplify future expansion of the honeynet environment.

Another key outcome is the creation of a centralized and scalable data ingestion pipeline capable of collecting telemetry from geographically distributed honeypot instances in near real time. By integrating streaming and message queue technologies, the framework is expected to support reliable event transport, fault-tolerant processing, and efficient handling of increased workloads as the number of deployed nodes grows.

The research is also expected to produce enriched datasets by correlating captured honeynet events with external cyber threat intelligence sources such as AbuseIPDB, VirusTotal, Shodan, and AlienVault OTX. This enrichment process should improve the contextual value of raw logs by adding information such as IP reputation, geolocation, known malicious indicators, and attack relevance. As a result, the system is expected to transform raw telemetry into more meaningful and actionable intelligence.

A further outcome is the development of interactive dashboards and monitoring interfaces that provide clear visualization of attack trends, geographic origins, targeted services, and temporal activity patterns. These visual outputs are expected to improve situational awareness and demonstrate the operational value of centralized monitoring across multi-cloud environments.

Finally, the evaluation phase is expected to provide evidence of scalability, reliability, and fault tolerance through metrics such as latency, throughput, uptime, and enrichment success rate. Beyond the immediate cybersecurity use case, the project is expected to serve as a broader case study in cloud-native system design, highlighting how multi-cloud architectures can support resilient, scalable, and intelligence-driven analytics platforms.

VII. CONCLUSION

This paper presented the design of a cloud-native, multi-cloud data pipeline for honeynet operations and cyber threat intelligence, addressing critical challenges associated with security visibility in distributed cloud environments. As organizations increasingly adopt multi-cloud strategies, the need for scalable, unified, and automated monitoring mechanisms becomes more important. Existing approaches often rely on isolated deployments, fragmented telemetry, and provider-specific tools, limiting their effectiveness in modern heterogeneous infrastructures.

The proposed framework demonstrates how honeynet deployment can be combined with centralized data ingestion, threat intelligence enrichment, automated classification, and real-time visualization to create a more operationally relevant security monitoring system. By leveraging multiple cloud providers such as AWS, Microsoft Azure, and Google Cloud Platform, the architecture improves resilience, portability, and coverage while reducing dependence on any single vendor. The use of cloud-native technologies including

Infrastructure-as-Code, message queues, scalable storage, and dashboarding platforms further strengthens the practicality of the solution.

A key contribution of this work lies in transforming raw honeynet telemetry into actionable intelligence. Rather than treating captured logs as isolated data sources, the proposed system enriches events with contextual information from external cyber threat intelligence platforms, enabling more meaningful analysis of attacker behaviour, threat trends, and emerging risks. This demonstrates the value of integrating cybersecurity workflows with modern data engineering practices.

Although the project is currently positioned as a proof-of-concept, it establishes a strong foundation for future enhancements such as machine learning–based attack detection, automated incident response, high-interaction honeypots, and deeper integration with enterprise SIEM or SOC platforms.

In conclusion, the research highlights that multi-cloud architectures, when combined with cloud-native engineering principles, can support effective, scalable, and intelligence-driven cybersecurity systems. Beyond honeynet operations, the proposed framework also serves as a broader model for resilient distributed monitoring and analytics platforms applicable to real-world enterprise environments.

Acknowledgment:

The authors thank the Department of Computer Science and Engineering, Vivekananda College of Technology and Management, for institutional support and academic guidance during the preparation of this manuscript.

REFERENCES

[1] L. Spitzner, *Honeypots: Tracking Hackers*. Addison-Wesley, 2003.

[2] N. Provos and T. Holz, *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. Boston, MA: Addison-Wesley Professional, 2007. Accessed: Aug. 31, 2025. [Online].

[3] The Honeynet Project, "Know Your Enemy: Learning About Security Threats (2nd ed.)." Addison-Wesley Professional, Boston, MA, 2004. Accessed: Aug. 31, 2025. [Online].

[4] D. M. Chin, B. Isfaq, and S. Y. Enoch, "A Practical Honeypot-Based Threat Intelligence Framework for Cyber Defence in the Cloud," Dec. 04, 2025, *arXiv*: arXiv:2512.05321. doi: 10.48550/arXiv.2512.05321.

[5] D. N. Gisolfi, M. Gutierrez, T. V. Rimaldi, C. DeCusatis, and A. G. Labouseur, "A HoneyNet Environment for Analyzing Malicious Actors," in *2018 IEEE MIT Undergraduate Research Technology Conference (URTC)*, Cambridge, MA, USA: IEEE, Oct. 2018, pp. 1–5. doi: 10.1109/URTC45901.2018.9244770.

[6] L. J. Möller, "An Adaptive Multi-Layered Honeynet Architecture for Threat Behavior Analysis via Deep Learning," arXiv.org. Accessed: Apr. 22, 2026. [Online]. Available: https://arxiv.org/abs/2512.07827v1

[7] Z. Wang *et al.*, "HoneyGPT: Breaking the Trilemma in Terminal Honeypots with Large Language Model," arXiv.org. Accessed: Apr. 22, 2026. [Online]. Available: https://arxiv.org/abs/2406.01882v2

[8] P. Lanka, K. Gupta, and C. Varol, "Intelligent Threat Detection—AI-Driven Analysis of Honeypot Data to Counter Cyber Threats," *Electronics*, vol. 13, no. 13, p. 2465, Jun. 2024, doi: 10.3390/electronics13132465.

[9] E. Karaarslan, E. Güler, E. E. Yüce, and C. Coban, "Towards Log Analysis with AI Agents: Cowrie Case Study," arXiv.org. Accessed: Apr. 22, 2026. [Online]. Available: https://arxiv.org/abs/2509.05306v1

[10] P. Balasubramanian, S. Nazari, D. K. Kholgh, A. Mahmoodi, J. Seby, and P. Kostakos, "TSTEM: A Cognitive Platform for Collecting Cyber Threat Intelligence in the Wild," arXiv.org. Accessed: Apr. 22, 2026. [Online]. Available: https://arxiv.org/abs/2402.09973v1

[11] V. Babu, F. Lu, H. Wu, and C. Moberg, "A Hybrid Cloud Management Plane for Data Processing Pipelines," Apr. 11, 2025, *arXiv*: arXiv:2504.08225. doi: 10.48550/arXiv.2504.08225.

[12] D. Maliga, R. Nagy and L. Buttyán, "A pipeline for processing large datasets of potentially malicious binaries with rate-limited access to a cloud-based malware analysis platform," 2024 32nd International Conference on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), Krakow, Poland, 2024, pp. 1-6, doi: 10.1109/MASCOTS64422.2024.10786551.

[13] K. Tan, D. Zhan, L. Ye, H. Zhang, B. Fang, and Z. Tian, "A High-performance Real-time Container File Monitoring Approach Based on Virtual Machine Introspection," Sep. 19, 2025, *arXiv*: arXiv:2509.16030. doi: 10.48550/arXiv.2509.16030.

[14] V. T. Madireddy, "Graph Neural Network Based Adaptive Threat Detection for Cloud Identity and Access Management Logs," Dec. 11, 2025, *arXiv*: arXiv:2512.10280. doi: 10.48550/arXiv.2512.10280.

[15] Kibana: Visualize, explore, and manage data in Elasticsearch," Elastic. Accessed: Apr. 22, 2026. [Online]. Available: https://www.elastic.co/kibana