A Hybrid Honeypot for Cyber Detection and Deception in Industrial Control Systems

Silvio Russo[1,*], Isabella Marasco[1] and Michele Colajanni[1]

[1]Department of Computer Science and Engineering, University of Bologna, Bologna, Italy

Abstract

Critical infrastructures have been facing sophisticated cyber threats, while operational requirements, safety constraints, and legacy architectures prevent all necessary cybersecurity solutions. Novel defensive mechanisms that do not interfere with production systems are necessary. Cyber deception offers a non-intrusive alternative because it enables defenders to observe adversarial behavior in parallel to real operations. Existing honeypots for Industrial Control Systems (ICS) are largely static and produce recognizable patterns that expert attackers can easily distinguish from real processes. We introduce a context-grounded honeypot architecture that leverages historical process data as a secure reference for generating realistic responses through a Large Language Model (LLM). The system maintains protocol fidelity through deterministic handling of EtherNet/IP traffic, while the LLM generates sensor values conditioned on the current emulated state. A validation layer enforces key physical constraints, ensuring coherence with process dynamics without exposing operational data. To enhance model reliability, we incorporate a Retrieval-Augmented Generation (RAG) mechanism that supplies the LLM with structured protocol and domain knowledge, improving consistency while preserving safety and isolation guarantees. This hybrid approach enables realistic and physically plausible interaction patterns while operating within the performance constraints of industrial environments. Our evaluations indicate that LLMs can augment ICS deception effectively, and that enriched honeypots are suitable for early threat detection in critical infrastructures with no changes to operational systems.

Keywords

Industrial Control Systems (ICS), Cyber Deception, Honeypots, LLM-based Honeypots

1. Introduction

Industrial control systems (ICS) form the operational backbone of modern society by orchestrating the function of critical infrastructures such as power grids, water treatment facilities, and manufacturing plants. A series of high-profile incidents e.g., Colonial Pipeline (2021) [1], the Ukraine power grid attacks (2015, 2016) [2], and TRITON malware (2017) [3]) shows the severe consequences of compromising an ICS, ranging from large-scale operational outages to human safety risks. These systems are increasingly interconnected with enterprise networks and Internet-facing services to support remote management, data aggregation, and predictive analytics, a trend commonly described as convergence of information technology (IT)–Operational Technology (OT) [4, 5]. While this integration offers significant operational and management benefits, it also expands the attack surface by exposing components that were historically isolated. Most industrial environments follow the Purdue Model, a hierarchical architecture separating Enterprise IT (Levels 4–5) from OT (Levels 0–3). At the lower levels reside Programmable Logic Controllers (PLCs), HMIs, drives, and field devices, while supervisory systems such as SCADA servers and historians operate at Level 3. The interface between IT and OT is commonly mediated through an Industrial Demilitarized Zone (IDMZ), whose security posture is increasingly framed by Zero Trust principles. Recent studies [6, 7, 8] highlight the importance of continuous verification, least-privilege access, and layered defenses at this boundary. In parallel, security research has explored cyber deception as a proactive defense strategy. Rather than solely relying on detection and prevention mechanisms, deception introduces controlled, realistic-looking artifacts or systems that lure adversaries into interacting with monitored environments. This approach can slow attackers, reveal their techniques,

Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 09-13, 2026, Cagliari, IT

*Corresponding author.

silvio.russo3@unibo.it (S. Russo); isabella.marasco4@unibo.it (I. Marasco); michele.colajanni@unibo.it (M. Colajanni)

