EMPIRICAL TELEMETRY-BASED METRICS FOR EVALUATING HONEYPOT REALISM AND DECEPTION EFFECTIVENESS

Teresita Noelia Nunez Migliorisi

Electrical and Computer Engineering, University of Delaware, Delaware, USA

ABSTRACT

Honeypots remain critical tools for cyber deception, adversarial observation, and proactive threat intelligence. However, despite decades of development, the field still lacks a standardized and empirically validated framework for assessing deception effectiveness. Existing studies rely heavily on raw connection counts or ad hoc indicators, limiting reproducibility, comparability, and operational relevance. This paper presents a telemetry-driven methodology for evaluating honeypot realism and deception effectiveness across measurable behavioral dimensions. Using both a baseline cloud honeynet and an Enhanced Realism-Driven Honeynet (ERDH) modeled on a healthcare research environment, it's empirically demonstrated that domain-consistent realism significantly increases attacker dwell time, interaction depth, behavioral diversity, and malware family richness.

KEYWORDS

Honeypots, Deception, Engagement, Telemetry, Metrics, Evaluation, Standardization, NIST.

1. INTRODUCTION

Modern organizations increasingly deploy cyber deception mechanisms to observe adversarial behavior, detect emerging threats, and delay or mislead attackers [1][2]. Technologies such as honeypots, honeytokens, and decoy services are widely used across cloud, enterprise, and critical infrastructure environments to lure adversaries into controlled settings. Despite their widespread adoption, however, defenders still lack reliable and empirical means to determine whether these systems genuinely deceive human attackers or merely attract automated scanning activity. Without rigorous evaluation, organizations risk investing in deceptive infrastructures that consume resources while producing limited intelligence or failing to sustain meaningful adversarial engagement.

Over the years, a broad spectrum of honeypots has been developed, ranging from low-interaction emulations to fully instrumented high-interaction hosts. While this diversity reflects significant engineering progress, it has not been matched by comparable advances in evaluation methodology. The field continues to lack a standardized framework for assessing deception effectiveness and realism across deployments, limiting both scientific comparison and operational decision-making.

1.1. Problem Statement

Existing honeypot evaluations predominantly rely on coarse indicators such as connection counts, IP totals, or malware samples collected. Although straightforward to measure, these metrics

DOI: 10.5121/ijnsa.2026.18101

provide little insight into the quality of attacker engagement. They fail to capture attacker intent, depth of interaction, fingerprinting resistance, or deception persistence, factors that are essential to determining whether a honeypot is effectively deceiving adversaries rather than passively absorbing automated probes. As prior studies have noted, this absence of structured, reproducible metrics has long hindered meaningful comparison across deception systems and limits the operational justification for deploying them [3].

1.2. Proposed Solution

This work introduces a telemetry-based evaluation methodology that derives reproducible metrics of deception directly from honeypot telemetry. By grounding evaluation in observable attacker behavior, the proposed approach enables systematic comparison of realism and deception effectiveness across deployments.

The need for such metrics is further underscored by existing security standards. NIST SP 800-160 Vol. 2 [4] recognizes deception as a key technique for achieving cyber resilience and highlights adversarial engagement as a desirable property, while NIST SP 800-53 Rev. 5[5] introduces deception-related controls. However, neither standard specifies concrete, measurable criteria for assessing quality of deception in practice. Although the importance of deception is widely acknowledged by the community, the absence of operational metrics continues to hinder its systematic evaluation and adoption.

To address these gaps, this study proposes and validates a telemetry-driven methodology for quantifying deception effectiveness through real-world honeynet deployments. The results demonstrate that increased realism, manifested through richer filesystem artifacts, plausible background activity, and domain-consistent context, produces measurable improvements in attacker dwell time, interaction depth, behavioral diversity, fingerprinting resistance, and deception persistence. These improvements aim to provide the foundation for practical, reproducible metrics that can be incorporated into future standards.

To evaluate the proposed methodology, two generations of honeynets: a baseline deployment and an Enhanced Realism-Driven Honeynet (ERDH), were deployed in parallel and instrumented to collect detailed telemetry. Across all measured dimensions, the ERDH consistently outperformed the baseline, yielding longer sessions, more diverse command sequences, and a richer set of malware families. To the best of my knowledge, this work is among the first controlled parallel-deployment studies to empirically demonstrate a causal relationship between realism and deception effectiveness.

