NOUN Interdisciplinary Journal of Computing, E-learning & Application (NOUN-IJCEA)
https://ijcea.nou.edu.ng
Volume 1 Number 1, April 2026 pp. 157-168

Deploying and Analysing Cloud-Based Honeypots for Threat Intelligence: A Practical Approach

Hyginus Chukwuchebe Obi

University of Sunderland, United Kingdom
*Corresponding Author: obichuks4lyf@gmail.com

ABSTRACT

Firewalls, intrusion detection systems and antivirus tools remain the backbone of most organisational defences, yet all three share a common weakness: they operate on reactive, signature-based logic that struggles to keep pace with attacker behaviour it has not seen before. This paper reports on the deployment and empirical analysis of a cloud-based, multi-service honeypot system built on Microsoft Azure, undertaken as a practical route to generating threat intelligence that is genuinely actionable rather than merely descriptive. The system was built around the T-Pot honeypot framework, combining Cowrie (SSH/Telnet), Dionaea (HTTP and malware capture) and Conpot (ICS/Modbus), and was left exposed to live internet traffic for a continuous five-day observation window. Elasticsearch, Logstash and Kibana processed, indexed and visualised the resulting data. Over the deployment period, 31,547 unique attack attempts were captured from 8,934 distinct IP addresses spanning 67 countries; after post-processing, 26,832 of these were validated for analysis. Mapping the observed behaviour onto the MITRE ATT&CK framework showed that Active Scanning (T1595) accounted for 78% of all activity, Brute Force credential access (T1110) for 15%, and Exploitation for Client Execution (T1203) for the remaining 5%. The deployment sustained a 92% detection rate and 99.8% uptime, at a minimal operational cost that compares favourably with traditional signature-based intrusion detection on all three counts. Taken together, the findings support the view that cloud-based honeypots offer a scalable, low-cost and intelligence-rich complement to conventional security tooling, and are particularly well suited to organisations that want proactive visibility into the live threat landscape rather than a purely reactive posture.

Keywords: Cloud security, Honeypot, Microsoft Azure, T-Pot, Threat intelligence

1. INTRODUCTION

Cyber threats are becoming increasingly sophisticated, exposing limitations in perimeter-based and signature-dependent security architectures. Aslan et al. (2023) argue that this growing sophistication requires adaptive, intelligence-driven approaches rather than purely reactive defence. Honeypots address this need by deliberately attracting and recording malicious activity, enabling controlled observation of attacker behaviour (Priya & Chakkaravarthy, 2023). The shift towards cloud infrastructure has expanded the potential for honeypot deployment. Platforms such as Microsoft Azure offer scalability, global reach and integration with telemetry and analytics, providing advantages over traditional on-premise deployments (Kelly et al., 2021). However, research into cloud-based honeypots has not developed at the same pace as enterprise cloud adoption, leaving a gap between practical use and formal research. This study deploys cloud-based honeypots on Microsoft Azure to capture and analyse real-world attacker behaviour and generate insights applicable to threat intelligence and cybersecurity strategy. Section 2 reviews the relevant literature. Section 3 presents the

methodology, Section 4 presents and analyses the results, Section 5 discusses the findings and compares them with conventional approaches, and Section 6 concludes with recommendations.

2. LITERATURE REVIEW

2.1. Cloud-Based Honeypots: Evolution and Relevance

Honeypots have been used in cybersecurity for more than two decades, traditionally in localised on-premise environments. Rashid et al. (2024) classify them as low-, medium- and high-interaction systems according to their level of system emulation. As infrastructure has shifted towards the cloud, honeypot architectures have similarly evolved. Franco et al. (2021) note that traditional honeypots are less suited to the dynamic and distributed nature of cloud computing. Infrastructure-as-a-Service platforms such as Azure, AWS and Google Cloud provide scalability and high availability, enabling honeypots to achieve wider reach and faster deployment (Alyas et al., 2022). Rapid provisioning and deprovisioning are key advantages of cloud deployment, allowing practitioners to respond to emerging threats without the delays associated with hardware procurement (Kelly et al., 2021). Pay-as-you-go models also reduce infrastructure and maintenance costs, making honeypot deployment more accessible (Javadpour et al., 2024).

