Deception Meets Diagnostics: Deception-based Real-Time Threat Detection in Healthcare Web Systems

Zeeshan Zulkifl Shah
School of Computing
Macquarie University
Sydney, Australia
zeeshanzulkifl.shah@hdr.mq.edu.au

Muhammad Ikram
School of Computing
Macquarie University
Sydney, Australia
muhammad.ikram@mq.edu.au

Hassan Jameel Asghar
School of Computing
Macquarie University
Sydney, Australia
hassan.asghar@mq.edu.au

Mohamed Ali Kaafar
School of Computing
Macquarie University
Sydney, Australia
dali.kaafar@mq.edu.au

***Abstract*—Increased cloud adoption in healthcare has amplified ransomware and malware threats, accounting for 19% of global breaches in 2024. Despite this surge, the behavior of attackers exploiting healthcare systems remains under-explored in academic literature. This paper bridges that gap by deploying a scalable and stealthy deception network specifically designed for healthcare environments. The network comprises 30 real-world vulnerable healthcare web applications, mimicking domain-specific workflows across multi-cloud infrastructures, such as patient registration and billing. We leveraged ATTACK-BERT to generate semantic embeddings and applied co-regularized spectral clustering with normalized cuts to analyze multi-protocol attack traffic. Our analysis revealed nuanced attacker behaviors, including regional and protocol-specific variations, exploitation of healthcare protocols like HL7, and the use of encryption to bypass detection. A comparative sub-study further showed that attackers deliberately engage with vulnerable systems, highlighting the strategic value of deception-based defenses. By focusing on behavioral insights within healthcare-specific settings, this work lays the groundwork for integrating deception into the broader security posture of critical infrastructures.**

I. Introduction

The post-COVID era has seen rapid healthcare digitization globally. A significant number of healthcare organizations have migrated to cloud-based technologies, driven by the need for scalability, improved data management, and enhanced collaboration [1]. By 2023, approximately 70% of healthcare organizations had transitioned to cloud-based solutions, with an additional 20% expected to migrate by 2025 [2].

While this shift has brought significant operational benefits, it has also introduced substantial cyber risks. The healthcare sector, in particular, has become a prime target for cyberattacks due to the sensitive and high-value data it processes, including Electronic Medical Records (EMRs), billing information, and diagnostic workflows. In 2023 alone, over 87 million patient records were compromised, a dramatic increase from 37 million in 2022 [3]. These statistics highlight not only the sector's growing vulnerability but also its attractiveness to sophisticated attackers who exploit healthcare systems' reliance on interconnected, cloud-based infrastructures.

Cloud-based healthcare applications provide access to healthcare professionals, patients, and other stakeholders via the Web. However, securing these applications presents unique challenges due to their dependence on domain-specific standards and protocols, such as: the International Classification of Diseases (ICD-10) [4] for diagnoses and the National Drug Code (NDC) [5] for drug identifiers and on the Health Level Seven (HL7) messaging protocol [6][1] to move those codes between systems.

While these structured coding systems and messaging standards enhance operational efficiency and interoperability in healthcare, they also introduce predictable attack surfaces that can be leveraged by adversaries. In particular, the widely adopted HL7 v2.x protocol including message types such as Admission-Discharge-Transfer (ADT) and Order Entry Message (ORM) is frequently transmitted over unsecured channels [7], [8]. This lack of encryption and authentication significantly increases the exposure of healthcare networks to targeted cyberattacks [9]–[11].

In addition, traditional general-purpose security solutions, such as firewalls, intrusion detection systems, and security information and event management (SIEM) platforms, are often insufficient to secure healthcare infrastructures, which operate under unique regulatory, operational, and technological constraints. Several studies [12] [13] [14] have highlighted that these conventional tools frequently overlook domain-specific threats, such as those exploiting healthcare protocols (e.g., HL7) or legacy medical systems. This highlights the need for customized cybersecurity strategies that integrate domain expertise with modern defense mechanisms.

In parallel, as healthcare organizations increasingly migrate to cloud environments, new risks, particularly those involving the protection of sensitive patient data have emerged. To mitigate these risks, Attack Surface Management (ASM) has become a widely adopted defense strategy. ASM typically involves: *(i)* the continuous discovery, monitoring, and as-

[1]ICD-10 and NDC are *classification* schemes; HL7 is a *transport* protocol. In our deployment, ICD-10/NDC values are carried inside HL7 message fields.

sessment of external-facing digital assets, and *(ii)* the identification of software vulnerabilities, exposed services, and mis-configurations [15]–[17]. While ASM offers a structured framework for identifying external threats, its applicability in healthcare remains limited. This is primarily due to domain-specific standards such as ICD-10, NDC, and HL7 which create unique and complex attack surfaces that generic ASM tools are ill-equipped to monitor.

These limitations are evident in real-world incidents. For example, in Australia, the healthcare sector accounted for 19% of all data breach notifications under the Notifiable Data Breaches (NDB) scheme in the first half of 2024 [18]. Globally, targeted ransomware attacks and data ex-filtration campaigns have increasingly focused on cloud-based healthcare infrastructures [19], underscoring the need for proactive, domain-aware security approaches that extend beyond traditional ASM capabilities.

To fill this gap, deceptive technologies particularly honeypots offer a promising, complementary solution. Honeypots are decoy systems designed to attract attackers and capture detailed telemetry of their behavior. When tailored to emulate realistic healthcare workflows and vulnerabilities, they yield rich insights into attacker methods, preferences, and intent [3]. Such intelligence enables the design of bespoke cybersecurity defenses that are responsive to the unique characteristics and evolving threat landscape of the healthcare domain.

This paper presents a healthcare-specific deception network designed to simulate real-world attack surfaces through vulnerable web applications. These applications replicate key healthcare workflows to attract and monitor malicious activity in cloud environments. To ensure consistency and fairness in our evaluation, we deployed honeypot virtual machines across the Sydney regions of three major cloud providers: AWS (`ap-southeast-2`), Azure (`australiaeast`), and OVH (`au-syd-1`). This geographic co-location minimized latency discrepancies and enabled unbiased, cross-cloud comparisons of attacker behavior.

Our deception environment addresses several non-trivial, domain-specific challenges unique to healthcare systems. Specifically, it integrates standardized coding systems ICD-10 for diagnoses and NDC for drug identifiers within HL7 message structures to faithfully emulate authentic medical communication. To enhance detection and maintain stealth, we incorporated canary tokens to identify early reconnaissance efforts and employed dynamic data rotation to prevent fingerprinting by adversaries. Leveraging open-source healthcare platforms embedded with honeypot frameworks like T-Pot [20], we captured rich telemetry on attacker interactions. To extract actionable insights, we applied advanced machine learning and co-regularized spectral clustering techniques, uncovering nuanced behavioral patterns, protocol-specific abuse vectors, and regional differences in attack strategies.

Our study is guided by the following key research questions:

- **RQ1:** How can deception-based analysis reveal healthcare-specific attack patterns and vulnerabilities that are typically missed by generic honeypots?
- **RQ2:** What regional and protocol-level variations exist in attacker tactics, and how do they manifest across cloud-hosted deception networks?
- **RQ3:** What trends and insights can be uncovered from malware and hash propagation across different cloud providers?

By addressing these questions, we aim to advance the understanding of attacker behavior in the healthcare domain and demonstrate the value of domain-specific deception strategies in supporting tailored cyber defense.

Our work makes the following key contributions:

- We design and deploy a scalable deception network (cf. §III) for healthcare environments, integrating 30 real-world open-source healthcare applications simulating realistic healthcare workflows by *embedding ICD-10 and NDC codes inside HL7 messages*. The setup was deployed across AWS, Azure, and OVH in Sydney, Australia. In a 30-day run, the network captured 1.2M attacker events, 73% of which hit healthcare-specific endpoints; its Shodan *Honeyscore stayed at 0.0 for the full month*, confirming stealth. Cross-cloud analysis reveals, for example, that OVH receives 3.9× more botnet traffic than AWS, while AWS attracts a TLS-encrypted ransomware pattern absent on Azure (§VII).
- We evaluate three specialized masked-language models ATTACK-BERT [21], SecRoBERTa [22], and SentSecBERT [23] tuned for security domain. We found that the ATTACK-BERT achieved the highest clustering utility (Silhouette = 0.72, Normalized Mutual Information (NMI) = 0.68) and the best downstream F1-score (0.83), delivering a +13% improvement over the best model §V).
- To gain behavioral insight of the malicious actors, we employ Multi-view spectral clustering (§V–§VII). In particular, we combine four complementary *views*—semantic, temporal, numerical, and categorical feature spaces with co-regularized spectral clustering ($\lambda = 1.0$, $K = 8$). As a result, we uncover eight key attack patterns. Moreover, this fused approach increases Adjusted Mutual Information by +37% over the strongest single-view baseline. It exposes patterns, such as an early morning automated scan campaign against EMR and EHR systems, and a TLS-encrypted ransomware staging pattern targeting specific clouds.

Artifact availability. All Docker build configurations and analysis scripts used in this study have been open-sourced under the Apache 2.0 license [24]. While the complete dataset is not publicly available due to sensitivity concerns, it can be provided to the research community upon request.

II. Background and Related Work

Previous studies have explored various techniques for investigating attacker behavior in distributed, networked applications. However, measuring and analyzing how attackers specifically target healthcare infrastructures remains challenging due to issues such as scalability, real-time analysis, and the correlation of attacker activities with network-layer observations and device context. We categorize these approaches into three main areas: deception-based defenses and honeypots, integrated ASM with honeypots, and honeypot detection

prevention. The *first* category of techniques lures attackers to interact with the system, enabling researchers to analyze their behavior and strategies. The *second* category focuses on monitoring and analyzing attacker activities relevant to specific organizational contexts, offering tailored insights for improved defense. The *last one* aims to protect honeypots from being detected by attackers, ensuring the integrity and effectiveness of the deception mechanisms.

Deception-Based Defenses and Honeypots. Deception-based defenses like honeypots are widely used to analyze attacker tactics and techniques [25]. Several studies have deployed honeypots to analyze vulnerabilities in applications, providing invaluable insights into attacker behavior. These honeypots have been implemented in diverse environments, including generic cloud instances [26], [27], educational networks [28]–[32], and network telescopes [33]–[36].

Commodity honeyfarms (e.g. Cowrie, Dionaea) mainly capture ambient SSH/SMB scans; §VF quantifies how attacker behaviour shifts when electronic-health-record (EHR) workflows are present.

ASM Integration with Honeypots. ASM has become a critical framework in modern cybersecurity, especially as the attack surfaces of organizations continue to expand due to cloud adoption. Zhang et al. [15] introduced network attack surface mapping as a method for identifying potential entry points for attackers, emphasizing continuous monitoring and adaptive defenses. In traditional ASM, the focus is often on passive monitoring and identifying attack surfaces through static analysis of network configurations, software vulnerabilities, and external-facing services [16], [17]. A related approach is presented in [37], where a honeynet architecture utilizing containers is proposed for generalized cyber deception. While their work focuses on synthetic environments and broad attack simulations, it does not specifically engage with sector-specific vulnerabilities or protocols. In contrast, our approach advances this concept by deploying a deceptive network comprising real vulnerable web applications targeting healthcare-specific threats.

Honeypot Detection. The effectiveness of honeypots relies heavily on their ability to remain undetected by adversaries. If attackers can easily identify honeypots, they may avoid interacting with them, reducing the honeypot's ability to capture malicious activities. Several studies have explored techniques to detect honeypots, focusing on how attackers identify such environments by exploiting protocol deviations, system responses, or virtualized settings [38], [39]. For example, [40] demonstrated how attackers use simple signatures to detect widely deployed open-source honeypots through services like Censys [41] and Shodan [42].

Vetterl et al., [39] revealed systematic methods to fingerprint medium-interaction honeypots by analyzing transport layer deviations in protocols like SSH and Telnet. These tools allow adversaries to detect honeypots without completing protocol handshakes, significantly reducing the honeypot's value. Furthermore, services like Shodan's "Honeyscore" provide estimates of an IP address's likelihood of being a honeypot, offering attackers a straightforward way to avoid detection [42]. Given the ongoing arms race between honeypot developers and attackers, it is crucial to continuously improve honeypot stealth, reduce detectable patterns, and make them harder to distinguish from legitimate systems.

III. Motivation and Design Requirements

Designing effective deception systems for healthcare environments necessitates addressing a range of domain-specific challenges–including complex clinical workflows, strict regulatory requirements, and increasingly sophisticated cyber threats. These challenges are further compounded by the sector's reliance on standards such as the HL7 messaging protocol [6], and coding schemes like ICD-10 [4] and NDC [5]. While existing deception systems have shown success in general-purpose settings, they frequently fall short in replicating the nuanced operational and technical characteristics of healthcare infrastructures. Generic honeypots, for instance, lack HL7 messages populated with ICD-10 and NDC codes, a combination that is essential to emulate real-world healthcare workflows. HL7 facilitates clinical data exchange through structured messages like Admission-Discharge-Transfer (ADT) (e.g., MSH...PID...). A simplified example of an ADT message is shown in Listing 1.

Listing 1: Example HL7 ADTˆA01 message segment containing a patient's data.

```
MSH|^~\&|HOSPITAL_A|EHR|20250120||ADT^A01|123|
    ↪ P|2.4PID|1|987654321||DOE^JOHN
    ↪ |19850515|M
```

This message indicates that a patient named John Doe was admitted to a hospital on January 20, 2025. While essential for interoperability, such messages are often transmitted over unencrypted networks, making them vulnerable to exploitation during reconnaissance or lateral movement phases of attacks.