and provide defenders with early indicators of reconnaissance or lateral movement. In ICS architectures, where operational constraints limit the deployment of invasive security controls, early detection of such activity becomes crucial, making deception technologies particularly valuable. The proposed honeypot offers a non-intrusive defensive mechanism that combines two functions: early detection of malicious activity and cyber deception [9]. We are aware that designing realistic honeypots for industrial environments is an open challenge. Template-based systems, such as Conpot [10], rely on static responses that attackers can fingerprint via timing analysis, malformed packet probing, or by testing for inconsistencies between actuator states and sensor readings [11]. Recent efforts explore Large Language Models (LLMs) as a means of increasing fidelity: LLMPot [12] shows that LLMs can emulate Modbus and S7Comm [13] interactions, while ShelLM [14] engages adversaries in realistic SSH sessions. However, effective ICS deception requires responses that remain consistent with underlying physical processes. To address this challenge, we introduce a LLM-based honeypot architecture that utilizes historical process data as a structured context for LLM-driven generation. The system targets EtherNet/IP (ENIP), a widely deployed protocol that remains under-explored in deception research despite its operational relevance. All values returned to the attacker are generated by a local LLM conditioned on this evolving simulated state. Protocol semantics, packet construction, and timing characteristics are implemented deterministically, while a physical validation layer enforces domain-specific constraints. This approach ensures that generated responses remain coherent with process behavior without exposing operational data, an essential requirement when leveraging real process traces for grounding. As a further contribution, we integrate Retrieval Augmented Generation (RAG) [15] to enhance the LLM's consistency and domain alignment. By supplying the model with protocol-specific and process-specific context, RAG improves the reliability of generated values and the stability of the interaction, enabling robust performance even in complex or previously unseen tag queries. Our experiments demonstrate that this combination of deterministic protocol logic, simulated process context, and RAG-enhanced LLM generation yields realistic, physically plausible responses with low latency, by supporting even practical deployment within networked ICS environments. In summary, our results demonstrate that LLMs, when grounded in a structured process context and complemented with targeted retrieval, can effectively augment ICS cyber deception in addition to early detection.
The paper is organized as follows: Section 3 provides background on EtherNet/IP and the SWaT dataset. Section 2 surveys prior work on ICS honeypots and LLM-based deception. Section 4 details the architecture and design rationale. Section 5 presents empirical results. Section 6 outlines conclusions.

2. Related Work

Early ICS honeypots relied on simple emulation to replicate device behavior. Conpot [10] is one of the most widely used platforms, but it offers low-interaction support for protocols such as Modbus, S7Comm, and SNMP through XML-defined device templates. While this enables rapid deployment, the resulting responses are largely static, offering limited adaptability to unexpected queries. As shown by Vetterl and Clayton [11], skilled attackers can identify such systems within minutes through timing analysis and response inconsistencies. Medium-interaction designs improve realism by incorporating more detailed protocol handling. HoneyPLC [16] emulates Siemens S7 controllers capable of executing ladder logic, offering a more convincing interface but still relying on scripted sensor behavior and significant configuration effort. High-interaction honeypots push fidelity further by using real ICS hardware or full-system virtualization [17, 13]. Although highly realistic, these systems require substantial resources, raise safety concerns when executing potentially harmful commands, and remain difficult to extend or adapt to new device types. Their dependence on physical equipment also limits scalability and coverage. A common limitation of traditional approaches is the generation of static responses. These approaches fail to reproduce the nuanced, context-dependent behavior of real-world systems. Our work addresses this limitation by using LLMs to generate dynamic, context-sensitive responses that maintain physical coherence. Franco et al. [18] note that the use of AI in deception environments remains underexplored, and the rapid evolution of LLMs has renewed interest in their potential application to

cyber deception. Early work focused on general-purpose services. ShelLM [14] uses LLMs to create interactive SSH honeypots that engage attackers in realistic command-line sessions, adapting responses based on attacker behavior. The system demonstrates improved engagement time compared to static alternatives, but does not address domain-specific constraints present in ICS environments. LLMPot [12] represents the first application of LLMs specifically to ICS honeypots, targeting Modbus and S7Comm protocols. The authors use GPT-4 to generate register values and coil states, showing that LLMs can produce convincing ICS traffic. However, LLMPot exhibits two significant limitations relevant to our work. First, it relies entirely on OpenAI's commercial API, raising concerns about cost (continuous operation generating thousands of queries daily), privacy (sending operational data to third parties), and availability (dependency on external service). Second, LLMPot lacks mechanisms to ensure physical coherence; the LLM generates values based solely on protocol context, without grounding in real process data or validation against physical constraints. LLMHoney [19] explores LLM-based Web application honeypots. It demonstrates adaptive behavior, but promising deception capabilities focus just on HTTP-based services.