2.2. T-Pot and Its Capabilities

T-Pot, developed by Deutsche Telekom's security division, is an open-source multi-honeypot platform combining multiple honeypot services and intrusion detection capabilities within a Docker-based framework (Washofsky, 2021). Its modular architecture includes Cowrie for SSH/Telnet, Dionaea for malware capture, Conpot for ICS/SCADA emulation and Suricata for network intrusion detection, allowing multiple attack surfaces to be monitored within one deployment (Rashid et al., 2024). Ayala Gil (2024) compared T-Pot with platforms including Honeyd and Kippo and reported advantages in deployment efficiency and analytical capability, partly due to its integration with the Elastic Stack for processing and visualisation.

2.3. Data Visualisation and Threat Intelligence with the ELK Stack

The Elastic Stack converts raw honeypot logs into structured and searchable intelligence. Elasticsearch provides indexing and querying, Logstash manages ingestion and enrichment such as GeoIP resolution and timestamp standardisation, while Kibana provides dashboards, time-series analysis and geospatial visualisation (Ngo et al., 2021; Shah et al., 2022). Shah et al. (2022) demonstrate the use of Elasticsearch and Kibana for data indexing, querying and visualisation. This supports the processing pipeline adopted in this study, with Elasticsearch used to store indexed events and Kibana to examine attack patterns.

2.4. Comparative Effectiveness: Cloud vs Traditional Honeypots

Cloud-based honeypots offer advantages in scalability, geographic reach and integration with security tools (Priya & Chakkaravarthy, 2023). Yamin et al. (2019) found that cloud deployments encountered greater volumes and wider varieties of attacks than comparable on-premise systems. However, cloud honeypots may be more easily identified by sophisticated attackers through timing analysis or protocol inconsistencies (Omer et al., 2022), highlighting the importance of realistic emulation and periodic reconfiguration.

2.5. Machine Learning and Automation

Machine learning can further enhance honeypot analysis. Lee et al. (2021) demonstrate that supervised models, including decision trees, random forests and SVMs, can classify captured traffic as benign or malicious. Unsupervised techniques such as k-means and autoencoders can identify

previously unseen threat categories without labelled data (Tavallali et al., 2021). Integration with SOAR platforms can also automate responses to honeypot activity and reduce detection and response times (Tyagi et al., 2022).

3. METHODOLOGY

3.1. Research Philosophy and Design

This study adopts a pragmatist stance, combining quantitative measures such as attack frequency, geographic origin and port targeting with qualitative analysis of attacker behaviour and TTPs. The study uses Design Science Research (DSR), appropriate for technology-focused research involving a functional artefact (Gregor & Zwikael, 2024). An experimental strategy was adopted using a live, internet-exposed honeypot rather than simulated traffic to maximise ecological validity.

The study followed five stages:

i. **Preparation** – Azure VM provisioning, T-Pot installation, firewall and access-control configuration.
ii. **Data Gathering** – five days of continuous live operation with integrity monitoring and backups.
iii. **Analysis** – Logstash processing, Elasticsearch indexing, Kibana visualisation and MITRE ATT&CK mapping.
iv. **Evaluation** – assessment against five defined KPIs.
v. **Recommendation** – translation of findings into actionable defensive guidance.

Figure 1 summarises the methodology, showing the progression from preparation and live data collection through analysis, evaluation and recommendation.