In addition to messaging standards, the use of structured medical codes is central to healthcare operations. ICD-10 codes, such as `E11.9`, represent diagnoses (e.g., "Type 2 diabetes mellitus without complications"), while NDC codes, like `0781-1842-10`, identify specific pharmaceuticals (e.g., a bottle of 100 furosemide 40 mg tablets). These codified standards not only streamline clinical workflows but also define predictable, high-value targets in underprotected systems.

Conventional deception systems often simulate static or generic environments and fail to scale dynamically. Many lack the fidelity required to deceive adversaries familiar with healthcare-specific technologies. Moreover, such systems frequently rely on low-interaction traps, which may detect superficial scans but fail to engage attackers employing advanced techniques such as lateral movement or targeted data exfiltration.

Rationale for Modeling HL7. In designing our deception network, we chose to emulate HL7 v2.x over FHIR due to its visibility at the network layer and its broader real-world usage. HL7 v2.x messages are typically transmitted in plaintext using MLLP over TCP port 2575, making them fully inspectable

by packet-level sensors such as Suricata. In contrast, FHIR is usually encapsulated in HTTPS and protected by OAuth, rendering payloads opaque to standard network monitoring tools.

Adoption statistics further support our choice: over 90% of U.S. Health Information Exchange (HIE) organizations routinely use HL7 v2.x, compared to only about 20% who have adopted FHIR [7], [8]. Moreover, HL7 v2.x has been the target of real-world attacks, making it a relevant and high-value deception surface. For instance, ICSMA-21-007-01 documents a segment injection vulnerability in patient monitors, while CVE-2023-43208 describes an unauthenticated RCE flaw in Mirth Connect, a widely used HL7 integration engine [43], [44]. These risks show the practicality of modeling HL7 v2.x in deception-based security infrastructures.

Emulating a Realistic Healthcare Environment. A natural question arises: how closely does our deception network resemble a professional healthcare environment? In practice, modern hospitals integrate subsystems for *patient registration, clinical order management, billing, pharmacy, laboratory services*, and *message brokering*. In high-income countries, such capabilities are dominated by proprietary EHR platforms, most notably Epic and Oracle Cerner, which serve approximately 42% and 23% of U.S. acute-care hospitals, respectively, and together account for more than half of all inpatient beds [45]. However, these platforms require licensing and deployment costs that range from $5–10M for a 300-bed facility, and can exceed $500M for large health systems [46]. Furthermore, contractual obligations often prohibit active security testing, making them unsuitable for use in deception networks.

To overcome these constraints, we constructed a legally shareable and technically faithful alternative by mapping real hospital workflows to open-source software analogues with known vulnerabilities. Our stack includes OpenMRS (used in over 8,100 sites across 80 countries, with 22 million patients [47]), OpenEMR (used by over 100,000 providers with over 200 million patient records [48]), Mirth Connect (with over 10,000 installations [49]), and 27 other applications. As summarized in Table V (in Appendix), this 30-application stack reproduces more than 70% of the functional surface found in professional healthcare settings while remaining freely redistributable for security research.

Design Requirements. Our proposed deception network addresses four core requirements that underpin effective emulation and resilience against advanced threats:

- *R1: Scalability.* The network must operate seamlessly across heterogeneous cloud environments, dynamically adapting to the evolution of the attack surface and ensuring continued functionality in diverse deployments.
- *R2: Fidelity to Healthcare Workflows.* To realistically mimic clinical operations, the network must generate HL7 messages populated with valid ICD-10 and NDC codes. It must also simulate common vulnerabilities in systems such as registration portals and billing interfaces.
- *R3: Real-Time Threat Analysis.* The system should support real-time collection and processing of attack data, enabling rapid identification and analysis of emerging threats.
- *R4: Stealth and Evasion.* To resist detection by advanced reconnaissance tools (e.g., Shodan), the deception network must implement techniques such as dynamic data rotation and regular validation of exposed services to maintain operational stealth.

These requirements form the foundation of our system architecture (cf. §IV), enabling the construction of a realistic, scalable, and interactive deception environment that both attracts sophisticated adversaries and yields actionable insights into attacker behavior within healthcare domains.

IV. Deception Network Design

In this section, we present our end-to-end methodology and the architecture of our deception network, developed to analyze attacker behavior in healthcare-specific multi-cloud environments. Figure 1 illustrates the architecture of our deception network. It comprises three interconnected layers: the *Application Layer*, the *Data Collection and Analysis Layer*, and the *Monitoring and Visualization Layer*. Each layer is specifically designed to fulfill distinct roles, ensuring the system is scalable, interactive, and capable of providing real-time insights while remaining undetectable to attackers [28].

In the following, we explain how this architecture directly addresses the core design requirements (cf. §III) for effective emulation and resilience against advanced threats.

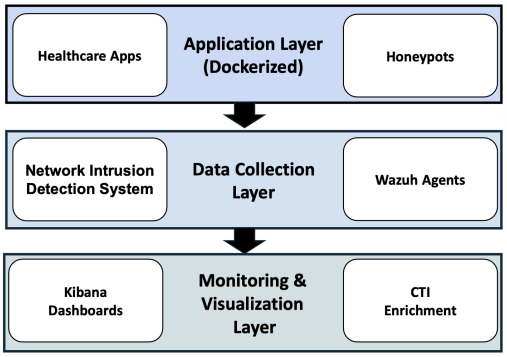


Fig. 1: Architecture of the proposed deception network, highlighting the layered structure and key components.

A. Application Layer: Emulating Real-World Workflows

As a foundational layer of our deception network, we simulated realistic healthcare workflows to attract attackers targeting domain-specific vulnerabilities. To achieve the scalability requirement (R1), we deployed 30 open-source healthcare web applications, each as an isolated Docker container, to replicate critical functions such as patient registration, billing, and diagnostics. We selected these applications based on high-severity vulnerabilities reported in the National Vulnerability Database (NVD) [50] and the CVE repository [51]. The chosen vulnerabilities reflect common healthcare-specific attack vectors, including SQL injection in patient registration portals, cross-site scripting (XSS) in billing systems, and cross-site request forgery (CSRF) in scheduling interfaces creating a realistic and attractive attack surface for malicious actors.

Table V in the Appendix provides details of the selected applications, ensuring a diverse and realistic foundation for potential attacks within healthcare environments.

For the fidelity requirement (R2) of our deception network, we generated synthetic data using Python's Faker library [52], which we extended with healthcare-specific standards to reflect domain-relevant structures and semantics. We synthesized patient demographics, diagnostic records, and billing metadata with internal logical consistency and stored them in MariaDB databases. To ensure clinical plausibility, we embedded structured medical codes such as ICD–10 and NDC identifiers directly into HL7 ADT messages.

We programmatically linked patient records across diagnostic and billing components to maintain coherence throughout the simulated workflow. We then integrated these enriched datasets into containerized healthcare applications deployed across three major cloud providers: AWS, Azure, and OVH. We dynamically generated HL7 ADT messages with randomized patient identifiers, timestamps, and control numbers to emulate authentic clinical communications. These messages were processed through RESTful endpoints, ensuring that the underlying databases reflected realistic healthcare operations and supported the fidelity of our deception environment.

For stealth and evasion requirement (R4), we enabled reconnaissance detection through the strategic deployment of Canary tokens [53], which we embedded in sensitive fields within the synthetic datasets, such as URLs in patient portals, metadata in downloadable files, and fake credentials within application forms. When accessed, these tokens silently logged attacker interactions without raising suspicion, allowing us to gain visibility into their targeting strategies. For example, a Canary token embedded in a patient registration URL successfully identified malicious IP addresses scanning for vulnerable endpoints. We integrated these alerts into our monitoring framework to support real-time analysis.

By combining intentionally vulnerable web applications, synthetic data aligned with healthcare standards, dynamic workflow simulations, and monitoring mechanisms such as Canary tokens, we constructed an application-layer environment with high fidelity. This design ensures that attackers engage with realistic scenarios, enabling us to collect actionable intelligence on domain-specific attack behaviors.

B. Data Collection Layer

We gathered, processed, and enriched logs generated by the Application Layer to enable structured, high-fidelity analysis of attacker behaviors. We deployed a suite of open-source tools and designed a custom pipeline to ensure comprehensive coverage and meaningful threat intelligence extraction.

Data Collection. We began by configuring Wazuh [54], an open-source security monitoring platform, to collect logs from our healthcare web applications, honeypots, and supporting infrastructure. Wazuh enabled us to detect threats, monitor file integrity, and manage compliance. We mapped observed attacker activities–such as reconnaissance, brute-force attempts, and exploitation techniques to the MITRE ATT&CK framework [55], allowing us to generate structured threat intelligence anchored in tactics, techniques, and procedures (TTPs), such as T1110 (brute force).

To monitor network activity within our containerized environment, we employed Suricata [56], a high-performance NIDS/IPS, configured with `AF_PACKET` for efficient packet processing. We developed custom Suricata rule sets tailored to healthcare environments using sources such as Emerging Threats Pro [57]. These rules were designed to detect healthcare-relevant behaviors including DNS tunneling, TLS-based reconnaissance, and data exfiltration. We iteratively tuned and validated the rules to minimize false positives while preserving detection accuracy, following established best practices [58], [59].

We used Logstash [60], part of the ELK stack [61], to normalize and parse logs from both Wazuh and Suricata. Our pipelines extracted essential features such as source and destination IPs, ports, timestamps, and payload metadata, standardizing them into a unified schema. We then enriched these logs by appending contextual information, including GeoIP locations, Autonomous System Numbers (ASN), and Common Vulnerabilities and Exposures (CVE) identifiers. This enrichment allowed us to derive deeper insights into attacker infrastructure, origin, and targeted vulnerabilities.

Filtering and Enrichment. To reduce noise, we filtered out benign traffic using metadata tags and threat intelligence from IP reputation databases such as GreyNoise [62]. This allowed us to focus our analysis on confirmed malicious activity. Enriched logs were retained for deeper inspection and clustering. We preprocessed raw logs to ensure consistency and reliability. We normalized timestamps, flattened nested data structures, and removed redundant entries. Protocol-specific fields were standardized across records to support efficient enrichment and downstream clustering.

Addressing the scalability (R1) and real-timeliness requirements (R2) of our deception network, we integrated Wazuh and Suricata by aligning architectural differences between their respective backends. Specifically, while our T-Pot [20] honeypot environment utilized the ELK stack for log management, Wazuh operated on an OpenSearch backend. To reconcile these differences, we customized Logstash pipelines to unify the schemas and merge the two data streams into a cohesive dataset. This integration enables real-time analysis at scale and ensures that our data collection framework remains scalable and coherent, effectively supporting deployment across expanding and heterogeneous environments.

Clustering Analysis. We employed ATTACK-BERT [21] to generate semantic embeddings from textual and categorical fields in the enriched logs. These embeddings captured nuanced relationships between attacker behaviors and system attributes. We then applied multi-view spectral clustering to the embedded data, which allowed us to identify protocol-specific abuses, geographic and regional targeting variations, and evidence of coordinated or campaign-driven attacks.

C. Monitoring and Visualization Layer

The Monitoring and Visualization Layer enables real-time tracking of attacker behavior while preserving the stealth and operational integrity of the deception network. This layer directly addresses the design requirements of *R3: Real-time Analysis* and *R4: Stealth*.

The enriched logs of the Data Collection Layer were indexed using Elasticsearch [61] and visualized using Kibana [63]. We developed interactive dashboards to monitor critical metrics such as attack frequency, protocol usage, and geographic origin of threats. For instance, SQL injection attempts against HL7 endpoints were traced back to attacker IP addresses and regions, providing immediate, actionable insights to inform defensive responses.

To meet the stealth requirement, we periodically regenerated synthetic patient records using the Faker library, introducing continuous variability into the dataset. This dynamic content reduced the risk of honeypot fingerprinting by adversaries. In addition, outbound traffic from the deception environment was tightly regulated. We enforced network policies that included IP and port whitelisting, monitoring of egress traffic, and blocking of unauthorized connections. These safeguards prevented the infrastructure from being repurposed for malicious operations and ensured that attacker interactions remained confined and observable. By combining real-time data processing with stealth-preserving mechanisms, this layer transforms raw attack telemetry into operational intelligence while maintaining the credibility and resilience of the deception network.

D. Dataset Overview and Filtering Strategy

To distinguish targeted healthcare-specific threats from global background noise, we deployed a deception network across AWS, Azure, and OVH cloud providers, each hosting 10 public IPs with Dockerized honeypots simulating HL7 endpoints, patient registration systems, and billing portals. The network was active for four weeks (Jul 25–Aug 25, 2024), and captured both passive scans and active attacker interactions using Suricata in IDS mode. Overall, our deception network captured 1,266.85 GB of traffic originating from 16,243 unique IP addresses, 913 autonomous systems (ASNs), and 153 countries. A summary of the dataset is provided in Table I.

TABLE I: Summary statistics of the captured dataset.

Metric	Value
Total records	1,200,000
Total traffic (GB)	1,266.85
Unique public IPs	16,243
Unique ASNs	913
Unique countries	153

To filter out non-targeted noise, we compared our dataset against the Orion Network Telescope [64], which monitors unsolicited traffic across the global IPv4 address space. This comparison allowed us to exclude IPs associated with indiscriminate scanning, while enriching our analysis of unique, localized behavior.