The contributions of this work are fourfold:

1. **Empirical telemetry-based evidence** that realism materially increases honeypot engagement, including dwell time, interaction depth, and malware diversity.
2. **A reproducible methodology** for deriving operational deception metrics from honeypot telemetry, including precise definitions and computation procedures.
3. **A domain-specific ERDH blueprint and dataset**, demonstrating that healthcare-themed realism elicits more diverse attacker behavior than generic deployments.
4. **A mapping between telemetry-derived metrics and NIST resilience objectives,** enabling practitioners to evaluate the quality of deception within established security frameworks.

Together, these contributions move the field beyond prior calls for standardized metrics by providing concrete, operational measurements and reproducible evidence of their utility.

The remainder of this work is structured as follows: Section 2 reviews foundational concepts and related work on honeypots and anti-honeypot techniques; Section 3 outlines the methodology and deployment procedures; Section 4 presents results and analysis; Section 5 concludes; and Section 6 provides a roadmap for future research and standardization efforts.

2. BACKGROUND AND RELATED WORK

The battle between honeypot designers and attackers has evolved into a fast-moving arms race, one where the pace of innovation increasingly favors the offensive side. Studies show that attackers are adapting approximately 2.3times faster than current honeypot defenses can respond [6]. While deception systems continue to improve realism and complexity, adversaries now routinely employ automation, machine learning, and layered evasion strategies that outpace traditional detection countermeasures.

Early systems like Honeyd [7] were effective in attracting unsophisticated threats, but today's adversaries can achieve a detection precision of over 94% against static or poorly randomized honeypots using ML classifiers trained in traffic and API behavior [8]. This growing asymmetry highlights three core tensions shaping the current deception landscape as follows:

Stealth vs. Resource Cost

High-interaction honeypots significantly improve stealth-reducing detection by as much as 40%, but this comes at a price. They require up to 3times more computing power, memory, and maintenance overhead compared to low-interaction or static decoys [9]. In large-scale or cloud-based deployments, this cost becomes a limiting factor.

Adaptation Speed Mismatch

Defenders have started adopting AI-driven dynamic deception systems, such as HoneyGAN [10] and ADAPT [11], in an effort to respond faster to attacker adaptation. However, offensive models continue to evolve more rapidly, often leveraging online learning and reinforcement mechanisms that allow them to bypass deception systems faster than defenders can retrain or redeploy them [6]. This widening adaptation-speed mismatch makes it difficult for defenders to maintain long-term stealth.

Cloud, Edge, and IoT Vulnerabilities

Cloud-native and IoT honeypots introduce additional weaknesses. As Surnin et al. [12] demonstrate, these environments often lack the microservice noise, process variability, and telemetry complexity present in real deployments. Such behavioral gaps contribute to detection rates up to 43% higher than those of traditional server-based honeypots, revealing critical blind spots as infrastructures continue shifting toward distributed and containerized architectures.

2.1. Taxonomy of Evaluation Metrics

Evaluating honeypot effectiveness requires metrics that go beyond simple event counts and engage with the deeper behavioral and operational qualities of deception. Traditional metrics often focus on stealth such as detection rate, fingerprinting resistance, and engagement duration,

while performance metrics assess resource overhead, latency, or deployment scalability. More advanced analyses incorporate behavioral entropy, service fidelity, or simulated user activity to quantify believability.

Despite the breadth of available metrics, the field still lacks standardized benchmarks or shared definitions. As a result, comparing systems across studies remains difficult, and claims of realism or effectiveness often rely on incompatible methodologies. The taxonomy in Table 1 consolidates representative metrics identified across the literature, illustrating both the diversity of evaluation approaches and the absence of unified standards.

Table 1. Taxonomy of Evaluation Metrics.