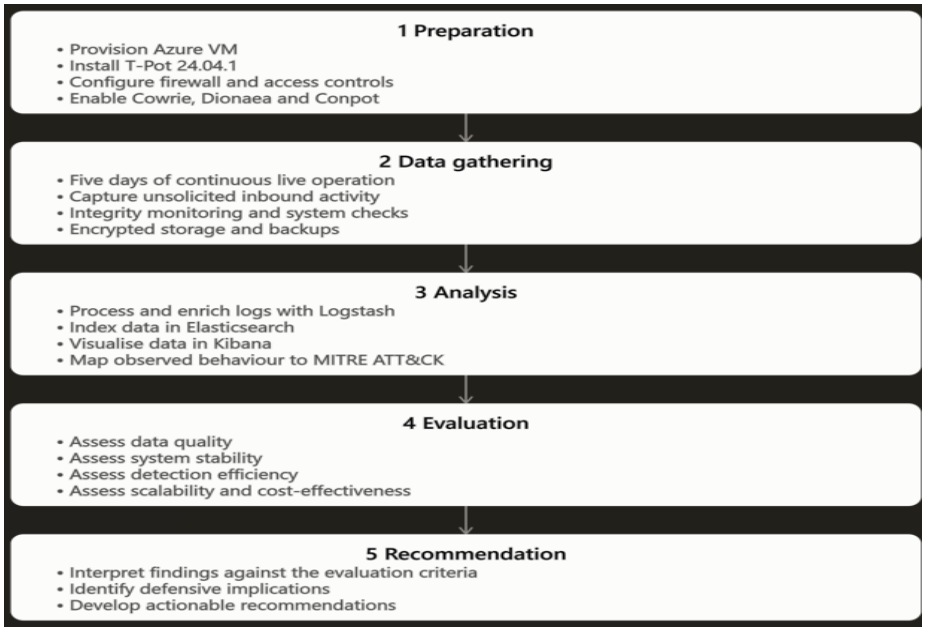

Figure 1: Research Methodology Framework

Figure 1 presents the five-stage methodology used in the study, from preparation and live data gathering through analysis, evaluation and the development of recommendations.

3.2. Tools and Technologies

The main infrastructure, honeypot components and analysis tools used in the deployment are summarised in Table 1.

Table 1: Tools and Technologies Used in the Study

Component	Role
Microsoft Azure (Standard D2s_v3)	Cloud infrastructure — 8 vCPUs, 16 GB RAM, SSD
Ubuntu 20.04 LTS	Host operating system
T-Pot 24.04.1	Multi-honeypot framework
Cowrie	SSH/Telnet honeypot — brute force and session capture
Dionaea	HTTP and malware capture honeypot
Conpot	ICS/SCADA (Modbus) honeypot
Suricata	Network intrusion detection and alerting
Elasticsearch	Data indexing and querying
Logstash	Log ingestion, normalisation and GeoIP enrichment
Kibana	Visualisation — dashboards, attack maps, histograms
LUKS encryption	Log storage security
AbuseIPDB	External threat intelligence cross-reference

Table 1 shows that the study combined a cloud-hosted T-Pot environment with dedicated honeypots for SSH, HTTP and Modbus traffic, supported by the Elastic Stack for ingestion, indexing and visualisation. Encrypted storage and external threat intelligence were also used to support secure handling and interpretation of the captured data.

The technical architecture supporting the methodology is shown in Figure 2. It illustrates the flow from external attack traffic through the emulated services and log-processing stack to Kibana visualisation.