Cross-Telescope Comparison. Out of 16,243 source IPs interacting with our honeypots, 6.5% were also observed by Orion, indicating they were part of globally visible background scanning activity. Conversely, 93.5% were unique to our deception environment, suggesting targeted or localized scanning behavior. Breakdown by cloud provider revealed significant differences: AWS instances received traffic with only 0.86% IP overlap with Orion, Azure showed 18.7%, and OVH had 80.4%. This highlights AWS's potential in attracting more targeted interactions, whereas OVH was more prone to botnet-driven bulk scanning.

Port-Level Patterns. We further compared port activity overlap with Orion. For AWS, only 11.3% of targeted ports were seen in both datasets, compared to 32% for Azure and 100% for OVH. The complete overlap on OVH indicates that traffic to those nodes was predominantly generic and botnet-driven. In contrast, AWS traffic demonstrated a preference for healthcare-specific services not visible in Orion data.

Filtering Heuristics. Guided by insights from Orion [64], we refined the dataset to focus on meaningful attacker behaviors. We excluded IP addresses flagged as benign scanners or linked to reputable organizations such as Palo Alto Networks [65], Censys [41], and Shodan [66] to minimize background noise. However, we retained traffic associated with known botnets, including Mirai, as it reflected ongoing malicious campaigns relevant to the analysis. To ensure alignment with the study's healthcare focus, we prioritized interactions targeting healthcare-specific endpoints such as HL7 systems and medical billing portals.

E. Limitations and Ethical Considerations

Limitations. While our deception-based approach provides valuable insights into attacker behavior within healthcare environments, several limitations remain. *First*, we limited our deployment to one month, which restricts our ability to observe long-term attacker trends or seasonal variations. A longer observation period could uncover persistent threats and adaptive behaviors. Although we deployed realistic synthetic healthcare data across multiple cloud platforms, we did not validate our results against real-world incident data. This may affect the generalizability of our findings. In future work, we plan to collaborate with healthcare providers to incorporate operational logs for validation.

Note that we generated synthetic workflows using Python's Faker library and enriched them with healthcare standards such as ICD-10 and HL7. However, these simulations may not capture the full complexity and variability of actual EHR systems, which could influence how attackers engage with the environment. We assessed stealth solely through Shodan's Honeyscore. While this provided an initial indication of our honeypots' detectability, it does not account for adversaries using custom heuristics or alternative reconnaissance tools. In future iterations, we aim to incorporate broader stealth evaluations and emulate live traffic more closely. Also, our focus was on system design and behavioral analysis, not operational incident reporting. As such, we did not examine

runtime challenges or deploy system-level mitigations. We intend to address these aspects in future work involving production environments.

We did not include a control group of non-healthcare honeypots, which limits our ability to conclusively attribute certain behaviors to healthcare-specific targeting. Although interactions with domain-specific services like HL7 endpoints and billing portals suggest focused targeting, parallel deployment of non-healthcare decoys would allow more precise differentiation between general and domain-specific attacks, a direction we plan to pursue.

Finally, our study enumerates vulnerabilities from authoritative databases [50] [51] and public PoCs [67] [68] rather than re-validating each one experimentally. While typical for large-scale field studies [69] [70], this approach may leave a small residual of mis-classified CVEs, which future work could revisit under controlled lab conditions.

Ethical Considerations. Our research does not involve human subjects or identifiable data, and thus does not require Human Research Ethics Committee (HREC) approval. Nonetheless, we followed strict ethical practices in deploying honeypots. To minimize any potential harm, we have taken several precautions to ensure that the honeypots do not inadvertently expose or contribute to malicious activities. Outbound traffic was controlled via a web application firewall (WAF) to prevent misuse, particularly DNS amplification. Additionally, UDP traffic was blocked to avoid potential DDoS abuse. Continuous monitoring via Wazuh [54]–a SIEM tool–ensured the honeypots remained uncompromised, aligning with ethical standards in cybersecurity to minimize harm and responsibly gather insights into attacker behavior.

V. Clustering Methodology

Analyzing attacker behavior within noisy high-dimensional datasets, particularly those derived from real-time deception networks, requires techniques capable of uncovering hidden structures and relationships that are not readily apparent through rule-based or signature-driven analysis. Traditional methods [71] often fail to identify previously unseen or evolving attack strategies, which are increasingly prevalent in integrated healthcare threats [72].

To address this challenge, we adopted an unsupervised learning approach centered on spectral clustering. This allowed us to explore the data (cf. §III) holistically, revealing protocol-specific abuses, coordinated attack campaigns, and regional variations in attacker behavior. By integrating semantics, temporal, numerical, and categorical features through multi-view spectral clustering [73], we are able to capture rich, multi-dimensional patterns across attacker interactions, aligning with our overarching goal of understanding attacker tactics in realistic, domain-specific settings.

The effectiveness of this approach lies on the careful feature design and representations. We engineered features to represent multiple dimensions of attacker behavior, including the semantic intent of payloads and signatures, temporal trends of activities, numerical indicators such as engagement persistence, and categorical attributes such as geographical origin or affiliated autonomous system number or organization. These engineered features form the basis of our multi-view clustering pipeline, as detailed in the following subsections.

A. Feature Engineering and Preprocessing

To ensure robust and effective clustering, we applied a comprehensive feature engineering strategy that transformed textual, categorical, temporal, and numerical data into 768-dim vector embeddings, one-hot binaries, cyclical (sine/cosine) time features, and normalized real-valued metrics respectively suitable for multi-view spectral clustering [73]. A key focus was on the quality of semantic representations, which play an important role in capturing the intent behind attacker interactions.

Selecting a transformer model for feature embedding. Given our dataset comprises both semantic (i.e., Suricata ‘flow_signature‘ and ‘http.request_body‘) and categorical fields requiring embedding for clustering, traditional approaches such as TF–IDF fail to capture contextual semantics and inter-token dependencies. To evaluate state-of-the-art transformer encoders, we tokenized each flow's signature and payload (when present), concatenated the resulting tokens (up to 128 per flow), and passed 1.2 million sequences through three pretrained models ATTACK-BERT, SecRoBERTa, and SentSecBERT without further fine-tuning. Using our curated dataset (cf. §III), we compared each model based on clustering-relevant metrics: Silhouette Score, Normalized Mutual Information (NMI), and F1-score on a leave-one-flow-out classification task. As shown in Table II, ATTACK-BERT outperformed the alternatives, achieving a Silhouette Score of 0.72, NMI of 0.68, and an F1-score of 0.83. These results validate its capacity to encode nuanced semantic relationships in cybersecurity telemetry, making it our preferred embedding model for subsequent clustering analyses.

TABLE II: Comparison of embedding models for semantic feature representation.

Model	Silhouette Score	NMI	F1-Score
ATTACK-BERT [21]	**0.72**	**0.68**	**0.83**
SecRoBERTa [22]	0.64	0.59	0.79
SentSecBERT [23]	0.58	0.52	0.76

Features representations. We used ATTACK-BERT to generate dense vector embeddings of semantic features–attack signatures and payloads. This transformer-based model applies a bidirectional self-attention mechanism to capture contextual token dependencies, enabling it to group semantically similar attacks even when textual descriptions differ. For instance, SQL injection attempts targeting both patient registration and billing systems were clustered together due to their shared intent. We extracted 768-dimensional embeddings from the final hidden layer using mean pooling across tokens. To balance computational cost with representational fidelity, we set the maximum sequence length to 128 and used a batch size of 32. We validated the quality of these embeddings

through cosine similarity heatmaps and t-SNE visualizations (Appendix Figure 8a), which confirmed that semantically related attacks formed distinct, interpretable clusters.

In parallel, we extracted categorical features such as `src_ip_asn_org` and `src_ip_country` to capture geographic and organizational contexts. We applied one-hot encoding to convert these into binary representations suitable for similarity calculations using Hamming distance. This approach preserved categorical distinctions while allowing for effective integration into our clustering pipeline.

For temporal features such as `timestamp` and `time_of_day`, we used sine and cosine encoding to preserve their cyclical nature. We computed pairwise similarities using Euclidean distance to detect temporal clustering trends, such as diurnal attack patterns or coordinated bursts of malicious activity. Finally, we incorporated numerical features that offered quantitative insights into attacker behavior. Metrics such as protocol entropy, geo-distance, and activity duration captured the breadth, geographic scope, and persistence of attacker campaigns. We applied Min-Max normalization to these features to ensure consistent scaling across dimensions and prevent bias during clustering.

B. Spectral Clustering with Normalized Cut

Spectral clustering is well-suited to cybersecurity datasets, particularly in deception-based environments, due to its ability to capture complex, non-linear relationships through graph-based representations. By modeling data points as nodes and their pairwise similarities as weighted edges, we are able to extract latent structural patterns across attacker behaviors. This representation enables the identification of semantically, temporally, and behaviorally similar attack vectors that may elude traditional clustering methods.

To partition this similarity graph effectively, we employed the *Normalized Cut* (Ncut) criterion, which optimizes cluster separation by minimizing the similarity between different clusters while maximizing intra-cluster coherence. This formulation ensures well-defined, behaviorally consistent groupings, enhancing interpretability and insight extraction.

a) Normalized Cut Criterion.: Formally, the normalized cut objective is defined as:

$$\text{Ncut}(A, B) = \frac{\text{Cut}(A, B)}{\text{Assoc}(A, V)} + \frac{\text{Cut}(A, B)}{\text{Assoc}(B, V)}, \quad (1)$$

where:

- $\text{Cut}(A, B) = \sum_{i \in A,\, j \in B} w_{ij}$ quantifies the total edge weight between clusters A and B.
- $\text{Assoc}(A, V) = \sum_{i \in A,\, j \in V} w_{ij}$ measures the total connection of cluster A to all nodes in the graph.

This objective penalizes cluster splits that sever strong internal connections, promoting well-separated and internally cohesive clusters. In our context, this captures behavioral divergence among attackers targeting different vulnerabilities, protocols, or services.

b) Graph Construction and Embedding.: We constructed the similarity graph using a Gaussian kernel:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \quad (2)$$

where x_i and x_j are multi-dimensional feature vectors (e.g., semantic, temporal, or numerical), and σ controls the neighborhood sensitivity.

The unnormalized graph Laplacian $L = D - W$, where D is the diagonal degree matrix, was transformed into the normalized form:

$$L_{\text{norm}} = D^{-1/2} L D^{-1/2}. \quad (3)$$

We then extracted the first k eigenvectors of L_{norm} to produce spectral embeddings, which projected the data into a lower-dimensional subspace suitable for clustering via k-means.

C. Nyström Approximation for Large-Scale Data

Given the high dimensionality and scale of our similarity matrices, computing the full eigendecomposition of the graph Laplacian was computationally infeasible. To address this, we adopted the Nyström approximation, which reduces complexity by sampling a representative subset $\mathcal{S}$ of the data.

The similarity matrix was partitioned as:

$$W = \begin{pmatrix} W_{\mathcal{S},\mathcal{S}} & W_{\mathcal{S},\mathcal{S}^c} \\ W_{\mathcal{S}^c,\mathcal{S}} & W_{\mathcal{S}^c,\mathcal{S}^c} \end{pmatrix},$$

where $W_{\mathcal{S},\mathcal{S}}$ denotes similarities within the sampled subset, and $W_{\mathcal{S}^c,\mathcal{S}}$ captures similarities between sampled and non-sampled data points.

Using this decomposition, we approximated the spectral embedding for the entire dataset via:

$$\tilde{U} = W_{\mathcal{S}^c,\mathcal{S}} \cdot U \cdot \Lambda^{-1/2},$$

where U and Λ are the eigenvectors and eigenvalues derived from $W_{\mathcal{S},\mathcal{S}}$. This approximation significantly reduced computational overhead while preserving clustering accuracy.

D. Co-Regularized Multi-View Spectral Clustering

In multi-dimensional cybersecurity datasets, distinct feature types such as semantic embeddings, categorical encodings, and numerical metrics offer complementary perspectives of attacker behavior. To leverage these multiple modalities, we adopted a co-regularized multi-view spectral clustering framework.

Each feature subset was encoded into its own similarity matrix $W^{(v)}$ and corresponding Laplacian $L^{(v)}$. We jointly optimized the spectral embeddings across all m views via the objective:

$$\min_{\{U^{(v)}\}} \sum_{v=1}^{m} \text{Tr}(U^{(v)\top} L^{(v)} U^{(v)}) + \lambda \sum_{v<w} \|U^{(v)} - U^{(w)}\|^2, \quad (4)$$

where $U^{(v)}$ denotes the spectral embedding for view v, and λ is a regularization term that encourages consensus across views.

The first term minimizes the normalized cut in each feature view independently, preserving intra-view structure, while the second term enforces alignment across embeddings, facilitating unified clustering. This approach enabled us to integrate semantically rich but structurally diverse representations into a single, interpretable clustering space.

E. Single-View vs. Multi-View Clustering.

Understanding attacker behavior in cybersecurity datasets necessitates clustering methods that account for the inherently multi-dimensional nature of the data. While single-view clustering approaches like those based solely on semantic, numerical, or categorical features can uncover meaningful patterns within individual feature spaces, they often fail to capture relationships that span multiple dimensions.

In contrast, multi-view clustering integrates diverse feature representations into a unified analysis framework. This enables the discovery of cross-dimensional correlations such as semantic similarities that align with geographic patterns or temporal signatures that would be invisible in isolation. Our co-regularized spectral clustering approach leverages this integration to enhance both the granularity and interpretability of the resulting clusters. By aligning insights from distinct views, we obtain a holistic understanding of adversarial tactics and campaign structures in healthcare-specific attack traffic.