Category	**Metric**	**Description**	**Representative References**
Stealth & Detectability	Detection Rate	Percentage of honeypots correctly identified by attackers (lower is better).	[7, 8, 13, 14]
	Fingerprinting Resistance	Ability to avoid being detected via behavioral or protocol-based fingerprinting tools.	[7, 8, 15]
	False Positives / Negatives	Incorrect classification of systems as honeypots or real hosts.	[8, 11, 14]
	Engagement Time	How long attackers interact with the honeypot before suspecting deception.	[11, 16]
	Behavioral Entropy	Variability in responses; higher entropy suggests less predictability and better stealth.	[14, 17, 18]
Realism & Believability	OS/Service Fidelity	Accuracy in mimicking real operating systems and services.	[7, 10, 19]
	User Interaction Simulation	Ability to mimic human-like activity (e.g., file access, typing delays, network chatter).	[17, 18, 19]
	Session Depth	Complexity and length of attacker sessions, indicating realism.	[16, 18]
Performance & Efficiency	Resource Overhead	CPU, memory, and storage cost of running the honeypot.	[6, 9, 20]
	Network Overhead	Bandwidth or latency added by honeypot communication.	[19, 20]
	Scalability	Ability to deploy and manage honeypots across large or distributed environments.	[6, 10, 12]
	Response Time	How fast the honeypot replies to requests (important for avoiding timing-based detection).	[11, 14]
Intelligence Quality	Threat Coverage	Variety and novelty of threats captured by the honeypot.	[16, 18, 20]
	Data Quality	Signal-to-noise ratio in logged data - whether logs contain useful attacker behavior or noise.	[14, 16]

2.2. Deployment Studies and Behavioral Insights

Operational honeypot deployments have provided valuable insight into attacker behavior, but few have rigorously evaluated deception *quality*. Early multi-regional studies such as HoneyLab [21] captured large volumes of attacks but focused primarily on encounter counts rather than

engagement depth or persistence. Research on IoT botnets, including Mirai [22], offers detailed analysis of malware propagation yet does not examine how realism influences adversarial behavior once inside a honeypot.

More recent work has begun moving toward a richer behavioral evaluation. HoneyFactory [23] introduced a container-based architecture with a formal "deception stage" model, enabling structured analysis of attacker progression. Other scalable honeynet platforms [24] highlight the importance and difficulty of faithfully emulating realistic services at cloud scale. Meanwhile, high-interaction honeypots deployed in operational networks have proven effective for capturing deeper behavioral traces, though evaluations against adaptive adversaries remain limited.

Taken together, these works reveal an important gap: while the community recognizes the need for standardized, behavioral, and realism-aware metrics, existing studies rarely provide empirical evidence linking realism to measurable deception outcomes. This gap motivates the telemetry-focused methodology and experimental design presented in the next section.

3. METHODOLOGY

The research followed an iterative, telemetry-driven methodology designed to (1) establish a baseline model of adversarial activity against unmodified honeypots, (2) evaluate the effect of realistic domain modeling on attacker behavior, and (3) compare these behaviors under controlled, parallel deployment conditions in public cloud environments. The methodology consists of three pillars: instrumentation, experimental design, and metric operationalization.

3.1. Instrumentation and Data Collection

All honeynet nodes were instrumented using a centralized ELK stack (Elasticsearch, Logstash, Kibana)[25]. This enabled a unified collection of fine-grained telemetry across deployments, including timestamps, connection metadata, full command transcripts, malware payloads, session termination patterns, and auxiliary system events. Standardizing the telemetry pipeline ensured that all deployments: baseline, enhanced, and parallel, were evaluated using identical data sources and feature extraction procedures.

Figure 1 presents the telemetry-driven evaluation workflow used in this study, illustrating the progression from honeynet deployment through telemetry collection, session extraction, metric computation, and comparative analysis. All telemetry analyzed in this study was collected by the author from original honeynet deployments conducted in public cloud environments; no external datasets were used.

Figure 1. Telemetry-Driven Honeypot Evaluation Workflow [26]

3.2. Experimental Design

The experiments followed a two-phase design:

1. **Baseline phase:** A default T-Pot [27] deployment with minimal system content was exposed to the Internet for several weeks to characterize typical attack behavior targeting generic cloud-hosted services.
2. **Enhanced phase:** An Enhanced Realism-Driven Honeynet (ERDH) was constructed by augmenting selected honeypots with domain-consistent healthcare artifacts, simulated user activity, and realistic system state. This permitted the evaluation of the extent to which increased realism influences attacker engagement.

To isolate realism as the variable of interest, a controlled parallel experiment was conducted in which a baseline T-Pot deployment and an ERDH deployment were launched simultaneously in the same region of Google Cloud Platform, with identical configurations, and synchronized observation periods. The architecture of this deployment is presented in Figure 2.