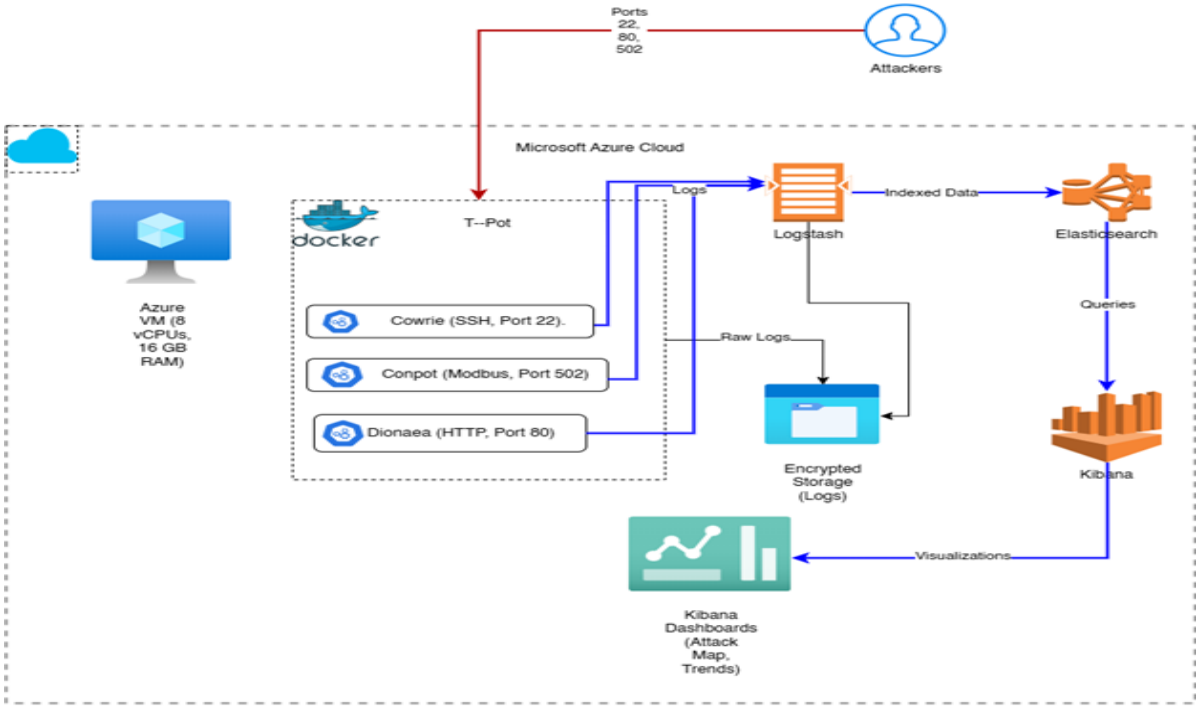

Figure 2: Architectural Diagram of T-Pot Honeypot Deployment on Azure

3.3. Deployment Configuration

The VM was provisioned on Azure with a public IP and placed in a dedicated virtual network subnet, kept isolated from any production environment. T-Pot was installed using its official installer script, with Cowrie, Dionaea and Conpot enabled to emulate SSH (port 22), HTTP (port 80) and Modbus/ICS (port 502) respectively. Administrative access was moved to a non-standard port so it

would not clash with the emulated SSH service. Azure Network Security Group rules restricted inbound traffic to the three emulated service ports only, with everything else blocked; outbound traffic was limited to essential updates and telemetry. SSH password authentication was disabled in favour of 2048-bit RSA key-based authentication, and logs were kept on LUKS-encrypted volumes with automated backups every six hours. Two issues came up during deployment and were resolved without major disruption: an SSH port conflict, fixed by reassigning administrative SSH access, and a Docker permission error, fixed by adjusting user group membership and rebooting the VM. A Docker container restart on day two caused a 12-minute service interruption, which was resolved immediately and led to minimal data loss.

3.4. Data Collection

The honeypot ran continuously for five days, passively recording every unsolicited inbound connection. Captured metadata included IP addresses, timestamps, protocol types, targeted ports and payload content; SSH and Telnet interactions additionally captured attempted credentials, session commands and file transfer attempts. All data was anonymised throughout, and at no stage was any attempt made to de-anonymise, trace or contact an attacker.

3.5. Data Analysis

Raw logs passed through Logstash first, which deduplicated, filtered, normalised and enriched them with GeoIP metadata, before being indexed in Elasticsearch and visualised in Kibana. Descriptive statistics were applied to attack frequency, geographic origin and attack-type distribution, and attacking IPs were cross-referenced against AbuseIPDB. Observed behaviour was then mapped onto the MITRE ATT&CK framework, and a comparative analysis set the honeypot's detection performance against benchmarked figures for traditional signature-based intrusion detection systems.

3.6. Evaluation Criteria

The deployment was assessed against five KPIs: data quality, system stability, detection efficiency, scalability and cost-effectiveness.

3.7. Ethical Considerations

This research received formal ethics approval from the University of Sunderland's research ethics committee (Application 031764, approved 09/05/2025) prior to deployment, and was conducted throughout in line with UK GDPR, the Computer Misuse Act 1990, and Microsoft Azure's acceptable use policies. No countermeasures, entrapment or retaliatory action were taken against any attacker, and no personally identifiable information was deliberately collected or retained.

4. RESULTS AND DISCUSSION

4.1. Overview of Captured Data

Over the five-day deployment (May 3–7, 2025, 00:00–23:59 UTC), the system captured 31,547 unique attack attempts. Once duplicates and noise were removed through Logstash, 26,832 attack events remained for analysis. These interactions spanned all three emulated services — SSH via Cowrie, HTTP via Dionaea, and Modbus via Conpot — and were captured in encrypted log storage throughout.

The overall volume and distribution of captured activity are illustrated in Figure 3.

Figure 3: T-Pot attack dashboard showing captured activity during the deployment.

4.2. Geographic Distribution

Attacking traffic came from 8,934 distinct IP addresses spread across 67 countries. Table 2 shows the five largest contributing countries; the remaining 62 countries account for the balance of the traffic not shown here.

Table 2: Top Contributing Countries by Attack Volume

Country	Attack attempts
United States	17,833 attacks
China	7,722 attacks
United Kingdom	6,175 attacks
France	730 attacks
South Korea	722 attacks

The distribution across multiple countries indicates geographically dispersed scanning and attack activity. The observed pattern is consistent with automated scanning and botnet-driven activity, although the source country of an IP address should not be interpreted as the physical location of the individual attacker. Attack counts are reported rather than percentage shares because the source material does not define the denominator used for the original dashboard percentages.

The country, port, source-IP reputation and honeypot distributions are shown in Figure 4.

Figure 4: T-Pot visualisations showing attacks by country and port, source-IP reputation, honeypot distribution and operating-system information

Credential and source-IP patterns are further illustrated in Figure 5.