VI. Clustering Performance Analysis

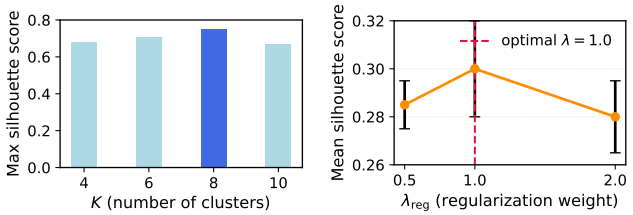
In this section, we present an evaluation of our clustering framework through sensitivity analysis, parameter optimization, and performance benchmarking across multiple feature dimensions. We assess both single-view and multi-view clustering schemes using established metrics to quantify their effectiveness in capturing meaningful attacker behavior patterns.

A. Parameter Sensitivity and Optimization.

Clustering performance in this framework depends critically on two hyperparameters: the number of clusters K and the regularization weight λ. The parameter λ mediates the trade-off between view-specific fidelity and consensus alignment. Smaller values favor independent view structure, while larger values prioritize convergence toward a shared representation.

To determine optimal values, we conducted a grid search over K and λ, evaluating each setting using the silhouette score. We observed that $\lambda = 1.0$ yielded the best trade-off between intra-view coherence and cross-view agreement, and that $K = 8$ maximized clustering quality, reflecting the diversity of attacker strategies observed across the dataset.

To identify optimal clustering parameters, we conducted a sensitivity analysis over a range of regularization weights $\lambda \in \{0.1, 0.5, 1.0, 1.5, 2.0\}$, evaluating clustering performance using the mean silhouette score. As shown in Figure 2, we observed that $\lambda = 1.0$ yielded the highest silhouette score (0.32), indicating the best trade-off between intra-cluster cohesion and inter-cluster separation across all feature views. This value was therefore selected for the final clustering configuration.



(a) Silhouette score vs. K. (b) Silhouette score vs. λ.

Fig. 2: Parameter tuning results for co-regularized multi-view spectral clustering.

Beyond this point, further increases in λ led to diminishing returns or slight reductions in performance, likely due to over-regularization, which can suppress important view-specific patterns in attacker behavior. In addition to tuning λ, we explored different values for the number of clusters K. As illustrated in Figure 2a, a value of $K = 8$ achieved the highest silhouette score, suggesting that eight clusters best captured the diversity and granularity of attacker strategies observed in our deception environment. By evaluating the influence of both λ and K on clustering performance, we ensured that the resulting consensus clusters were both robust and interpretable. These empirical results provide a principled basis for parameter selection and reinforce the value of the co-regularized multi-view spectral clustering framework in modeling complex attacker behavior.

B. Clustering Effectiveness Across Feature Views

We evaluated the performance of our clustering approaches using two complementary metrics: Silhouette score and Adjusted Mutual Information (AMI). The Silhouette score measures both intra-cluster cohesion and inter-cluster separation, with values ranging from -1 (poor clustering) to 1 (well-separated and compact clusters). While high Silhouette scores indicate strong structural quality within a single feature space, they do not account for alignment across heterogeneous data types. In contrast, AMI quantifies the agreement between clustering outcomes, normalized against random chance, and ranges from 0 (no agreement) to 1 (perfect agreement). In particular, AMI is useful for comparing single-view and multi-view clustering results to measure the added value of integrating multiple feature perspectives. Table III summarizes the performance of our single-view and multi-view clustering approaches, evaluated using the aforementioned metrics.

Single-View clustering results. Among the individual feature views, we found that clustering in the semantic space using ATTACK-BERT embeddings achieved the highest Silhouette score (0.97), indicating well-defined and tightly grouped semantic clusters. However, this approach was limited in capturing behavioral diversity expressed through other dimensions or features such as attack timing or IP-level engagement intensity. Numerical feature clustering yielded a Silhouette score of 0.88, successfully surfacing broad attack volume and trends (e.g., protocol usage or geo-distribution), but lacking contextual nuance. The clustering of temporal features achieved a lower Silhouette score 0.64, highlighting diurnal patterns, but

failed to capture the volumetric aspects of attacks–critical to understanding the coordination of attackers.

Multi-View clustering results. Our co-regularized spectral clustering approach integrated semantic, temporal, numerical, and categorical features into a unified representation. While the resulting Silhouette score 0.75 was lower than the best single-view score 0.97, this reduction reflects the increased complexity and diversity of multi-modal data rather than a degradation in cluster quality. Crucially, the multi-view clustering achieved a substantially higher AMI compared to any pair of single-view results, validating its ability to capture cross-dimensional patterns and offer a more comprehensive understanding of attacker behavior.

The AMI scores presented in Table III quantify the alignment between individual single-view clusterings and the final multi-view consensus. Moderate AMI values (e.g., 0.4025 for semantic, 0.2821 for categorical) indicate that while each single-view captures meaningful structure, none independently reconstructs the comprehensive segmentation produced by multi-view clustering. This reinforces the premise that attacker behaviors are inherently multi-dimensional, and that capturing their full complexity requires the integration of complementary perspectives or dimensions–semantic, temporal, numerical, and categorical.

TABLE III: Performance overview of single-view vs. multi-view clustering.

Method	Silhouette Score	AMI (vs. Multi-View)
Single-View (Semantic)	0.97	0.4025
Single-View (Categorical)	0.88	0.2821
Single-View (Temporal)	0.64	0.1957
Single-View (Numerical)	0.72	0.578
Multi-View (Final)	0.75	–

Visual Validation. We assessed the quality of the final embeddings using cosine similarity heatmaps (Appendix Figure 8) and t-SNE visualizations (Figure 3). Both analyses confirmed that semantically related attacks clustered together in well-separated, interpretable groups. The t-SNE plot further illustrates the distinctness of behavioral clusters within the multi-view consensus space, with healthcare-targeted activity forming compact and visually separable clusters from broader, non-specific traffic. These results underscore the value of domain-specific, multi-dimensional analysis in revealing nuanced targeting behaviors in deception-based environments.

Discoveries Enabled by Multi-View Clustering. To further illustrate the advantage of our multi-view clustering approach, we analyzed the patterns and correlations uncovered through this integrated representation. Notably, multi-view clustering exposed attacker behaviors that were obscured or fragmented in single-view clustering analyses.

Temporal-semantic associations. Temporal features revealed strong diurnal attack patterns that aligned with semantic clusters targeting specific CVEs. For instance, attacks clustered by payload similarity (e.g., SQLi or HL7 abuse) also showed in Figure 10) coordinated timing concentrated during early

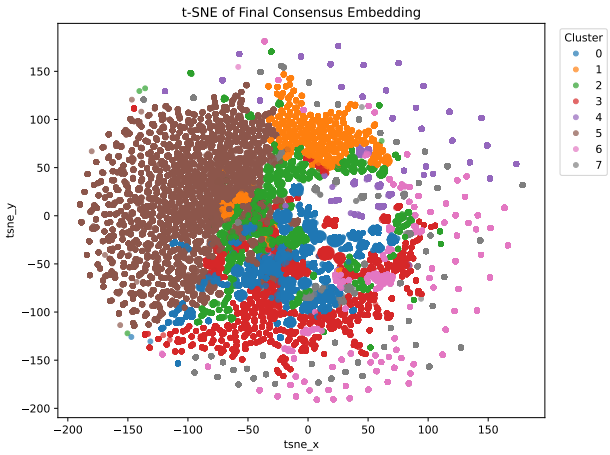


Fig. 3: t-SNE visualization of clusters, illustrating distinct attack behaviors. The different colors show distinct clusters.

morning hours potentially indicating adversaries' strategic scheduling to avoid peak monitoring periods.

Numerical refinement of semantic clusters. Semantic clusters were further refined by incorporating numerical attributes such as IP activity frequency and session durations. This allowed us to distinguish between botnet-driven campaigns, characterized by high-frequency, short-duration bursts, low-volume, persistent reconnaissance activities often indicative of more targeted adversaries.

Cross-dimensional correlations. Multi-view clustering also surfaced correlations between categorical and semantic attributes, such as the geographic origin or ASN of source IPs and their associated payload families. For example, certain clusters tied to specific IP blocks were consistently associated with exploit attempts targeting healthcare systems hosted on particular cloud platforms. These coordinated behaviors spanning cloud infrastructure, region, and exploit vector–would likely remain hidden without a multi-view analytical lens.

Overall, these insights validate the effectiveness of multi-view clustering in resolving cross-dimensional conflicts and producing a richer, more holistic representation of attacker tactics. This approach contextualizes observed attack patterns within real-world operational and regional dynamics, enabling defenders to prioritize mitigation strategies based on informed threat intelligence. In the following section (§VII), we present the key behavioral insights extracted from the resultant clusters in our dataset obtained at our deception networks.

VII. Answering Research Questions

In the following, we answer our research questions (cf. §I), grounded in the empirical findings of our clustering framework.

A. RQ1: Detecting Healthcare-Specific Attack Patterns and Vulnerabilities.

Our analysis identified distinct healthcare-specific attack behavior, particularly in terms of timing and persistence. As shown in Figure 10, several clusters such as cluster 6 and cluster 7 exhibited clear diurnal patterns, peaking during early

morning hours. These bursts likely reflect automated scans exploiting vulnerabilities in services like HL7 interfaces or patient registration portals during operational downtimes.

In contrast, Cluster 2 demonstrated sustained, low-frequency engagement over an extended period, consistent with targeted reconnaissance attempts against healthcare endpoints. This suggests a deliberate probing strategy, possibly aimed at evading threshold-based detection systems.

In addition, Cluster 5 exhibited a distinct spike in activity around 15:00 hours, as shown in Figure 4. Such temporal targeting may coincide with mid-shift transitions in healthcare-environments periods when monitoring is less intensive providing adversaries with a tactical advantage. These patterns diverge from those observed in Cluster 3, which maintained steady activity during off-peak hours, consistent with stealthy enumeration.

Clusters 4, 6, and 7 were dominated by attacks exploiting healthcare endpoints, including HL7 interfaces and billing systems. These clusters demonstrated persistent activity throughout the one-month deployment period, with a notable spike observed on August 16-17 (Figure 4). This spike was linked to a sophisticated attacker leveraging AWS infrastructure for TLS-based attacks, targeting healthcare endpoints such as billing portals. These findings underscore the critical need for defenses tailored to detect and mitigate persistent threats in healthcare systems.

B. RQ2: Regional and Protocol-Specific Variations in Attacker Tactics.

To examine how attacker strategies vary by geography and protocol, we analyzed both the regional origin and technical characteristics of observed threat behaviors. Our clustering analysis revealed significant correlations between geographical attributes, protocol usage, and attacker tactics, underscoring the necessity of contextual, region-aware deception strategies.

We first assessed the relationship between attacker geography and behavioral clustering by conducting a Pearson Chi-Square test on the *country* $\times$ *cluster* and *ASN* $\times$ *cluster* contingency tables. The results showed a statistically significant dependence between region and behavioral cluster:

- **Country** $\times$ **Cluster:** $\chi^2(3936) = 132{,}310$, $p < 0.001$, Cramér's $V = 0.41$ (medium–large effect).
- **ASN** $\times$ **Cluster:** $\chi^2(12) = 77{,}248$, $p < 0.001$, Cramér's $V = 0.31$ (medium effect).

These results suggest that geographical origin and network ownership significantly shape the behavioral profiles of observed attacks. Full contingency matrices are available in our public artifact [24]. For example, as shown in Figure 10, Cluster 7 represented a bursty Mirai-like scanner with /32 sources distributed across the United States (29.5%), China (21.2%), Australia (13.5%), and Brazil (10.9%). This global mix mirrors recent Mirai-variant campaigns like Murdoc and CatDDoS, reported by Qualys and Red Piranha [74], [75], in which attackers spun up ephemeral VPS nodes across these regions. Notably, the strong Australian representation likely reflects the proximity of our honeypots, which were co-located in Sydney, suggesting a preference for region-local probing to reduce latency.

In contrast, Cluster 2 was dominated by single-shot DNS probes, often consisting of a few `TXT/ANY` queries before disengagement. Its leading origin countries included China (31.9%), Australia (15.2%), and Bulgaria (7.0%), consistent with DNS-tunnel abuse previously documented by Akamai in credential-stuffing operations [76]. Together, Clusters 2 and 7 illustrate distinct regional and protocol-specific behaviors *(i)* a globally dispersed Mirai scanner and *(ii)* DNS abuse for covert ex-filtration reinforcing the need for deception systems capable of detecting diverse and transient threats.

C. RQ3: Malware and Hash Analysis Across Cloud Providers.

To further investigate protocol-specific and provider-specific variations, we analyzed malware file hashes observed across AWS, Azure, and OVH honeypots.

Figure 9 shows several malware hashes appeared across multiple providers, indicating shared tool-sets used by attackers targeting cloud-based healthcare systems. Certain hashes (e.g., H60–H80) were heavily concentrated on AWS and OVH, suggesting coordinated campaigns reusing malware families across platforms. Other hashes were exclusive to a single provider, highlighting targeted payload deployment strategies.

We statistically validated the relationship between file hashes and cloud providers using a Chi-Square test: $\chi^2(206) = 4482.04$, $p < 0.001$, Cramér's $V = 0.58$, indicating a moderate to strong association between malware variants and cloud infrastructure hosting healthcare applications [77]–[79].

Table VI presents the top five malware hashes per cloud provider, including associated MITRE ATT&CK techniques and assessed threat levels. Notably, hashes H62 and H63 appeared across all three providers and mapped to tactics such as Command-and-Control (T1071) and Encrypted Channel (T1573.001), underscoring the broad reach of some malware campaigns. Among them was a *LockBit 3.0* variant– an increasingly prevalent ransomware strain in healthcare– alongside multiple *Mirai* variants commonly used for botnet propagation and DDoS attacks.