3. Background

This section provides an overview of EtherNet/IP (ENIP) industrial protocol. It is an industrial communication protocol developed by Rockwell Automation and standardized by ODVA. It is widely deployed across sectors such as automotive manufacturing, food processing, pharmaceutical production, and utility infrastructure. Unlike traditional IT-oriented protocols, EtherNet/IP is engineered to support deterministic and time-sensitive communication among industrial controllers, I/O modules, drives, robots, HMIs, and other field devices. ENIP adopts a layered architecture that leverages Commercial Off-the-Shelf (COTS) networking technologies:

- Physical/Data Link: IEEE 802.3 Ethernet.
- Network/Transport: TCP/IP and UDP/IP.
- Application: ENIP encapsulation and the Common Industrial Protocol (CIP).

The protocol supports two complementary communication modes: explicit messaging (request/response over TCP) for configuration and diagnostics, and implicit messaging (real-time cyclic I/O over UDP) for time-critical control tasks. This dual-mode design enables both flexible device management and high-frequency industrial I/O. An explicit EtherNet/IP message consists of an ENIP encapsulation header, followed by a CIP payload. The encapsulation header (24 bytes) includes the command code, length, session handle, status, sender context, and options. The payload contains a CIP message structured as an object-oriented request using CIP paths. Class/instance/attribute addressing is used for generic CIP services, while tag-based operations adopt symbolic path segments. Representative commands include:

- `ListIdentity` ($0x0063$): device discovery returning vendor, device type, and product codes.
- `RegisterSession` ($0x0065$): establishment of an ENIP communication session.
- `SendRRData` ($0x006F$): transmission of encapsulated CIP explicit messages, including tag read-/write operations.

In contrast to protocols such as Modbus (numeric registers) or S7Comm (memory areas), EtherNet/IP supports symbolic, human-readable tag names to address controller variables. Industrial naming conventions e.g., `FIT101` (Flow Indicator Transmitter 101), `LIT201` (Level Indicator Transmitter 201), `P301` (Pump 301), facilitate maintainability but also ease attacker reconnaissance: adversaries may enumerate common naming patterns to infer a plant's I/O structure. A honeypot must therefore respond plausibly to both known and previously unseen tag names to avoid detection via coverage probing.

4. System Design

In this paper, we propose a hybrid architecture that integrates dataset-derived context with LLM-driven response. Our system uses historical process data as context for LLM generation, ensuring that all