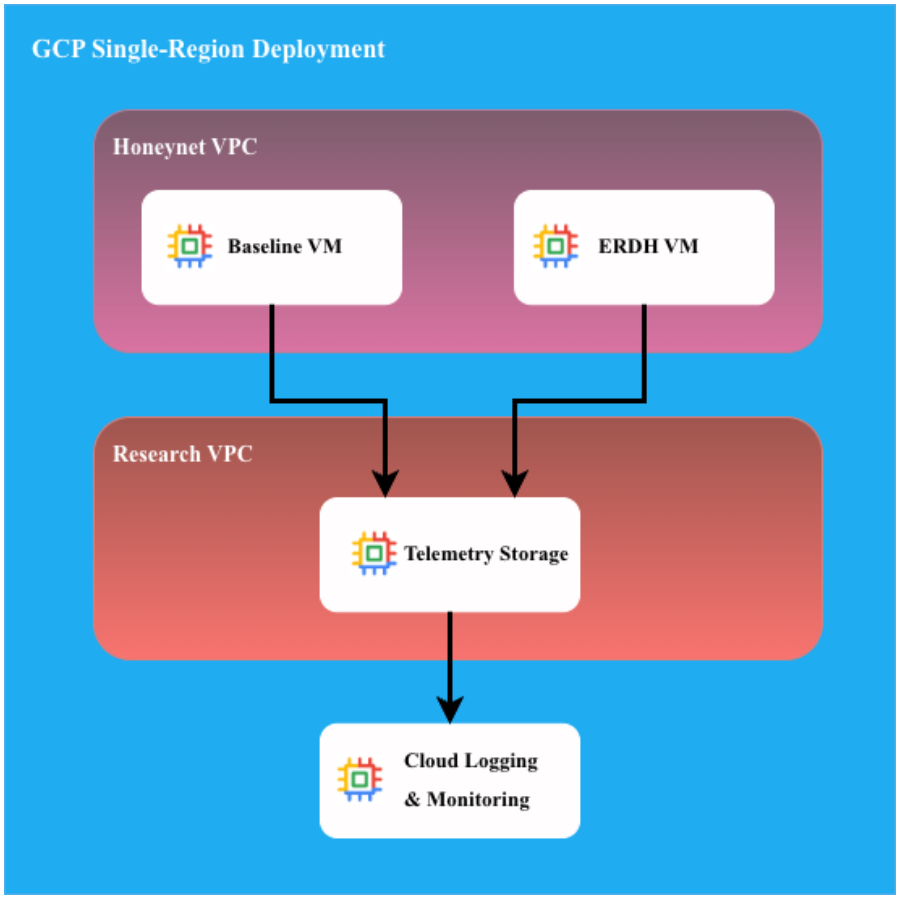

Figure 2. Parallel Deployment Architecture

3.3. Metric Operationalization

Telemetry was converted into quantitative deception metrics to evaluate attacker engagement and behavioral richness:

- **Dwell time:** Duration from session initiation to disengagement.
- **Interaction depth:** Count of distinct attacker actions per session.
- **Behavioral diversity:** Entropy over command categories, TTPs, and malware families.
- **Fingerprinting resistance:** Quantifies the extent to which a honeypot avoids being inferred as deceptive by attackers. It is calculated by applying a heuristic detection function, as defined in Pseudo-code 1, to individual attacker sessions based on observed probing behavior and engagement dynamics. The heuristic distinguishes between detection-induced disengagement, where interaction terminates shortly after probing, and non-deceptive engagement, in which attackers remain active but exhibit only generic behavior. Fingerprinting resistance is then computed as an aggregate deployment-level metric over all sessions, as formalized in Pseudo-code 2.