These results demonstrate attackers' tendency to reuse malware across cloud environments while adapting their tactics to infrastructure-specific characteristics. Consequently, deception-based monitoring systems must account for cross-provider threat dynamics to maintain comprehensive coverage.

VIII. Validation of System Design Requirements

In this section, we empirically validate our design requirements for our deception network and demonstrate its robustness as well as the operational viability in realistic cloud environments.

A. Validation of Design Requirements

We validate the four design requirements (R1–R4), demonstrating how our proposed deception network meets the key objectives of scalability, healthcare specificity, real-time threat detection, and stealth.

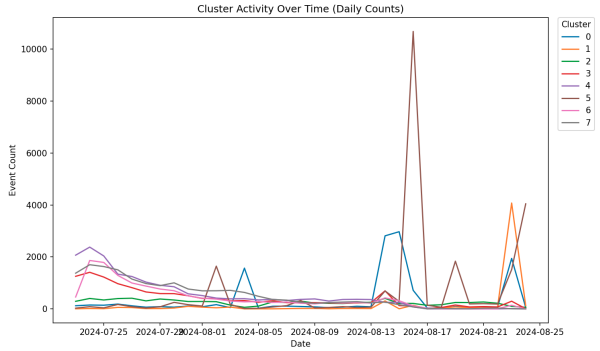
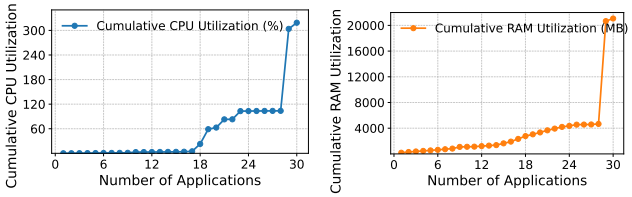


Fig. 4: Temporal analysis across clusters revealing targeted patterns. Daily trends show the spike on August 16-17.

Scalability (R1). To evaluate scalability, we monitored resource utilization as the number of deployed applications increased. Figure 5 reports CPU and RAM usage for 30 deployed services, including resource-intensive components such as T-Pot honeypots. As shown in Figure 5a, CPU utilization scaled linearly with the number of active applications, with a modest increase associated with T-Pot components. Similarly, RAM usage followed a predictable linear trend (Figure 5b), with dynamic resource allocation mitigating potential performance bottlenecks. These results confirm that the architecture supports scalable deployment across heterogeneous environments without compromising performance.



(a) CPU vs. # of applications (b) RAM vs. # of applications

Fig. 5: Validation of scalability via resource monitoring.

Healthcare Specificity (R2). To validate domain relevance, we compared captured attack traffic between our healthcare-specific deception network and a generic honeypot deployment. The deception system recorded 73% more attacks targeting healthcare-relevant endpoints such as HL7 brokers and billing APIs, underscoring its effectiveness in attracting domain-specific threats not observed in generic honeypots.

Real-Time Detection and Responsiveness (R3). We assessed the system's ability to deliver low-latency threat detection across the entire data pipeline. Based on 10,000 randomly sampled network events, the average end-to-end processing latency–from packet capture to visualization in the monitoring dashboard–was 2.3 seconds, with 95% of events processed within 5 seconds. For critical attack vectors, such as HL7 abuse and SQL injection, custom Suricata signatures triggered alerts with an average delay of 1.8 seconds. These metrics validate the system's capability for real-time triage and timely defensive response. Also, under peak conditions (1,500 events / minute), the system sustained sub-5-second latency without data loss, demonstrating resilience under operational stress.

Stealth and Evasion Resistance (R4). To evaluate stealthiness, we used Shodan's Honeyscore [42] to monitor the detectability of our infrastructure over a one-month period. As shown in Figure 6, our deception VMs consistently maintained a Honeyscore of 0, indicating successful evasion from honeypot detection engines. In contrast, traditional honeypots such as Cowrie [80] and Dionaea [81] exhibited increased Honeyscores over time. These findings confirm that our design remains covert and resists fingerprinting by external scanning engines.

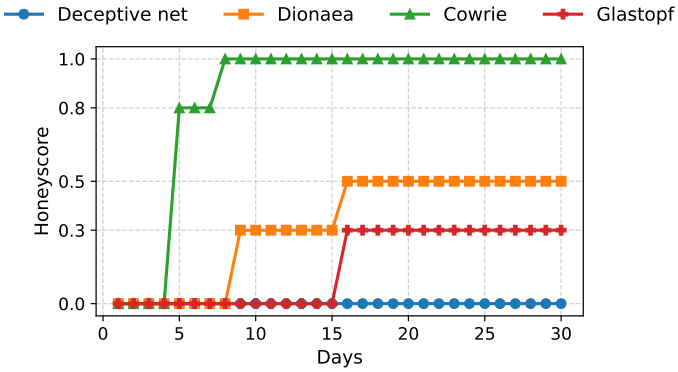


Fig. 6: Stealth validation using Shodan Honeyscore.

B. Ablation Study: Secure vs. Vulnerable Web Applications

To further evaluate attacker adaptability to system configurations, we conducted an ablation study comparing attacker behavior across two web applications, one secure and one deliberately vulnerable, deployed in the same cloud environment (OVH). The secure application was containerized using Docker on a hardened Ubuntu instance [82], [83], served via Gunicorn [84], and enforced security headers including CSP, X-XSS-Protection, and X-Content-Type-Options [85], [86]. Vulnerability assessments using OWASP ZAP [87] and Nikto [88] ensured the absence of known CVEs.

Key Findings. Table IV summarizes the results of our statistical comparison across alert event types, protocols, geolocation, and temporal attributes.

TABLE IV: Chi-square test results for the analyzed secure vs vulnerable web applications.

Feature	Chi-Square	p-value	Cramér's V	Effect Size
Alert Types	222.70	$4.53e{-}18$	0.073	Weak
Event Types	586.57	$8.24e{-}76$	0.096	Weak
Application Protocols	214.74	$1.35e{-}40$	0.071	Weak
Source IP Country	2013.57	0.0	0.220	Moderate
Source IP ASN	1873.29	0.0	0.212	Moderate
Day vs Night Interaction	1979.28	0.0	0.330	Moderate

- *Event Types and Protocols.* The vulnerable system attracted a wider range of attack vectors, including increased DNS tunneling and file manipulation attempts. HTTP and DNS were disproportionately targeted, reflecting the attack surface presented by healthcare-specific services.
- *Geolocation and ASNs.* Attackers interacting with the vulnerable system originated from a broader range of countries and ASNs, including external cloud infrastructures (e.g.,

AWS, Google), whereas the secure application saw more localized OVH-based traffic.
- *Temporal Behavior.* Night-time activity was significantly higher on the vulnerable system (Cramér's $V = 0.33$, $p < 0.001$), suggesting that attackers deliberately exploit reduced operational oversight during off-peak hours.
- *IP Address Overlap.* Only 5% of source IPs were common across the two systems, indicating selective targeting rather than indiscriminate scanning.

Healthcare-Specific vs. Generic Honeypots. To contrast domain-specific deception with generic deployments, we compared our healthcare-targeted web apps with a co-located Cowrie honeypot. Statistical analysis of destination port distribution yielded $\chi^2(14) = 3{,}897{,}878$, $p < 10^{-4}$, and Cramér's $V = 0.60$, indicating a large effect. Healthcare traffic targeted endpoints such as `/openmrs/ws/rest/v1/patient` and `/openemr/interface/billing/`, with Suricata flagging healthcare CVEs like CVE-2023-43208 and CVE-2022-31496. These threats were absent from the Cowrie instance, highlighting the necessity of healthcare-specific deception to surface relevant attack vectors.

Summary. Our ablation analysis shows the value of deploying deception networks that emulate healthcare-specific infrastructure. By capturing real-world attacker interactions, such setups reveal targeted exploitation patterns across protocols and configurations. These insights inform patch prioritization, support proactive defenses, and strengthen monitoring during high-risk periods, enhancing resilience in critical environments like healthcare.

IX. Recommendations

To enhance the practical utility of insights from our healthcare deception network, we offer the following recommendations aligned with current capabilities and future development.

Deploying Workflow-Specific Decoys and Prioritizing Patching. We recommend deploying decoy HL7 interfaces to attract adversaries targeting EHR workflows without exposing real patient data. Given HL7's plaintext transmission and frequent targeting, it presents an ideal deception surface. These decoys produced actionable findings that can guide patching priorities for production systems in the same application class. Integrate telemetry from attacker interactions into vulnerability management systems. Persistent targeting of specific endpoints e.g., billing portals or known CVEs (e.g., SQLi, XSS) should trigger higher remediation priority. Our ablation study (§VIII-B) confirms deliberate, targeted behavior, underscoring the value of data-driven patch management.

Enhancing Real-time Threat Analysis and Intelligence Integration. We recommend ensuring that alerts and events generated by the deception network are formatted using standard schemas (e.g., JSON, CEF, Syslog) for smooth integration with SIEM platforms such as Splunk and Wazuh. This compatibility allows for real-time correlation with broader threat telemetry, improving situational awareness. Our system maintains sub-5-second latency for log processing even under sustained attack volumes, supporting timely triage and threat response.

Additionally, we suggest incorporating nuanced patterns uncovered during the study such as protocol-specific, regional, and temporal attack variations into threat detection strategies. For example, recognizing peak scanning activity during specific hours (e.g., early mornings or afternoon lulls) can help optimize resource allocation for monitoring (see § VII). Observing consistent malware signatures (e.g., LockBit 3.0) across cloud environments further enables targeted defensive postures.

Strengthening Deception Resilience and Stealth. To preserve deception effectiveness, we recommend employing dynamic data rotation and regularly validating honeypots against reconnaissance tools beyond Shodan's Honeyscore. This helps maintain stealth and prevents early detection by sophisticated adversaries. Future enhancements should include live traffic emulation and broader stealth evaluations to simulate realistic service behavior and evade advanced scanning techniques. We also emphasize the importance of outbound traffic control from deception environments. Implementing WAFs and restricting high-risk protocols such as UDP can prevent abuse–such as DNS amplification or DDoS attacks. Continuous integrity monitoring via tools like Wazuh ensures honeypots remain uncompromised and operationally safe.

Operationalization and Broader Application. Future work will focus on analyzing real-world deployments to assess detection failures, false positives, and response effectiveness, enabling iterative improvements. We also propose extending this approach to other critical sectors like finance and energy, which present distinct threat models. To support attribution, incorporating a control group of generic honeypots can help isolate healthcare-specific targeting patterns.

Ensuing compliance and proactive risk management. By logging and simulating attempted breaches on healthcare services, our system supports continuous monitoring mandates under frameworks such as HIPAA and ISO 27001. These logs provide tangible evidence of proactive risk management and can streamline audit preparation.

X. Conclusion and Future Work

We introduced a healthcare-focused deception network that reveals multi-dimensional attack patterns regional, protocol-specific, and temporal through co-regularized spectral clustering. The results confirmed that attackers deliberately targeted healthcare systems using protocols like HTTP, DNS, and TLS. Our findings provided actionable insights for deploying deception networks and prioritizing patching efforts based on real-world attacker behavior. The work established a foundation for integrating deception into healthcare security frameworks.

Future research may focus on adaptive deception networks that respond to real-time threat intelligence, as well as privacy-preserving analysis of encrypted traffic (e.g., TLS). Expanding to other critical sectors, incorporating protocols like FHIR, and analyzing real-world deployment incidents will be key to refining strategies and improving resilience.

References

[1] McKinsey, "Healthcare players moving towards the cloud," 2023. [Online]. Available: https://www.mckinsey.com/industries/healthcare/our-insights/healthcare-players-moving-towards-the-cloud

[2] DuploCloud. (2023) 70% of healthcare businesses have adopted cloud computing. Press release. [Online]. Available: https://www.globenewswire.com/news-release/2023/02/22/2613339/0/en/70-of-Healthcare-Businesses-Have-Adopted-Cloud-Computing-DuploCloud-Report.html

[3] C. Van Alstin, "Healthcare data breaches have doubled this year, impacting a quarter of americans," *Health Exec*, 2023. [Online]. Available: https://healthexec.com/topics/health-it/cybersecurity/healthcare-data-breaches-2023-report

[4] World Health Organization, "International statistical classification of diseases and related health problems (icd-10)," 2016, 10th Revision. [Online]. Available: https://www.who.int/classifications/icd/en/

[5] U.S. Food and Drug Administration, "National drug code (ndc) directory," 2023. [Online]. Available: https://www.fda.gov/drugs/drug-approvals-and-databases/national-drug-code-directory

[6] Health Level Seven International, "Health level seven (hl7) standards," 2023. [Online]. Available: https://www.hl7.org/

[7] "Standards adoption among health information exchange organizations," Office of the National Coordinator for Health IT, Tech. Rep. Data Brief No. 65, 2023, accessed 14 Jul 2025. [Online]. Available: https://www.healthit.gov/data/data-briefs/standards-adoption-among-health-information-exchange-organizations

[8] HL7 International and Firely, "2024 state of fhir survey results," 2024, accessed 14 Jul 2025. [Online]. Available: https://interop.esante.gouv.fr/ig/doctrine/0.1.0/2024%20StateofFHIRSurveyResults_final.pdf

[9] T. Guest, "Hl7 data interfaces in medical environments," accessed: 2021-01-20. [Online]. Available: https://www.tripwire.com/state-of-security/hl7-data-interfaces-in-medical-environments

[10] M. Christian Dameff, M. Bland, and J. Tully, "Pestilential protocol: How unsecure hl7 messages threaten patient lives."