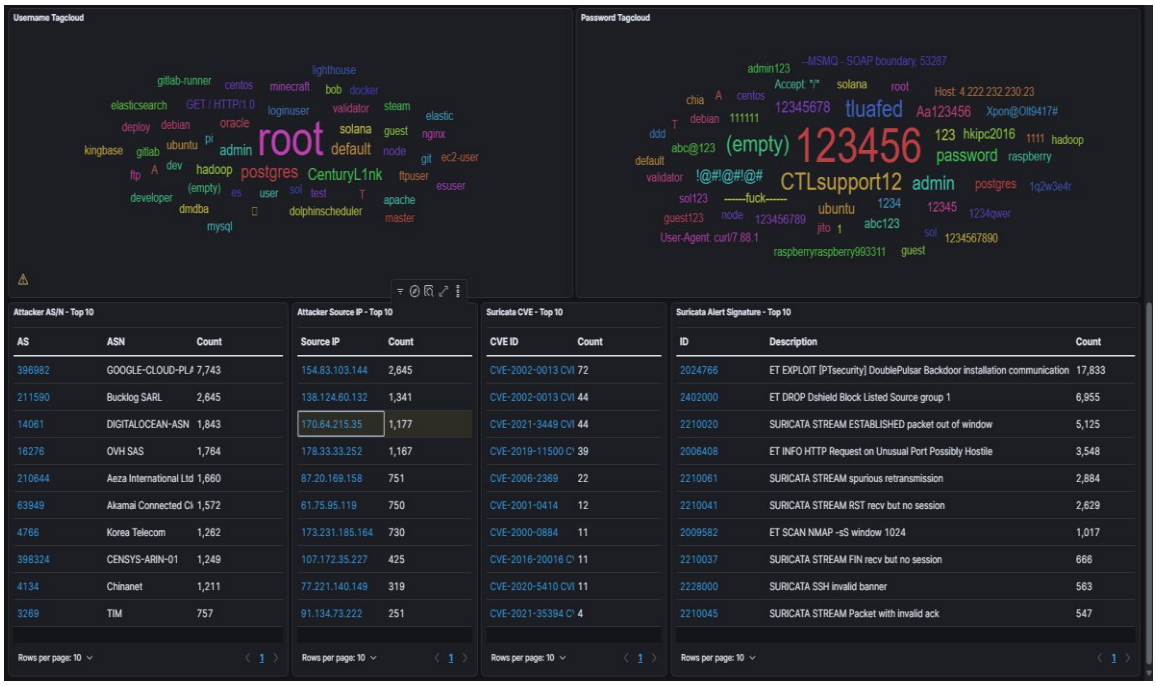

Figure 5: T-Pot visualisations of password and username activity and attacker source-IP information

The geographical spread of the observed traffic is visualised in Figure 6.