Pseudo-code 1: DETECTFINGERPRINTING(s, τ): Heuristic Fingerprinting Detection

Input: Session telemetry s, dwell threshold τ
Output: $D(s) \in \{0, 1\}$

$detected \leftarrow 0$; // Assume undetected
if $S.dwell\_time < \tau$ **then**
    $detected \leftarrow 1$; // Early disengagement
else if *S.behavior transitions from Active to Inactive* **then**
    $detected \leftarrow 1$; // Disengagement after probing
else if *S.behavior remains Active* ***and*** *interaction is generic* **then**
    $detected \leftarrow 1$; // Non-deceptive engagement
return $D(s)$

Pseudo-code 2: Computation of Fingerprinting Resistance

Input: Set of sessions $S = \{s_1, \ldots, s_N\}$, dwell threshold τ
Output: Fingerprinting Resistance score FR

$detected\_count \leftarrow 0$
foreach $s_i \in S$ **do**
    $detected\_count \leftarrow detected\_count +$ DETECTFINGERPRINTING(s_i, τ)
FR $\leftarrow 1 - \frac{detected\_count}{|S|}$
return FR

- **Deception persistence:** Measures the ability of a honeypot to sustain attacker engagement beyond initial probing. It is defined as the fraction of sessions in which attacker behavior escalates to higher-impact actions, as determined by the function in Pseudo-code 3 and aggregated across all sessions, as formalized in Pseudo-code 4.

Pseudo-code 3: DETECTESCALATION(s): Session-Level Deception Escalation

Input: Session telemetry s
Output: $E(s) \in \{0, 1\}$

$E(s) \leftarrow 0$; // Assume no escalation
if *Privilege escalation, lateral movement, or payload deployment observed* **then**
    $E(s) \leftarrow 1$; // Escalation inferred
return $E(s)$

Pseudo-code 4: Computation of Deception Persistence

Input: Set of sessions $S = \{s_1, \ldots, s_N\}$
Output: Deception Persistence score DP

$escalated\_count \leftarrow 0$
foreach $s_i \in S$ **do**
  $escalated\_count \leftarrow escalated\_count + \text{DETECTESCALATION}(s_i)$
$\text{DP} \leftarrow \frac{escalated\_count}{|S|}$
return DP

These metrics enable systematic comparison across deployments and provide a foundation for telemetry-based honeypot evaluation.

3.4. Mapping Telemetry Metrics to NIST Resilience Objectives

NIST SP 800–160 Vol. 2 [4] defines four cyber-resiliency objectives: anticipate, withstand, recover, and adapt. The telemetry-derived metrics proposed in this work map naturally onto these objectives. Dwell time and interaction depth provide indicators of an adversary's ability to progress through an attack path, supporting *withstand*. Behavioral diversity and deception persistence inform *anticipate* by characterizing the breadth of techniques adversaries may employ. Fingerprinting resistance contributes to *adapting* by quantifying how well a honeynet avoids detection and forces adversaries to adjust their strategies. Finally, improved telemetry derived from realistic honeynets supports *recovery* by enabling faster incident reconstruction and threat understanding. This alignment situates metrics of deception within an established cyber-resilience framework.

4. RESULTS AND ANALYSIS

The following subsections present empirical findings from each stage of the study, demonstrating how environmental realism and controlled deployment conditions affect attacker engagement, behavior, and malware diversity.

4.1. Local Manual Testing: Identifying Honeypot Cues

Initial testing in a local VMware Fusion environment revealed multiple inconsistencies in default honeypot configurations, such as unsupported commands (e.g. **shutdown now**), incomplete filesystem structures, non-persistent system state, and missing background activity. These cues are subtle to defenders but highly salient to adversaries and automated scanners. This qualitative analysis informed the specific realism enhancements later introduced in the ERDH and highlighted the shortcomings of low-interaction honeypots as deception tools.

4.2. Checkpot-Guided Hardening and Fingerprinting Resistance

Automated evaluation using Checkpot [28] confirmed that baseline honeypots exhibited numerous detectable fingerprints. Iterative hardening such as normalizing banners and enriching the filesystem significantly reduced detection confidence until Checkpot classified the system as a non-honeypot host. This served as a measurable indicator of fingerprinting resistance and provided a validated stealth baseline before deploying realistic enhancements.

4.3. Baseline T-Pot Deployment in Oracle Cloud

The three-week baseline cloud deployment was deployed using Oracle Cloud [29], attracted over 150,000 attacks, dominated by automated SSH brute-force activity and lightweight HTTP

probing. Mean dwell time remained under five seconds, and most sessions involved only one or two commands, indicating rapid disengagement. Malware diversity remained limited, with approximately twenty distinct samples, nearly all associated with low-effort coin-mining campaigns. These results demonstrate that default honeypots collect primarily broad, shallow, and low-value telemetry.