[11] Auriga Press Center, "HL7 Security Issues – How to Protect Patient Data," https://auriga.com/blog/2016/hl7-security-issues-how-to-protect-patient-data/, accessed: 2024-04-15.

[12] T. Bari and M. Abualkibash, "The impact of intrusion detection systems upon healthcare environments: A research review," in *Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management (IEOM)*. Singapore: IEOM Society International, Mar. 2021, pp. 2363–2365. [Online]. Available: https://www.ieomsociety.org/singapore2021/papers/453.pdf

[13] G. Dupont, D. dos Santos, S. Dashevskyi, S. Vijayakumar, S. P. Murali, E. Costante, J. den Hartog, and S. Etalle, "Demonstration of new attacks on three healthcare network protocols in a lab environment," *Journal of Computer Virology and Hacking Techniques*, vol. 20, pp. 301–314, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s11416-023-00479-w

[14] R. W. Anwar, T. Abdullah, and F. Pastore, "Firewall best practices for securing smart healthcare environment: A review," *Applied Sciences*, vol. 11, no. 19, p. 9183, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/19/9183

[15] M. Zhang, L. Wang, S. Jajodia, and A. Singhal, "Network attack surface: Lifting the attack surface concept to network level for evaluating the resilience against zero-day attacks," in *Proceedings of the International Conference on Network and System Security (NSS)*. New York, NY, USA: Springer, 2015, pp. 10–23.

[16] A. Singhal, X. Ou, and J. Erickson, "A network-based framework for modeling and measuring the cyber attack surface," *Journal of Information Security and Applications*, vol. 19, no. 2, pp. 132–146, 2016.

[17] J. Everson and J. Cheng, "A survey on network attack surface mapping," *Computers & Security*, vol. 122, pp. 103–123, 2024.

[18] O. of the Australian Information Commissioner (OAIC), "Notifiable data breaches report: January to june 2024," September 2024, oAIC Report. [Online]. Available: https://www.oaic.gov.au

[19] H. Magazine, "The future of healthcare in the public cloud," 2023. [Online]. Available: https://healthtechmagazine.net/article/2023/10/future-healthcare-public-cloud

[20] D. T. S. GmbH, "T-pot community edition," 2023. [Online]. Available: github.com/telekom-security/tpotce

[21] B. Al-Nazer, "Att&ck bert: a cybersecurity language model," https://huggingface.co/basel/ATTACK-BERT, 2023.

[22] K. Ma, "Secroberta: A pretrained language model for cyber security text," https://huggingface.co/jackaduma/SecRoBERTa, 2020, accessed: 2025-01-16.

[23] U. Kumarasinghe, A. Lekssays, H. T. Sencar, S. Boughorbel, C. Elvitigala, and P. Nakov, "Semantic ranking for automated adversarial technique annotation in security text," *arXiv preprint arXiv:2403.17068*, 2024, https://huggingface.co/qcri-cs/SentSecBert_10k.

[24] A. Author1, A. Author2, A. Author3, and A. Author4, "Healthcare deception: Real-time threat detection in healthcare web systems," 2025, accessed 31 Jul 2025. [Online]. Available: https://github.com/Cyb3rD0c/healthcare-deception

[25] P. B. López, M. Gil, and P. Nespoli, "Cyber deception: State of the art, trends, and open challenges," *arXiv preprint arXiv:2409.07194*, 2023.

[26] J. Barron and N. Nikiforakis, "Picky attackers: Quantifying the role of system properties on intruder behavior," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1681–1692.

[27] T. Dang, H. Nguyen, and J. H. Kim, "Understanding fileless attacks on linux-based iot devices with honeypots," in *Proceedings of the 2019 ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '19)*. ACM, 2019, pp. 113–118.

[28] F. Favale, A. Fontana, and F. Maggi, "What scanners do at l7: Exploring horizontal honeypots at the application layer," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 1267–1283.

[29] Z. Cao, T. Xu, X. Feng, X. Luo, and J. Wang, "Cloud-watching: Monitoring and analyzing public clouds through passive observation and probing," in *Proceedings of the 2019 USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2019, pp. 125–141.

[30] Á. Balogh, M. Érsok, L. Erdődi, A. Szarvák, E. Kail, and A. Bánáti, "Honeypot optimization based on ctf game," in *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2022, pp. 000 153–000 158.

[31] J. S. López-Yépez and A. Fagette, "Increasing attacker engagement on ssh honeypots using semantic embeddings of cyber-attack patterns and deep reinforcement learning," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2022, pp. 389–396.

[32] D. Barradas, C. Novo, B. Portela, S. Romeiro, and N. Santos, "Extending c2 traffic detection methodologies: From tls 1.2 to tls 1.3-enabled malware," in *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. ACM, 2024, pp. 1–16.

[33] B. Irwin, "A source analysis of the conficker outbreak from a network telescope," *South African Institute of Electrical Engineers*, vol. 104, pp. 38–49, 2013.

[34] ——, "A baseline study of potentially malicious activity across five network telescopes," in *2013 5th International Conference on Cyber Conflict*. IEEE, 2013, pp. 1–15.

[35] P. Richter and A. Berger, "Scanning the scanners: Sensing the internet from a massively distributed network telescope," in *Proceedings of the 2019 ACM Internet Measurement Conference (IMC)*, 2019, pp. 144–160.

[36] H. Heo and S. Shin, "Who is knocking on the telnet port: A large-scale empirical study of network scanning," in *Proceedings of the 2018 ACM Asia Conference on Computer and Communications Security*. ACM, 2018, pp. 636–648.

[37] T. Yu, Y. Xin, and C. Zhang, "Honeyfactory: Container-based comprehensive cyber deception honeynet architecture," *Electronics*, vol. 13, no. 2, p. 361, 2024.

[38] T. Holz and F. Raynal, "Detecting honeypots and other suspicious environments," in *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security*. IEEE, 2005, pp. 29–36.

[39] A. Vetterl and R. Clayton, "Bitter harvest: Systematically fingerprinting low-and medium-interaction honeypots at internet scale," in *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, 2018.

[40] S. Morishita, T. Hoizumi, W. Ueno, R. Tanabe, C. Ganaán, M. J. van Eeten, K. Yoshioka, and T. Matsumoto, "Detect me if you... oh wait. an internet-wide view of self-revealing honeypots," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 134–142.

[41] Censys, "Censys search," https://search.censys.io, accessed on: July 25, 2024.

[42] Shodan, "Shodan.io: Honeypot or not?" https://honeyscore.shodan.io/, accessed on: March 25, 2024.

[43] CISA, "ICSMA-21-007-01: Innokas Yhtymä Vital Signs Monitor — HL7 v2.x Segment Injection," 2021, accessed 14 Jul 2025. [Online]. Available: https://www.cisa.gov/news-events/ics-medical-advisories/icsma-21-007-01

[44] HIPAA Journal. (2023) Critical nextgen healthcare mirth connect vulnerability under active exploitation. Describes CVE-2023-43208; accessed 14 Jul 2025. [Online]. Available: https://www.hipaajournal.com/critical-nextgen-healthcare-mirth-connect-under-active-exploitation/

[45] S. Beale, "Epic expands ehr market share as rivals lose customers," *Dark Daily*, 2025. [Online]. Available: https://darkdaily.com/2025/06/04/epic-expands-ehr-market-share-as-rivals-lose-customers/

[46] N. Diaz, "How much 13 health systems are paying for ehrs," *Becker's Hospital Review*, 2024. [Online]. Available: https://www.beckershospitalreview.com/ehrs/how-much-13-health-systems-are-paying-for-ehrs

[47] OpenMRS Community, "Our impact," 2025. [Online]. Available: https://openmrs.org/what-we-do/our-impact/

[48] OpenEMR Foundation, "Openemr achieves onc certification with groundbreaking release 7.0," 2022. [Online]. Available: https://www.open-emr.org/blog/openemr-achieves-onc-certification-with-groundbreaking-release-70/

[49] Quality Systems Inc., "Quality systems releases mirth connect 3.0," 2013. [Online]. Available: https://www.openhealthnews.com/content/quality-systems-inc-releases-mirth-connect-30-open-source-healthcare-integration-engine

[50] N. I. of Standards and T. (NIST), "National vulnerability database (nvd) - search vulnerabilities," 2024, accessed on: January 8, 2024. [Online]. Available: https://nvd.nist.gov/vuln/search

[51] M. Corporation, "Common vulnerabilities and exposures (cve) - cve list search," 2024, accessed on: January 8, 2024. [Online]. Available: https://cve.mitre.org/cve/search_cve_list.html

[52] F. Community, "Faker documentation," 2024, accessed on: June 18, 2024. [Online]. Available: https://faker.readthedocs.io/en/master/

[53] T. Canary, "Canarytokens - free, quick, disposable intrusion detection tokens," 2024, accessed on: June 22, 2024. [Online]. Available: https://www.canarytokens.org/nest/

[54] I. Wazuh, *Wazuh Documentation: Release Notes for 4.7.5*, 2023, accessed on: June 8, 2024. [Online]. Available: https://documentation.wazuh.com/current/release-notes/release-4-7-5.html

[55] MITRE Corporation, "MITRE ATT&CK Framework," https://attack.mitre.org/, accessed: 2024-04-15.

[56] OISF, *Suricata 6.0.5 Documentation*, 2021, accessed on: May 21, 2024. [Online]. Available: https://docs.suricata.io/en/suricata-6.0.5/

[57] Emerging Threats, "Emerging threats ruleset," https://community.emergingthreats.net/, accessed: 2024-06-22.

[58] Anonymous, "Pastebin content," 2023, accessed on: May 22, 2024. [Online]. Available: https://pastebin.com/EWSQQkBf

[59] ——, "Pastebin content," 2023, accessed on: May 22, 2024. [Online]. Available: https://pastebin.com/eqGtVvdX

[60] Elastic, *Logstash: Collect, Parse, Transform Logs*, 2025, accessed: 2024-03-10. [Online]. Available: https://www.elastic.co/logstash

[61] ——, *Elasticsearch Reference Documentation*, 2023, accessed on: May 23, 2024. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html

[62] GreyNoise, "Greynoise: Noise reduction for threat intelligence," https://www.greynoise.io, accessed on: March 1, 2024.

[63] Elastic, *Kibana: Explore, Visualize, Discover Data*, 2025, accessed: 2024-03-10. [Online]. Available: https://www.elastic.co/kibana

[64] I. Merit Network, "Orion network telescope," 2023, accessed on: July 21, 2024. [Online]. Available: https://www.merit.edu/initiatives/orion-network-telescope/

[65] Palo Alto Networks, "Built-in external dynamic lists (edls) – known malicious ip addresses," 2025. [Online]. Available: https://docs.paloaltonetworks.com/pan-os/10-1/pan-os-admin/policy/use-an-external-dynamic-list-in-policy/built-in-edls

[66] Shodan, "Shodan: The search engine for the internet of everything," https://www.shodan.io, accessed on: July 23, 2024.

[67] Offensive Security, "Exploit Database (ExploitDB)," 2025, accessed: 2025-07-31. [Online]. Available: https://www.exploit-db.com

[68] GitHub, Inc., "Proof-of-concept exploit repositories on github," 2025. [Online]. Available: https://github.com/search?q=proof+of+concept+exploit

[69] L. Miranda, C. Figueiredo, D. S. Menasché, and A. Kocheturov, "Patch or exploit? nvd assisted classification of vulnerability-related github pages," in *Cyber Security, Cryptology, and Machine Learning (CSCML 2023)*, ser. Lecture Notes in Computer Science. Springer, Cham, 2023, vol. 13467, pp. 511–522. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-34671-2_36

[70] E. Iannone, G. Sellitto, E. Iaccarino, F. Ferrucci, A. De Lucia, and F. Palomba, "Early and realistic exploitability prediction of just-disclosed software vulnerabilities: How reliable can it be?" *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 33, no. 2, pp. 1–41, 2024. [Online]. Available: https://giuliasellitto7.github.io/pdf/Iannone-TOSEM2024-Early-and-Realistic-Exploitability-Prediction-of-Just-Disclosed-Sotware-Vulnerabilities.pdf

[71] J. Díaz-Verdejo, J. Muñoz-Calle, A. E. Alonso, R. E. Alonso, and G. Madinabeitia, "On the detection capabilities of signature-based intrusion detection systems in the context of web attacks," *Applied Sciences*, vol. 12, no. 2, p. 852, 2022.

[72] T. T. Pham, T. M. Loo, A. Malhotra, C. A. Longhurst, D. Hylton, C. Dameff, J. Tully, G. Wardi, R. E. Sell, and A. K. Pearce, "Ransomware cyberattack associated with cardiac arrest incidence and outcomes at untargeted, adjacent hospitals," *Critical Care Explorations*, vol. 6, no. 4, p. e1079, 2024.

[73] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[74] S. Trivedi, "Mass campaign of murdoc botnet mirai: A new variant of corona mirai," 2025, qualys Threat Research Unit blog post, accessed 15 Jul 2025. [Online]. Available: https://blog.qualys.com/vulnerabilities-threat-research/2025/01/21/mass-campaign-of-murdoc-botnet-mirai-a-new-variant-of-corona-mirai

[75] Red Piranha TIR Team, "Threat intelligence report – 31 dec 2024 to 6 jan 2025," 2025, weekly report describing a new Mirai variant (CatDDoS); accessed 15 Jul 2025. [Online]. Available: https://redpiranha.net/news/threat-intelligence-report-december-31-2024-january-6-2025

[76] Akamai SIRT, "Credential stuffing in the era of serverless bots," https://www.akamai.com/blog/security/2024-credential-stuffing-report, 2024, accessed 15 Jul 2025.