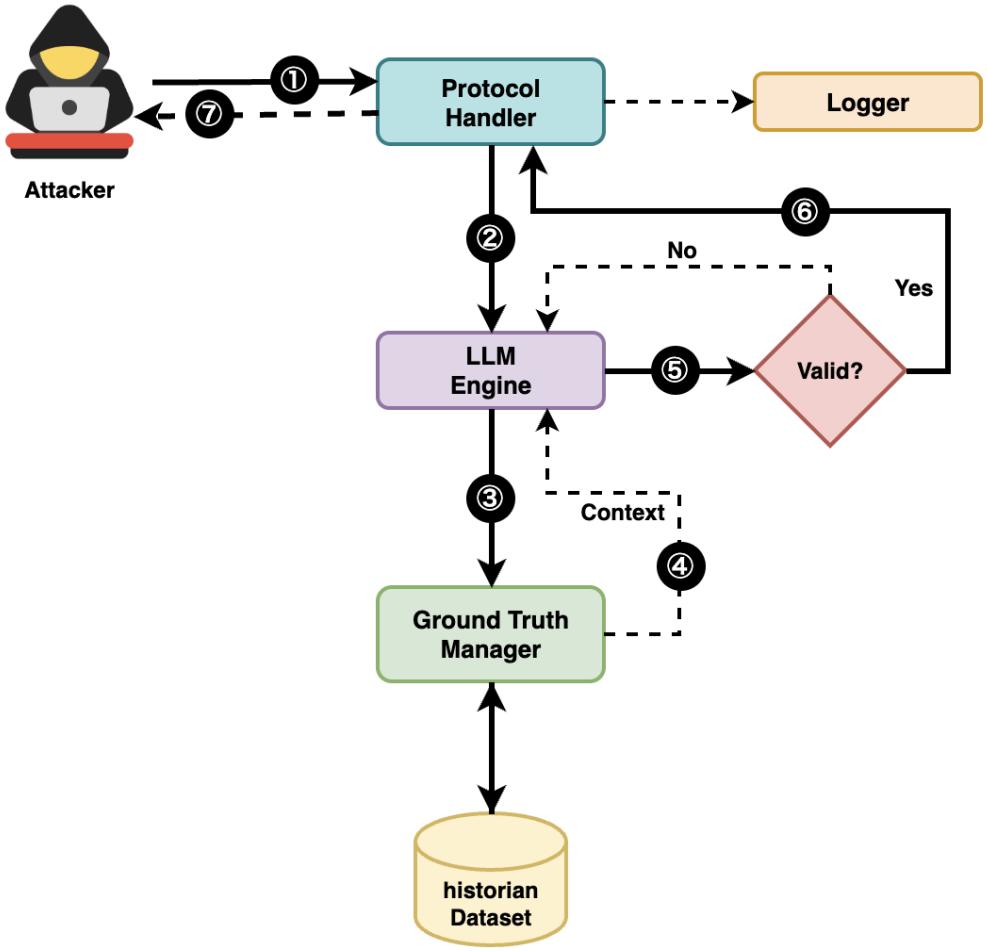

Figure 1: Interaction flow illustrating how the components of the proposed architecture coordinate during request processing.

responses are synthesized and that actual operational data is not disclosed. The architecture balances five objectives: (i) information protection, (ii) physical coherence, (iii) protocol fidelity, (iv) adaptive coverage, and (v) operational performance.

Architecture Overview

The architecture comprises four principal components: the `Protocol Handler`, `Ground Truth Manager`, `LLM Engine`, and `Attack Logger`, whose interactions are depicted in Figure 1. Incoming EtherNet/IP requests are parsed by the Protocol Handler, which delegates all tag value generation to the LLM Engine. The Ground Truth Manager provides contextual state information to guide LLM generation, never directly exposing dataset values, ensuring zero information leakage. All LLM outputs undergo physical constraint validation before transmission.

Protocol Handler. This component implements the EtherNet/IP protocol stack, supporting core encapsulation commands (e.g., `ListIdentity`, `RegisterSession`, `SendRRData`). To emulate PLC behaviour, the handler generates 24-byte ENIP headers, manages session state, and applies realistic communication delays ($50-150$ ms). Protocol semantics and packet structures are produced through deterministic code, preventing protocol-level hallucinations and ensuring authenticity from an attacker's perspective.

Ground Truth Manager. Physical coherence is ensured using historical data of the system. For our tests, we use the SWaT testbed dataset [20], which provides 11 days of operation and 40 process variables. However, the Ground Truth Manager is deliberately designed as a modular and dataset-agnostic component. The subsystem can be reconfigured to ingest any time-series dataset or domain-specific source of process data. The honeypot operates entirely as a standalone deception environment: no live infrastructure is ever queried, and all outputs correspond to internally simulated or replayed states. The manager supports both sequential (cycle) and random sampling modes, and introduces low-level Gaussian noise to reduce fingerprintability. A validation layer enforces domain-appropriate physical constraints, including min/max bounds with safety margins, pump–flow or actuator–sensor correlations, and rate-of-change limits. When violations occur, automatic correction maintains coherent physical behavior. This modular design ensures that the subsystem remains adaptable, allowing the same deception framework to be deployed across heterogeneous environments by simply substituting or augmenting the underlying dataset without modifying core logic.

LLM Engine. All tag-read requests involving process values are handled by the LLM Engine. The model receives the current simulated state from the Ground Truth Manager, enabling context-grounded generation aligned with ongoing process dynamics. This ensures temporal coherence across queries and consistency with multivariable dependencies (e.g., pump–flow relationships). The engine supports both local and API-backed execution, with local deployment preferred for privacy, autonomy, and reduced external dependencies. All generated values are validated against dataset-derived physical constraints before transmission. The engine optionally incorporates Retrieval-Augmented Generation (RAG) [21] for tasks that benefit from structured examples. For protocol-focused testing, RAG provides curated ENIP packet templates with explicit field breakdowns, supporting correct encoding and command structure without revealing sensitive information. For value generation, RAG augments prompts with process-state snapshots sampled from the historical dataset, improving grounding while avoiding disclosure of operational sequences. This approach substantially improves generation quality with respect to baseline prompting, enabling the model to produce more coherent, structured, and physically consistent outputs. Some models achieve near-perfect protocol construction when augmented with relevant examples, while others show more modest gains. This variability highlights the complementarity between learning-based and deterministic components in our design.