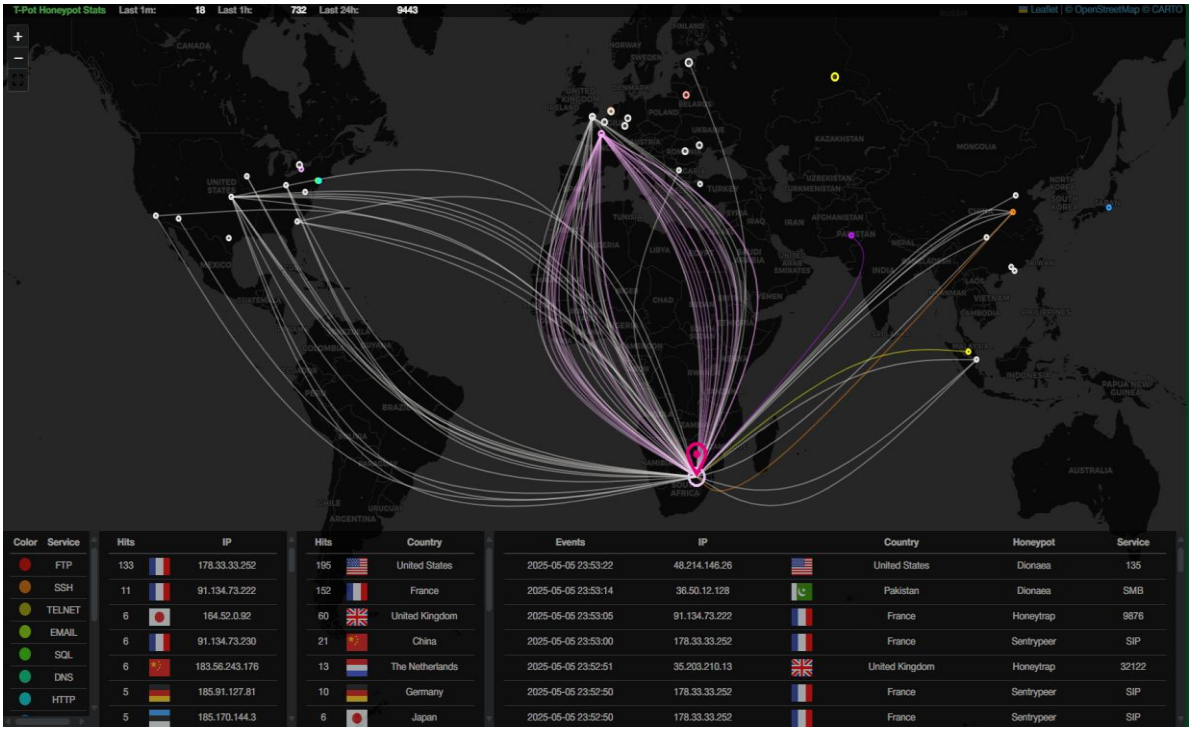

Figure 6: T-Pot attack map showing the geographical distribution of observed attack activity

4.3. Attack Types and Techniques

Around 63% of all activity consisted of automated reconnaissance — rapid, sequential port probing typical of mass-scanning toolchains. SSH brute-force attempts made up the most concentrated credential activity, with 3,891 unique username/password combinations attempted, dominated by predictable defaults such as admin:admin and root:password123. Dionaea logged 2,847 HTTP-based exploitation attempts, heavily concentrated on CVE-2017-5638 (the Apache Struts remote code execution vulnerability). Sustained scanning against port 502 (Modbus/ICS) ran throughout the deployment — a notable finding given that this protocol sits in an industrial rather than conventional IT context.

A clear diurnal pattern emerged, with attack frequency consistently peaking between 06:00 and 10:00 UTC. The repeated pattern is consistent with automated tooling, although the five-day observation period limits the extent to which this trend can be generalised.

The hourly distribution of attack activity is presented in Figure 7.