[77] AlienVault, "Otx: Open threat exchange," https://otx.alienvault.com/, accessed on: September 1, 2024.

[78] VirusTotal, "Virustotal: Free online virus, malware, and url scanner," https://www.virustotal.com/gui/home/upload, accessed on: September 1, 2024.

[79] Hybrid Analysis, "Hybrid analysis: Free automated malware analysis service," https://www.hybrid-analysis.com, accessed on: September 1, 2024.

[80] M. Oosterhof, *Cowrie Documentation*, 2023, accessed on: June 10, 2024. [Online]. Available: https://cowrie.readthedocs.io/en/latest/index.html

[81] M. Koetter and the Dionaea Development Team, "Dionaea honeypot," https://github.com/DinoTools/dionaea, accessed: 2024-06-20.

[82] I. Docker, "Docker documentation," 2024, accessed on: February 3, 2024. [Online]. Available: https://docs.docker.com

[83] C. Ltd., "Ubuntu: The leading operating system for cloud and iot," 2024, accessed on: March 25, 2024. [Online]. Available: https://ubuntu.com/#get-ubuntu

[84] G. Community, "Gunicorn: Python wsgi http server for unix," 2024, accessed on: August 5, 2024. [Online]. Available: https://gunicorn.org

[85] OWASP, "Content security policy - owasp," 2024. [Online]. Available: https://owasp.org/www-project-secure-headers/#content-security-policy

[86] Mozilla, "Security headers - mozilla developer network (mdn)," https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers, 2024.

[87] OWASP, "Owasp zap - zed attack proxy," 2024. [Online]. Available: zaproxy.org/

[88] D. L. Chris Sullo, "Nikto web server scanner," https://cirt.net/Nikto2, 2024.

[89] SourceCodester, "Clinic's patient management system in php/pdo free source code," 2022, accessed on: February 10, 2024. [Online]. Available: https://www.sourcecodester.com/tags/clinics-patient-management-system-source-code-php

[90] ——, "Covid 19 testing management system (ctms) in php free source code," 2021, accessed on: February 9, 2024. [Online]. Available: https://www.sourcecodester.com/php/14960/covid-19-testing-management-system-php-free-source-code.html
[91] ——, "Doctor's appointment system using php free source code," 2022, accessed on: February 9, 2024. [Online]. Available: https://www.sourcecodester.com/php/14182/doctor-appointment-system.html
[92] HashenUdara, "Edoc doctor appointment system using php," 2023, accessed on: February 9, 2024. [Online]. Available: https://github.com/HashenUdara/edoc-doctor-appointment-system
[93] P. Mvuma, "Electronic medical records system (emr) using php with source code," 2021, accessed on: February 9, 2024. [Online]. Available: https://www.sourcecodester.com/php/13475/electronic-medical-records-system-emr.html
[94] M. K., "Free hospital management system for small practices," 2023, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/16720/free-hospital-management-system-small-practices.html
[95] Razormist, "Health center patient record management system using php with source code," 2021, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/11058/health-center-patient-record-management-system.html
[96] L. Technologies, "Hhims: Hospital health information management system," 2024, accessed on: February 7, 2024. [Online]. Available: https://github.com/tsruban/HHIMS
[97] M. K., "Upturn hospital management system using php with free source code," 2018, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/tags/hms
[98] Code-Projects, "Hospital information system in php with source code," 2024, accessed on: February 7, 2024. [Online]. Available: https://code-projects.org/hospital-information-system-in-php-with-source-code/
[99] SourceCodester, "Hospital management system using php free source code," 2022, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/14795/hospital-management-system-php-free-source-code.html
[100] M. K., "Free hospital management system for small practices," 2023, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/16720/free-hospital-management-system-small-practices.html
[101] Oretnom23, "Medical certificate generator app using php and mysql free source code," 2023, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/16105/medical-certificate-generator-app-using-php-and-mysql-free-download.html
[102] M. S. Chen, "Nosh chartingsystem - free open source health charting system," 2024, accessed on: February 7, 2024. [Online]. Available: https://github.com/shihjay2/nosh2
[103] SourceCodester, "Hospital management system in php free source code," 2021, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/15112/hospital-management-system-php-free-source-code.html
[104] ——, "Hospital's patient records management system in php free source code," 2021, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/15116/hospitals-patient-records-management-system-php-free-source-code.html
[105] LibreHealth, "Librehealth ehr - free open source electronic health records," 2023, accessed on: February 7, 2024. [Online]. Available: https://gitlab.com/librehealth/ehr
[106] janobe, "Online doctor appointment system in php with full source code," 2021, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/14663/online-doctor-appointment-system-php-full-source-code.html
[107] ——, "Online health care system in php with full source code," 2020, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/14526/online-health-care-system-php-full-source-code-2020.html
[108] W. Daloyan, "Online hospital management system using php/mysql," 2020, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/14386/online-hospital-management-system-using-phpmysql.html
[109] Oretnom23, "Patient appointment scheduler system using php free source code," 2021, accessed on: February 10, 2024. [Online]. Available: https://www.sourcecodester.com/php/14928/patient-appointment-scheduler-system-using-php-free-source-code.html
[110] O. Project, "Openemr - the most popular open source electronic health records and medical practice management solution," 2024, accessed on: February 10, 2024. [Online]. Available: https://github.com/openemr/openemr
[111] O.-C. Community, "Open-clinic: Open source hospital management system," 2024, accessed on: February 10, 2024. [Online]. Available: https://sourceforge.net/projects/open-clinic/
[112] J. A. Chavarría, "Openclinic: Open source medical records system," 2024, accessed on: February 10, 2024. [Online]. Available: https://openclinic.sourceforge.net/
[113] O. Community, "Openmrs - open source medical record system," 2024, accessed on: February 10, 2024. [Online]. Available: https://github.com/openmrs/openmrs-core
[114] fkgeo, "Pharmacy/medical store point of sale system using php/mysql and bootstrap framework with source code," 2020, accessed on: February 10, 2024. [Online]. Available: https://www.sourcecodester.com/php/14957/pharmacy-medical-store-point-sale-system-php-free-source-code.html
[115] ——, "Pharmacy/medical store point of sale system using php/mysql and bootstrap framework with source code," 2020, accessed on: February 10, 2024. [Online]. Available: https://www.sourcecodester.com/php/14398/pharmacymedical-store-sale-point-using-phpmysql-bootstrap-framework.html
[116] Projectworlds, "Hospital management system in php with source code," 2024, accessed on: February 11, 2024. [Online]. Available: https://projectworlds.in/free-projects/php-projects/hospital-management-system-in-php/
[117] R. Community, "Remoteclinic - open source clinic management system," 2024, accessed on: February 11, 2024. [Online]. Available: https://github.com/remoteclinic/RemoteClinic
[118] Oretnom23, "Simple doctor's appointment system using php/mysql with source code," 2020, accessed on: February 7, 2024. [Online]. Available: https://www.sourcecodester.com/php/14467/simple-doctors-appointment-system-using-phpmysql-source-code.html
[119] A. Kumar, "Hospital management system using php with source code," 2024, accessed on: February 11, 2024. [Online]. Available: https://phpgurukul.com/sdm_downloads/download-source-codehospital-management-system/

ACKNOWLEDGMENTS

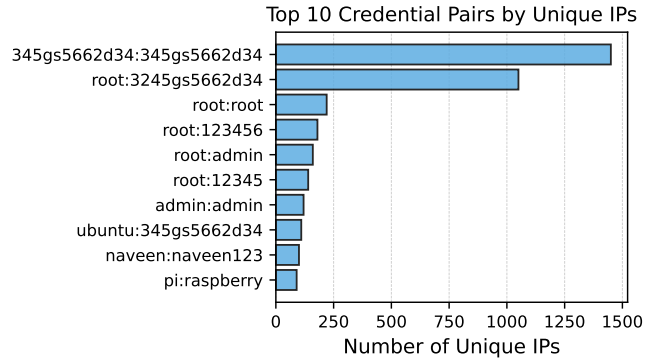
This work was partially supported by the Macquarie University Cybersecurity Hub (MQCHUB). The Macquarie University International High Degree Research Scholarship Program supported Zeeshan Zulkifl Shah. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors or originators and do not necessarily reflect the views of the MQCHUB or Macquarie University.

APPENDIX

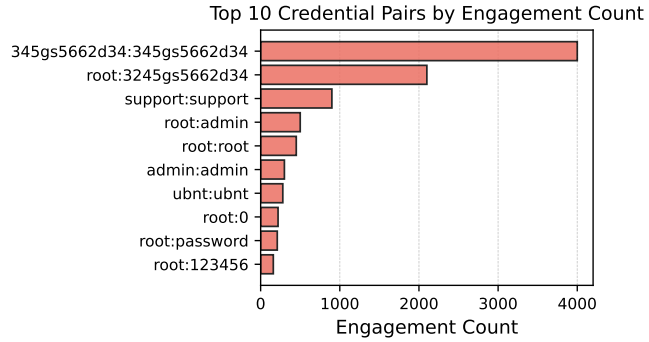
Figure 1 shows the architecture of our deception network. In the following, we provide a detailed summary of the vulnerable healthcare web applications used in this study, which were specifically selected based on their documented security weaknesses, such as SQL injection, XSS, and Path Traversal vulnerabilities. These applications, including Electronic Medical Records (EMR) systems and Hospital Management Platforms (HMS), were sourced from publicly available repositories and enriched with synthetic patient data to emulate real-world healthcare environments.

Table V summarizes the key attributes of the selected applications, including their category, versions, associated CVEs, CWE references, and information on whether the vulnerabilities have been exploited or patched. The data gathered from these applications informed the analysis of attacker behavior in healthcare-specific cloud environments, as discussed in the main body of this paper.

Additionally, Figures 7a, and 7b visualize findings from our deception-based threat detection network showing the most frequently targeted credential pairs by unique IPs and engagement counts.



(a) Top 10 Credential Pairs by Unique IPs.



(b) Top 10 Credential Pairs by Engagement Count.

Fig. 7: Analysis of top credential pairs: (a) by Unique IPs and (b) by engagement count.

This subsection provides explanations of the key technical tools and standards referenced in this study. These definitions and examples are included to assist readers unfamiliar with domain-specific terminology.

HL7 messaging protocol [6]. It is an internationally recognized set of standards for the exchange of clinical and administrative data in healthcare systems. HL7 defines structured messages like Admission-Discharge-Transfer (ADT) and Order Entry Message (ORM) to facilitate interoperability between healthcare applications. For instance, an ADT message is shown in Listing 1. This message communicates that a patient named John Doe was admitted to Hospital A on January 20, 2025.

International Classification of Diseases (ICD-10). The International Classification of Diseases (ICD-10) is a globally accepted system for coding diagnoses and health conditions. For example, ICD-10 code `E11.9` represents "Type 2 diabetes mellitus without complications". These codes streamline documentation and facilitate healthcare analytics.

National Drug Code (NDC). The National Drug Code (NDC) is a unique identifier for medications in the United States, maintained by the Food and Drug Administration (FDA). An example is `0781-1842-10`, which represents a bottle of 40 mg Furosemide tablets (100 count). NDC codes ensure precise identification of pharmaceuticals.

Wazuh. It is an open-source security platform that integrates threat detection, compliance management, and integrity monitoring. It collects logs from distributed sources and maps them to the MITRE ATT&CK framework, providing insights into tactics, techniques, and procedures (TTPs). In this study, Wazuh identified activities like brute-force login attempts (T1110) and data exfiltration attempts.

Suricata. It is a high-performance, open-source network intrusion detection and prevention system (NIDS/IPS). It operates at multiple layers of the network stack and can process custom rule sets. For example, a rule targeting DNS tunneling might detect abnormal packet flows indicative of data exfiltration.

Logstash. As a part of the ELK stack, it is a powerful tool for collecting, parsing, and enriching log data. It supports input from various sources and can transform data into a standardized format for downstream analysis. For instance, it was used in this study to aggregate logs from Wazuh and Suricata while adding GeoIP and CVE enrichment.

Kibana. It is an open-source visualization tool in the ELK stack that allows users to explore, visualize, and interact with data stored in Elasticsearch. Dashboards were created to visualize attack trends, such as SQL injection attempts or geographic distributions of malicious activity.

T-Pot Honeypot Framework. T-Pot is a multi-honeypot framework developed by Deutsche Telekom. It combines several open-source honeypot technologies (e.g., Cowrie, Dionaea) into a containerized environment. It includes a native ELK stack for log management and is tailored for high-interaction deployments.

Canary Tokens. These are lightweight intrusion detection mechanisms embedded in sensitive resources (e.g., URLs, files). When accessed, they generate alerts, providing insights into unauthorized reconnaissance activities. For example, embedding a token in a fake patient record URL helped detect malicious scanning.

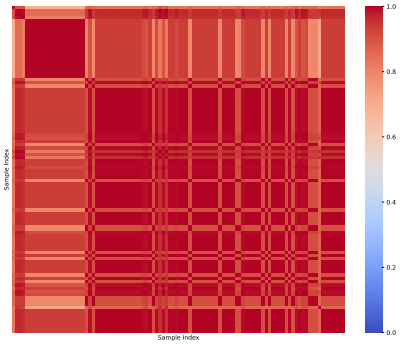
MITRE ATT&CK Framework. It is a curated knowledge base of adversary tactics, techniques, and procedures (TTPs). For example, T1110 refers to brute force attacks. This framework enables standardized threat classification and supports structured defense strategies.