General behavior. The architecture ensures temporal and physical consistency across the simulated process while maintaining information protection. Temporal evolution is driven by the Ground Truth Manager operating in cycle mode, which maintains an internal state that follows realistic SWaT dataset transitions, preserving multi-variable dependencies (e.g., pump activations correlate with flow increases, valve closures propagate to downstream pressure drops). For every tag read request, the LLM receives the complete current system state, comprising all 40 process variables, as context. This grounding mechanism constrains LLM outputs to be consistent with ongoing process dynamics, preventing physically implausible responses such as non-zero flows when all pumps are inactive or sensor readings violating conservation principles. Crucially, the LLM synthesizes new values rather than echoing context; physical validation provides a final safeguard, automatically correcting any constraint violations. The result is a unified simulation where all tag values, whether previously in the dataset or not, exhibit coherent behavior aligned with industrial process evolution, maintaining deception effectiveness under sustained reconnaissance without disclosing actual operational data. All interactions are recorded to a lightweight SQLite backend, including timestamps, source IPs, session identifiers, commands, payload sizes, and requested tag names. Since all responses are LLM-generated, logging also captures whether validation corrections were applied. Session-level aggregation supports forensic reconstruction and identification of reconnaissance patterns typical of ICS-targeted adversaries. Read requests follow a unified, LLM-centric processing pipeline designed to prevent information leakage: The Protocol Handler parses the incoming ENIP packet and extracts the requested tag name. The Ground Truth Manager is queried for the current simulated system state, which includes all process variables but is

Table 1
LLM protocol generation success rates (6 tests per model)

Model	Passed	Success Rate
llama3.2:latest (3.2B)	2/6	33.3%
qwen2.5-coder:14B	2/6	33.3%
qwen2.5-coder:7B	1/6	16.7%
deepseek-coder-v2:16B	1/6	16.7%
llama3.1:8B	0/6	0.0%

never directly returned. The LLM Engine receives this state as context and generates a plausible value for the requested tag, synthesizing a response consistent with process dynamics. The generated value undergoes physical validation using dataset-derived constraints; implausible outputs are automatically corrected. The validated value is encapsulated into a properly formatted ENIP response and transmitted. Write requests update the internal simulated state maintained by the Ground Truth Manager to ensure temporal coherence across subsequent reads; updates never propagate to real systems and serve only to maintain realistic process evolution for future LLM context.

5. Evaluation

We evaluate our LLM-augmented honeypot through controlled experiments designed to answer four research questions: whether responses maintain physical plausibility and respect industrial process constraints; whether the system achieves real-time performance suitable for ICS protocol emulation; whether LLMs can reliably generate valid protocol structures, thereby justifying our choice to keep protocol handling deterministic; and whether Retrieval-Augmented Generation (RAG) improves protocol generation capabilities. To assess physical consistency, we executed a 5-minute experiment generating 832 mixed tag requests and validated each response against four key constraints: sensor range bounds, pump–flow coupling, tank level rate limits, and type consistency. The honeypot maintained 99.2% coherence, with all minor deviations (e.g., slight range excesses or integer–float mismatches) automatically corrected before transmission. No pump–flow inconsistencies were observed. These results confirm that context-grounded LLM value generation, combined with a lightweight validation layer, yields responses that remain physically coherent throughout operation. We also assessed latency under the same workload. End-to-end response times averaged 56.8 ms, with the 99th percentile under 70 ms, comfortably within ICS timing requirements (typically < 500 ms). Performance remained stable across the entire experiment, and the system sustained up to 25 req/s before exceeding this threshold. These results indicate that LLM-based value generation with context is fully compatible with real-time ICS honeypot deployment on modest hardware.