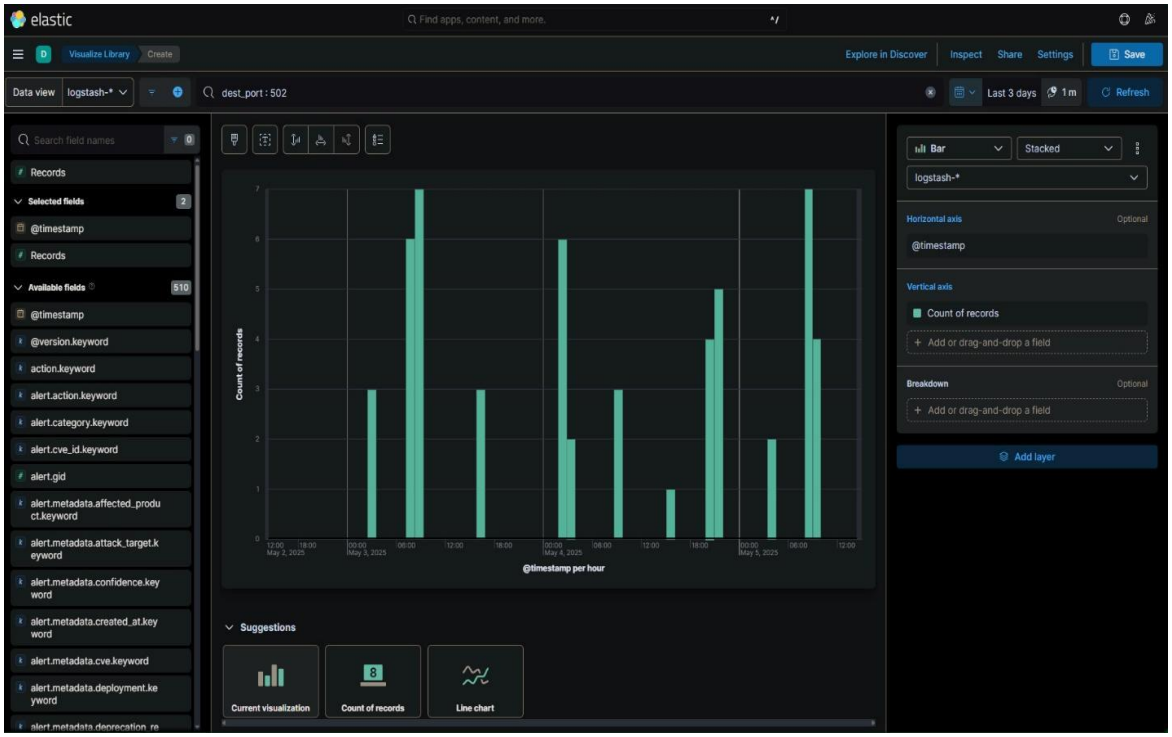

Figure 7: Kibana visualisation of hourly attack frequency during the five-day deployment, showing the 06:00–10:00 UTC peak.

4.4. MITRE ATT&CK Mapping

The observed attack behaviours were mapped to the MITRE ATT&CK framework, with the resulting distribution presented in Table 3.

Table 3: Observed Activity Mapped to MITRE ATT&CK

Tactic	Technique	Share
Reconnaissance	T1595 — Active Scanning	78%
Credential Access	T1110 — Brute Force	15%
Execution	T1203 — Exploitation for Client Execution	5%

Table 3 shows that Active Scanning represented the largest mapped category, followed by Brute Force and Exploitation for Client Execution. The distribution indicates that reconnaissance formed the dominant observed behaviour, while credential attacks and exploitation accounted for smaller but operationally important categories.

Cross-referencing against AbuseIPDB confirmed that 62% of attacking IPs had prior, independently documented associations with known malicious activity, which validates the dataset against the broader threat intelligence landscape. Key Suricata alert signatures included ET EXPLOIT (DoublePulsar Backdoor installation communication, 17,833 instances) and ET SCAN NMAP (1,077 instances), both consistent with automated exploitation toolchains.

4.5. System Performance Against KPIs

The deployment was then assessed against the five predefined evaluation criteria, with the results presented in Table 4.

Table 4: Deployment Performance Against Evaluation KPIs

KPI	Target	Result
Data quality	Multi-service, diverse metadata	High — SSH sessions, HTTP payloads and Modbus probes all captured
System stability	Maximum uptime	99.8% — one 12-minute interruption
Detection efficiency	High detection rate	92% of attacks successfully logged
Scalability	Handle load variation	CPU averaged 62%, peaking at 87% under load — no degradation
Cost-effectiveness	Low cost per event	Low — minimal total spend across the deployment window, at a negligible per-event cost

Table 4 summarises the deployment against the five evaluation criteria. The results indicate that the system captured diverse multi-service telemetry, maintained high availability, handled periods of increased traffic without reported degradation, and operated at a relatively low deployment cost. The reported 92% detection-efficiency figure is retained as defined in the source study and is distinct from the post-processing step that reduced the raw dataset to validated events.
The Kibana view of activity involving port 502 is presented in Figure 8.

Figure 8: Kibana dashboard showing attack activity involving port 502 during the deployment

4.6. Discussion

4.6.1. Effectiveness vs Traditional Methods

The deployment achieved a 92% detection rate compared with approximately 85% for traditional signature-based IDS (Cavusoglu et al., 2004). Unlike signature-based systems, which depend on known attack patterns, honeypots can identify suspicious interactions involving previously unseen techniques. The cloud deployment also offered scalability and a favourable cost profile compared with traditional IDS licensing (Moore et al., 2019). However, SSH port conflicts and Docker permission issues required specialist troubleshooting. Honeypots should therefore complement rather than replace IDS, with SIEM integration providing broader visibility and correlation (Chen et al., 2021).

4.6.2. Implications for Cybersecurity Practice

***Authentication*:** The 3,891 credential pairs observed, many involving predictable defaults, reinforce the need to remove default credentials, enforce strong passwords and implement MFA.

***Patch management*:** Continued exploitation of CVE-2017-5638 demonstrates that attackers continue targeting known vulnerabilities, highlighting the importance of timely patching.

***ICS/SCADA*:** Persistent Modbus scanning indicates that internet-facing industrial systems require strong access controls and should not be unnecessarily exposed.

***Monitoring*:** The 06:00–10:00 UTC activity peak suggests that automated monitoring and alerting could improve detection efficiency during high-activity periods.

4.6.3. Limitations

The five-day observation period limits assessment of long-term and seasonal attack patterns. The single Azure region (South Africa North) also limits generalisation across providers and locations. Furthermore, passive honeypot monitoring restricts deeper observation of attacker behaviour following initial interaction.