4.4. T-Pot with IP Rotation: Visibility Effects and Reputation Churn

A replicated baseline deployment with periodic IP rotation experienced substantially higher attack volume. This effect reflects the sensitivity of automated threats to IP reputation: rotating IPs circumvents prior black-listing and reintroduces the host into global scanning cycles as a "new" target. While useful for increasing dataset size, this result also shows that raw attack counts are poor indicators of deception effectiveness and must be contextualized within attacker visibility and scanning ecosystem behavior.

4.5. Parallel Deployments in Google Cloud: Isolating the Effect of Realism

The most controlled phase of the study consisted of launching a baseline T-Pot instance and an ERDH instance simultaneously in the same Google Cloud region [30]. With identical network conditions, hardware profiles, and observation windows, this setup isolated realism as the only meaningful independent variable. Within the first 24 hours, the ERDH attracted approximately 50% more attacks, as shown in Figure 3, and exhibited markedly richer behavioral characteristics. Average dwell time increased from seconds to minutes; interactions frequently exceeded five commands, and malware diversity expanded by nearly 50%. These differences cannot be attributed to cloud region, IP aging, time-of-day exposure, or provider reputation. Instead, the results provide strong causal evidence that environmental realism directly enhances adversarial engagement and the quality of collected telemetry.

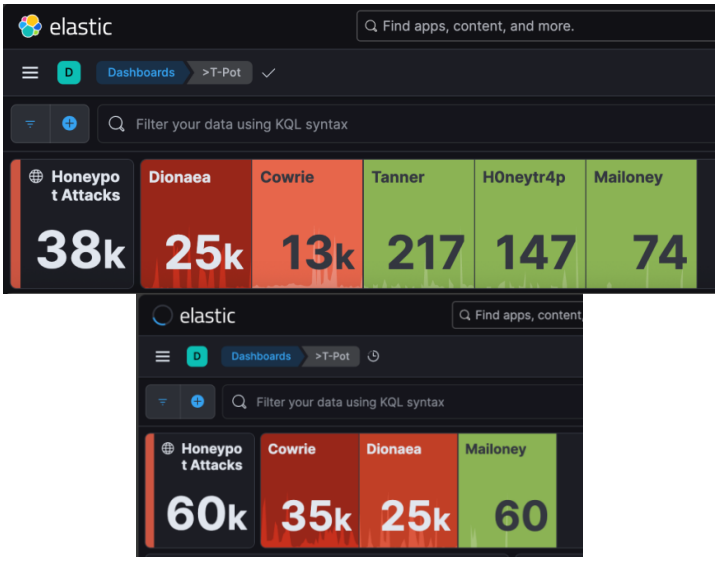

Figure 3. Attacks Comparison: default deployment (top) vs ERDH (bottom)

4.6. Analytical Overhead and Motivation for AI-Driven Telemetry Processing

Across all deployments, the volume of logs and the complexity of correlating attacker actions across sessions posed significant analytical challenges. Manual reconstruction of attacker

workflows required integrating authentication logs, terminal transcripts, malware samples, and network traces, an effort that does not scale with deployment size. This motivates the integration of agent-based AI systems to automatically classify TTPs, cluster sessions, track behavioral evolution, and compute deception metrics at scale, enabling future honeynet research to move beyond manual, labor-intensive workflows.

4.7. Resource Cost vs. Stealth Trade-off

While the ERDH incurred higher computational, as shown in Figure 4, and storage overhead due to richer artifacts and extended session logging, the resulting gains in stealth and engagement demonstrate a favorable trade-off for environments prioritizing intelligence quality over minimal resource usage.

Figure 4. CPU Utilization Comparison: default deployment (ambere-default) vs ERDH (ambere-health)

5. CONCLUSION

This work introduces a telemetry-driven methodology for evaluating honeypot realism and deception effectiveness. Through baseline, enhanced, and controlled parallel cloud deployments, it's shown that domain-consistent realism significantly increases attacker dwell time, interaction depth, behavioral richness, and malware diversity. While default honeypots attract high scan volumes, the results demonstrate that realistic, context-aware environments generate far more meaningful insights into adversarial behavior.

By operationalizing reproducible metrics: dwell time, interaction depth, behavioral diversity, fingerprinting resistance, and deception persistence, provides measurable indicators that move beyond simple connection counts. Mapping these metrics to NIST cyber-resilience objectives further grounds deception evaluation within established security frameworks and enables defenders to assess honeypot effectiveness through the lenses of anticipate, withstand, recover, and adapt.