To validate the similarity matrices generated during the preprocessing step of our analysis, we provide the following heatmaps for four key feature groups: semantic, categorical, numerical, and temporal. Each heatmap presents a 100×100 subset of the corresponding similarity matrix, offering a clear representation of the relationship between data points across different feature dimensions. These visualizations help assess the consistency and alignment of the similarity computations with the intended feature attributes.

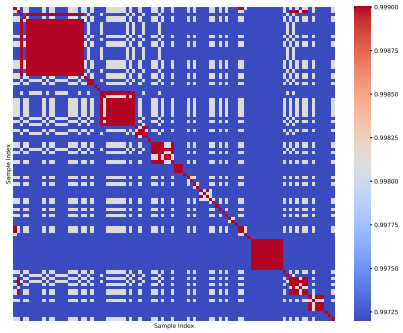
The purpose and observations of the heatmaps are detailed below:

TABLE V: Summary of selected vulnerable healthcare web applications (cf. §IV).

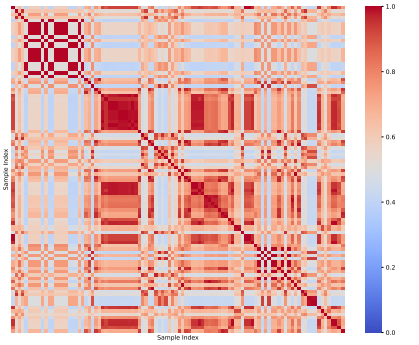
Software	Category	Version	CVE(s)	CWE(s)	Exploit	Patch
Clinic's Patient Management System Project [89]	EMR	1 & 2	CVE-2022-3120, CVE-2022-3122, CVE-2022-36242	CWE-89 (SQL Injection), CWE-79 (XSS)	Yes	No
Covid 19 Testing Management System Project [90]	Laboratory System	1	CVE-2023-1300	CWE-89 (SQL Injection)	Yes	No
Doctor Appointment System Project [91]	Patient Appointment System	1	CVE-2021-27314, CVE-2021-27315, CVE-2021-27316	CWE-22 (Path Traversal)	Yes	No
Edoc-Doctor-Appointment-System Project [92]	Patient Appointment System	1.0.1	CVE-2022-36543, CVE-2022-36544, CVE-2022-36545, CVE-2022-36546	CWE-79 (XSS)	Yes	No
Electronic Medical Records System Project [93]	EMR	1	CVE-2022-2676, CVE-2022-2693, CVE-2023-1151	CWE-89 (SQL Injection)	Yes	No
Free Hospital Management System For Small Practices Project [94]	HMS	1	CVE-2023-4440, CVE-2023-4441, CVE-2023-4442, CVE-2023-4443, CVE-2023-4444, CVE-2023-5587	CWE-89 (SQL Injection), CWE-79 (XSS)	Yes	No
Health Center Patient Record Management System [95]	HMS	1	CVE-2023-1253	CWE-89 (SQL Injection)	Yes	No
HHIMS Project [96]	HMS	2.1	CVE-2022-3956	CWE-89 (SQL Injection)	Yes	No
HMS Project [97]	HMS	1	CVE-2022-23365, CVE-2022-23366	CWE-89 (SQL Injection)	Yes	No
Hospital Information System Project [98]	HMS	N/A	CVE-2022-36669	CWE-89 (SQL Injection)	Yes	No
Hospital Management Center Project [99]	HMS	N/A	CVE-2022-4012, CVE-2022-4013	CWE-89 (SQL Injection), CWE-79 (XSS)	Yes	No
Mayurik Free Hospital Management System For Small Practices [100]	HMS	1	CVE-2023-4179, CVE-2023-4180, CVE-2023-4181, CVE-2023-4185	CWE-89 (SQL Injection), CWE-79 (XSS)	Yes	No
Medical Certificate Generator App Project [101]	Other	1	CVE-2023-0706, CVE-2023-0707, CVE-2023-0774, CVE-2023-1566	CWE-89 (SQL Injection), CWE-79 (XSS)	Yes	No
Nosh Chartingsystem Project [102]	Patient Appointment System	1	CVE-2023-24610	CWE-434 (Unrestricted file upload)	Yes	No
Hospital Management System Project [103]	HMS	1	CVE-2021-38754, CVE-2023-43909, CVE-2021-44095	CWE-89 (SQL Injection), CWE-79 (XSS)	Yes	No
Hospital's Patient Records Management System Project [104]	EMR	1	CVE-2022-22854, CVE-2022-24232, CVE-2022-25003	CWE-89 (SQL Injection)	Yes	No
Librehealth EHR [105]	EMR	2.0.0	CVE-2020-11436, CVE-2020-11438, CVE-2020-11439	CWE-89 (SQL Injection), CWE-79 (XSS), CWE-22 (Path Traversal)	Yes	No
Online Doctor Appointment Booking System Php And Mysql	Patient Appointment System [106]	1	CVE-2020-29168, CVE-2020-29283	CWE-89 (SQL Injection), CWE-79 (XSS)	Yes	No
Online Health Care System Project [107]	HMS	1	CVE-2022-46471	CWE-89 (SQL Injection)	Yes	No
Online Hospital Management System Project [108]	HMS	N/A	CVE-2023-37069	CWE-89 (SQL Injection)	Yes	No
Patient Appointment Scheduler System Project [109]	Patient Appointment System	N/A	CVE-2021-41660	CWE-89 (SQL Injection)	Yes	No
Open-Emr [110]	EMR	N/A	CVE-2020-19364, CVE-2019-8371	CWE-434 (Unrestricted file upload), CWE-89 (SQL Injection), CWE-79 (XSS)	Yes	No
Openclinic Ga Project [111]	HMS	5.173.3	CVE-2020-27229, CVE-2020-27230, CVE-2020-27231	CWE-89 (SQL Injection), CWE-79 (XSS)	Yes	No
Openclinic Project [112]	HMS	0.8.2	CVE-2020-28937, CVE-2020-28939	CWE-89 (SQL Injection)	Yes	No
Openmrs [113]	EMR	1.6, 2.2.0, 2.3.0	CVE-2021-43094, CVE-2022-23612	CWE-79 (XSS)	Yes	No
Pharmacy Medical Store And Sale Point Project [114]	Telemedicine	N/A	CVE-2020-24862	CWE-89 (SQL Injection)	Yes	No
Pharmacy Point Of Sale System Project [115]	Telemedicine	N/A	CVE-2021-41676	CWE-89 (SQL Injection)	Yes	No
Projectworlds Hospital Management System In Php [116]	HMS	1	CVE-2021-43628, CVE-2021-43629	CWE-89 (SQL Injection), CWE-79 (XSS)	Yes	No
Remote Clinic [117]	Telemedicine	N/A	CVE-2023-33481	CWE-89 (SQL Injection)	Yes	Yes
Simple Doctor's Appointment System Project [118]	Patient Appointment System	1	CVE-2022-28568	CWE-89 (SQL Injection)	Yes	No
Phpgurukul Hospital Management System [119]	HMS	1, 4	CVE-2023-7172, CVE-2020-22164, CVE-2020-22165	CWE-79 (XSS)	Yes	No



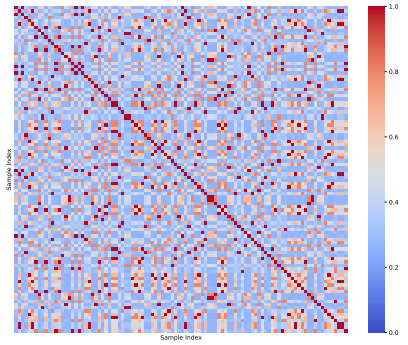
(a) Semantic Similarity Matrix



(b) Categorical Similarity Matrix



(c) Numerical Similarity Matrix



(d) Temporal Similarity Matrix

Fig. 8: Heatmaps representing similarity matrices for different feature groups: semantic, categorical, numerical, and temporal. Each heatmap uses a 100×100 subset of the corresponding similarity matrix (see §VI-B for details).

A. *Purpose of the Heatmaps*

These heatmaps serve a dual purpose:

- **Validation.** They confirm that the similarity computations align with the expected feature characteristics, ensuring the preprocessing steps are accurate and meaningful.
- **Visualization.** They provide insights into the structure and density of the data, highlighting potential clusters and relationships even before clustering is applied.

B. *Observations*

- The **semantic similarity heatmap** reveals well-defined alignments, which are essential for clustering semantically related behaviors.
- The **categorical and numerical heatmaps** display distinct diagonal patterns, indicating strong intra-category relationships and coherent numerical groupings.
- The **temporal similarity heatmap** captures periodic patterns that align with the diurnal attack trends discussed in the main analysis.

These heatmaps not only support the methodological rigor of our approach but also enhance the interpretability of the subsequent clustering and analysis steps.

C. *Temporal Analysis.*

As discussed in Section VII, here we present Figure 10 to show the temporal behavior of attackers targeting our deception network.

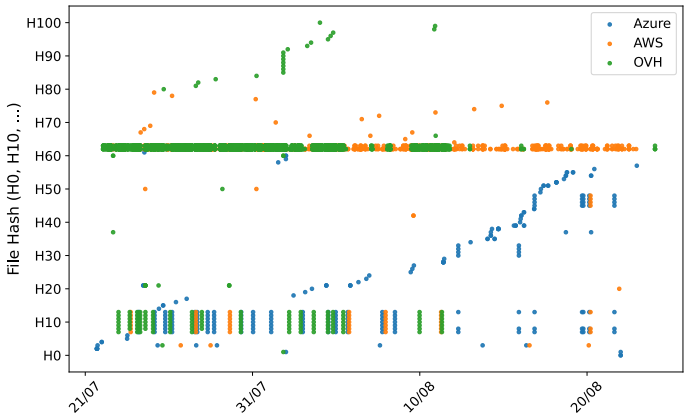
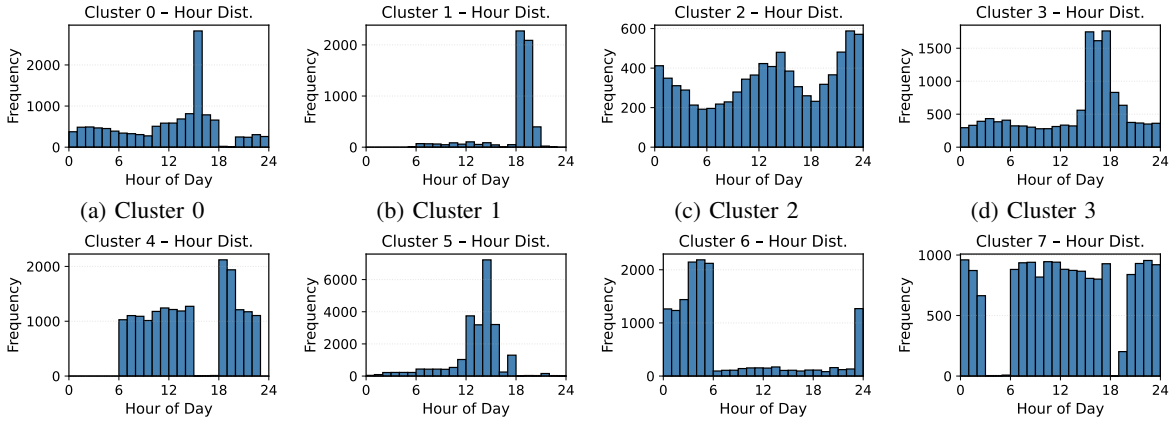


Fig. 9: Timeline of unique file hashes across cloud providers (see §VII).

TABLE VI: Top five malware file hashes observed per cloud provider (AWS, Azure, OVH), including their occurrence count, associated MITRE ATT&CK techniques, and threat level classifications. This highlights coordinated multi-cloud campaigns and provider-specific targeting patterns. (See §VII-C for detailed analysis.)

File Hash	Cloud Provider	Occurrence Count	Malicious Count	MITRE ATT&CK IDs	Threat Level
H8	Azure	50	15.0	T1573, T1071	Informative
H13	Azure	50	28.0	None	None
H7	Azure	47	64.0	T1078, T1059.001, T1078.002, T1547.001, T1543.003	Informative
H12	Azure	42	36.0	T1014, T1543, T1564, T1082, T1036, T1083, T1027	None
H9	Azure	42	38.0	T1573.001	Informative
H62	AWS	3976	28.0	T1071	Malicious
H63	AWS	2099	0.0	T1573.001, T1573, T1105, T1071	Malicious
H8	AWS	50	15.0	T1573, T1071	Informative
H13	AWS	50	28.0	None	None
H7	AWS	47	64.0	T1078, T1059.001, T1078.002, T1547.001, T1543.003	Informative
H62	OVH	3976	28.0	T1071	Malicious
H63	OVH	2099	0.0	T1573.001, T1573, T1105, T1071	Malicious
H8	OVH	50	15.0	T1573, T1071	Informative
H13	OVH	50	28.0	None	None
H7	OVH	47	64.0	T1078, T1059.001, T1078.002, T1547.001, T1543.003	Informative



(a) Cluster 0 (b) Cluster 1 (c) Cluster 2 (d) Cluster 3

(e) Cluster 4 (f) Cluster 5 (g) Cluster 6 (h) Cluster 7

Fig. 10: Hour-of-day activity for each cluster (cf. §VII-A).