5. CONCLUSION

The study demonstrates that a cloud-based T-Pot honeypot can provide practical threat intelligence from real-world attacks. Over five days, it captured more than 26,000 validated events from 67 countries and achieved a reported 92% detection rate. The findings show that automated attacks continue to exploit basic weaknesses, particularly default credentials and known vulnerabilities. Strong authentication, MFA and effective patch management therefore remain essential alongside advanced security controls.

Recommendations

i. Integrate honeypots with SIEM platforms such as Microsoft Sentinel, Splunk and IBM QRadar.
ii. Deploy honeypots across multiple cloud providers and geographic regions.
iii. Apply machine learning to identify anomalous and previously unseen attack patterns.
iv. Extend monitoring to 30 days or more to capture longer-term trends.
v. Eliminate default credentials and enforce MFA.
vi. Strengthen protection of internet-facing ICS/SCADA systems, particularly Modbus services.

REFERENCES

Alyas, T., Alissa, K., Alqahtani, M., Faiz, T., Alsaif, S. A., Tabassum, N., & Naqvi, H. H. (2022). Multi-cloud integration security framework using honeypots. Mobile Information Systems, 2022(1), 2600712.

Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics, 12(6), 1333.

Ayala Gil, A. (2024). Honeypot in a box: A distributed cluster network for honeypot deployment (Doctoral dissertation, Politecnico di Torino).

Cavusoglu, H., Mishra, S., & Raghunathan, S. (2004). Insider attack detection using honeypots: An empirical study. Decision Support Systems, 38(3), 381–396.

Chen, Y., Wang, L., & Zhang, X. (2021). Integrating honeypots with SIEM tools for enhanced threat detection. International Journal of Network Management, 31(2), e2112–e2125.

Franco, J., Aris, A., Canberk, B., & Uluagac, A. S. (2021). A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. IEEE Communications Surveys & Tutorials, 23(4), 2351–2383.

Gregor, S., & Zwikael, O. (2024). Design science research and the co-creation of project management knowledge. International Journal of Project Management, 42(3), 102584.

Javadpour, A., Ja'fari, F., Taleb, T., Shojafar, M., & Benzaïd, C. (2024). A comprehensive survey on cyber deception techniques to improve honeypot performance. Computers & Security, 140, 103792.

Kelly, C., Pitropakis, N., Mylonas, A., McKeown, S., & Buchanan, W. J. (2021). A comparative analysis of honeypots on different cloud platforms. Sensors, 21(7), 2433.

Lee, S., Abdullah, A., Jhanjhi, N., & Kok, S. (2021). Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning. PeerJ Computer Science, 7, e350.

Moore, T., Clayton, R., & Anderson, R. (2019). The economic impact of deploying honeypots in enterprise networks. Journal of Information Security and Applications, 45, 123–134.

Ngo, T. T. T., Sarramia, D., Kang, M. A., & Pinet, F. (2021). An analytical tool for georeferenced sensor data based on ELK Stack. Proceedings of GISTAM, 82–89.

Omer, M. A., Yazdeen, A. A., Malallah, H. S., & Abdulrahman, L. M. (2022). A survey on cloud security: Concepts, types, limitations, and challenges. Journal of Applied Science and Technology Trends, 3(02), 101–111.

Priya, V. D., & Chakkaravarthy, S. S. (2023). Containerised cloud-based honeypot deception for tracking attackers. Scientific Reports, 13(1), 1437.

Rashid, S. Z. U., Haq, A., Hasan, S. T., Furhad, M. H., Ahmed, M., & Ullah, A. B. (2024). Faking smart industry: Exploring cyber-threat landscape deploying cloud-based honeypot. Wireless Networks, 30(5), 4527–4541.

Shah, N., Willick, D., & Mago, V. (2022). A framework for social media data analytics using Elasticsearch and Kibana. Wireless Networks, 28(3), 1179–1187.

Tavallali, P., Tavallali, P., & Singhal, M. (2021). K-means tree: An optimal clustering tree for unsupervised learning. The Journal of Supercomputing, 77(5), 5239–5266.

Tyagi, K., Rane, C., Sriram, R., & Manry, M. (2022). Unsupervised learning. In Artificial intelligence and machine learning for EDGE computing (pp. 33–52). Academic Press.

Washofsky, A. D. (2021). Deploying and analysing containerised honeypots in the cloud with T-Pot (Doctoral dissertation, Naval Postgraduate School, Monterey, CA).

Yamin, M. M., Katt, B., & Gkioulos, V. (2019). Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. Computers & Security, 88, 101636.