Taken together, these findings aim to: establish a foundation for standardized, telemetry-based honeypot evaluation; support community adoption and reproducibility through the release of the enhanced honeynet configurations, realistic artifacts, and supporting automation scripts as contributions to The Honeynet Project; and encourage researchers and practitioners to build on this methodology and drive progress toward consistent, resilience-aligned evaluation of cyber deception systems.

6. ROADMAP OF FUTURE WORK

Future work will extend this research along several dimensions:

- **Large-scale ERDH deployments:** Expanding experiments to longer-duration, multi-region, and multi-cloud environments, while exploring sector-specific realism models (e.g., finance, education) informed by threat prevalence in the Verizon's Data Breach Investigations Report [31].
- **Benchmark dataset release:** Publishing sanitized telemetry, annotated sessions, feature extraction scripts, and analysis pipelines to support reproducible comparison and community-driven validation of deception metrics.
- **Metric refinement and composite scoring:**Developing normalization procedures, calibration studies, and multi-metric composite indices that capture honeypot performance across realism, engagement, and evasion dimensions. This includes extending beyond honeypot telemetry to incorporate environmental context.
- **Adaptive deception and autonomous control loops:** Integrating agent-based AI systems capable of dynamically modifying system fingerprints, content, and responses based on real-time adversarial behavior, enabling continuous measurement of adaptive resilience.
- **Standards and community engagement:** Mapping proposed metrics to NIST SP 800–160 and related resilience frameworks, and collaborating with academia, industry, and open-source communities to formalize benchmark practices for honeypot evaluation.
- **Agentic AI for autonomous telemetry analysis:** Implementing multi-agent LLM frameworks to automate cross-log correlation, malware triage, attacker clustering, and TTP extraction. By embedding reasoning-driven AI agents directly into the telemetry pipeline, honeynets can evolve toward self-analyzing systems, reducing human workload while providing richer and more timely insights into adversarial behavior. This capability is essential for operating large-scale, high realism honeynets and supports the eventual development of adaptive deception control loops.

These efforts collectively support the long-term agenda of advancing honeypot research toward rigor, reproducibility, and cross-domain standardization.

REFERENCES

[1] N. Provos, “A virtual honeypot framework,” in Proc. 13th USENIX Security Symposium, 2004.
[2] L. Spitzner, Honeypots: Tracking Hackers. Addison-Wesley, 2003.
[3] Z. Aradi, et al., “Metrics-driven evaluation and optimization of honeypots: Toward standardized measures of deception effectiveness,” Acta Polytechnica Hungarica, vol. 22, no. 12, pp. 1–15, 2025.
[4] R. Ross, M. McEvilley, and J. Oren, “Developing cyber-resilient systems: A systems security engineering approach,” National Institute of Standards and Technology, NIST Special Publication 800-160, vol. 2, Dec. 2021.
[5] Joint Task Force, “Security and privacy controls for information systems and organizations,” National Institute of Standards and Technology, NIST Special Publication 800-53, Rev. 5, Sept. 2020.
[6] Z. Morić, et al., “Advancing cybersecurity with honeypots and deception strategies,” Informatics, vol. 12, no. 1, p. 14, 2025.
[7] T. Holz, F. Freiling, and F. Raynal, “Detecting honeypots and other suspicious environments,” in Proc. 6th IEEE SMC Information Assurance Workshop, 2005, pp. 29–36.
[8] C. Huang, Y. Zhang, and J. Liu, “Automatic identification of honeypot servers using machine learning techniques,” Security and Communication Networks, vol. 2019, Article ID 2627608, 2019.
[9] M. Tsikerdekis, S. Zeadally, and A. Khalil, “Approaches for preventing honeypot detection and compromise,” in Proc. IEEE Global Information Infrastructure and Networking Symposium (GIIS), 2018, pp. 1–6.

[10] R. Gabrys, M. Jasiński, and P. Rosiński, “HoneyGAN pots: A deep learning approach for generating honeypots,” arXiv preprint arXiv:2407.07292, 2024.
[11] V. S. C. Putrevu, A. Sood, and S. Chattopadhyay, “ADAPT: Adaptive camouflage-based deception orchestration for trapping advanced persistent threats,” Digital Threats: Research and Practice, vol. 5, no. 3, pp. 1–35, 2024.
[12] O. Surnin, A. Koucheryavy, and Y. Koucheryavy, “Probabilistic estimation of honeypot detection in Internet of Things environments,” in Proc. IEEE Int. Conf. on Computing, Networking and Communications (ICNC), 2019, pp. 191–196.
[13] S. Srinivasa, D. Fraunholz, and M. Schotten, “Gotta catch ’em all: A multistage framework for honeypot fingerprinting,” Digital Threats: Research and Practice, vol. 4, no. 3, pp. 1–28, 2023.
[14] S. Morishita, K. Kanai, and H. Esaki, “Detect me if you… oh wait: An Internet-wide view of self-revealing honeypots,” in Proc. IFIP/IEEE Int. Symp. on Integrated Network and Service Management (IM), 2019, pp. 134–143.
[15] N. Krawetz, “Anti-honeypot technology,” IEEE Security & Privacy, vol. 2, no. 1, pp. 76–79, 2004.
[16] L. Zobal, M. Ponec, and T. Jirsik, “Current state of honeypots and deception strategies in cybersecurity,” in Proc. IEEE Int. Congress on Ultra-Modern Telecommunications and Control Systems (ICUMT), 2019, pp. 1–9.
[17] N. Naik, S. Jenkins, and P. S. Wang, “A fuzzy approach for detecting and defending against spoofing attacks on low-interaction honeypots,” in Proc. IEEE Int. Conf. on Information Fusion, 2018, pp. 904–910.
[18] D. Fraunholz, F. Pohl, and H. Schotten, “An adaptive honeypot configuration, deployment and maintenance strategy,” in Proc. IEEE Int. Conf. on Advanced Communication Technology (ICACT), 2017, pp. 53–57.
[19] S. Kandanaarachchi, J. Leckie, and S. Krishnan, “HoneyBoost: Boosting honeypot performance with data fusion and anomaly detection,” Expert Systems with Applications, vol. 201, p. 117073, 2022.
[20] K. E. Silaen, R. F. Siahaan, and M. Situmorang, “Usefulness of honeypots towards data security: A systematic literature review,” in Proc. IEEE Int. Workshop on Artificial Intelligence and Image Processing (IWAIIP), 2023, pp. 422–427.
[21] W. Chin, E. Markatos, S. Antonatos, and S. Ioannidis, “HoneyLab: Large-scale honeypot deployment and resource sharing,” in Proc. IEEE Int. Conf. on Network and Service Management, 2009, pp. 381–388.
[22] M. Antonakakis, et al., “Understanding the Mirai botnet,” in Proc. 26th USENIX Security Symposium, 2017, pp. 1093–1110.
[23] T. Yu, J. Zhang, and Y. Li, “HoneyFactory: Container-based comprehensive cyber deception honeynet architecture,” Electronics, vol. 13, no. 2, p. 361, 2024.
[24] M. Rabzelj, B. Slivnik, and G. Kandus, “Designing and evaluating a flexible and scalable HTTP honeypot platform: Architecture, implementation, and applications,” Electronics, vol. 12, no. 16, p. 3480, 2023.
[25] A. S. Shaji and M. M. George, "Elastic Stack: A Comprehensive Overview," 2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Kothamangalam, Kerala, India, 2024, pp. 1-5.
[26] Image generated by ChatGPT, OpenAI, December 19th, 2025, https://chat.openai.com/chat.
[27] Deutsche Telekom Security GmbH, “T-Pot: The all-in-one multi-honeypot platform,” GitHub repository, 2025. Available: https://github.com/telekom-security/tpotce
[28] V. Florea, “Checkpot: Honeypot detection tool,” GitHub repository, 2025. Available: https://github.com/honeynet/checkpot
[29] Oracle, “Oracle Cloud Infrastructure Documentation,” Oracle Cloud, 2025. Available: https://docs.oracle.com/en-us/iaas/Content/General/Reference/aqswhitepapers.htm
[30] K. Dhanagopal and S. He, “Single-zone deployment on Compute Engine,” Google Cloud, 2025.Available: https://docs.cloud.google.com/architecture/single-zone-deployment-compute-engine
[31] Verizon DBIR Team, “2025 Data Breach Investigations Report,” Verizon, 2025. Available: https://www.verizon.com/dbir