Assessing Protocol Generation and the Impact of RAG

The previous experiments assume deterministic protocol handling and LLM-only value generation. To empirically validate this design decision, we first assess whether LLMs can reliably generate binary EtherNet/IP (ENIP) packet headers. We then evaluate whether Retrieval-Augmented Generation can strengthen this capability. We test five models covering different capability classes: two general-purpose models (llama3.2:latest 3.2B, llama3.1:8B), two code-specialized models (qwen2.5-coder:7B, qwen2.5-coder:14B), and one "advanced reasoning" model (deepseek-coder-v2:16B). Each model is evaluated on six protocol generation tasks using distinct prompt strategies, ranging from concrete examples with breakdown, to step-by-step byte construction, explicit validation rules, template filling, real-world PLC scenarios, and code-execution contexts. All prompts operate at low temperature (0.1) and explicitly request exactly 24 space-separated hex bytes representing a valid ENIP header. Table 1 reports success rates. Even the best-performing models (llama3.2:latest and qwen2.5-coder:14B) achieve only 33.3% success (2/6 tests) despite careful prompt engineering. Common errors include incorrect

Table 2
RAG enhancement evaluation across models (3 tests each)

Model	Without RAG	With RAG
llama3.2:latest (3.2B)	33.3%	33.3%
llama3.1:8B	0.0%	66.7%
qwen2.5-coder:7B	16.7%	66.7%
qwen2.5-coder:14B	33.3%	**100.0%**
deepseek-coder-v2:16B	16.7%	33.3%

byte counts, endianness mistakes (e.g., encoding $0x0065$ as "0065" instead of "6500"), inconsistent formatting (extra spaces, newlines), and off-by-one field offsets. While these issues are minor at the textual level, they are catastrophic for binary protocols: a single incorrect byte suffices to invalidate a packet. These results motivated our transition toward RAG/few-shot conditioning, which provides the model with structured exemplars that substantially improve its protocol-related generation capabilities. To investigate whether Retrieval-Augmented Generation can mitigate these limitations, we incorporate a RAG layer that supplies few-shot protocol examples and realistic process context from SWaT. For each protocol generation task, the system: (i) provides three valid ENIP packet examples with field-by-field breakdowns (command codes, byte lengths, little-endian encoding), (ii) includes a snapshot of industrial process state (pump states, flows, tank levels) sampled from SWaT to illustrate realistic ICS conditions, and (iii) constructs a structured prompt that combines examples, process context, and the target packet specification. The process context demonstrates plausible operational states but does not directly determine packet structure; its role is to ground the generation task in realistic ICS scenarios. We test three ENIP tasks (RegisterSession responses, SendRRData responses, and a context-aware variant with dynamic session IDs) across all models, with and without RAG. Table 2 summarizes the results.

RAG yields substantial improvements for several models. The code-specialized qwen2.5-coder:14B reaches 100% success (3/3 tasks), and two other models (llama3.1:8B, qwen2.5-coder:7B) improve to 66.7%. On average, success rates increase from 16.7% to 50.0%, indicating that exposing the model to concrete protocol exemplars and structured retrieval significantly enhances its ability to emit valid binary structures. Gains remain model-dependent: some architectures benefit considerably, while others (e.g., llama3.2:latest) show little or no improvement, and any residual error still produces invalid packets. These findings confirm RAG as a powerful tool for development-time analysis and protocol-oriented prompting, but also underscore that deterministic protocol handling remains the most reliable strategy for a runtime honeypot, where every packet must be valid.

Summary

Our evaluation shows that the proposed architecture allows for maintaining high physical coherence and real-time performance. The protocol-generation experiments reveal that, while baseline LLM prompting remains insufficient, RAG/few-shot conditioning markedly improves model performance, enabling one code-specialized model to reach 100% correctness across tasks. These results demonstrate that LLMs are effective for context-grounded value generation and can be substantially strengthened for protocol-oriented tasks when supplied with structured examples. The evaluation also highlights areas for future work. Our system does not implement a physical process model: it relies solely on lightweight plausibility checks rather than simulating the underlying industrial dynamics. As a consequence, long-term behaviors such as slow drifts, actuator–sensor feedback loops, or multi-minute transients are not explicitly captured, and the honeypot may exhibit simplified temporal evolution. In future work, we plan to integrate a richer process-dynamics component to better emulate long-term physical behavior and further strengthen deception fidelity.

6. Conclusion

Industrial Control Systems operate under strict safety and reliability constraints, yet they remain exposed to increasingly sophisticated cyber threats. Their long-lived architectures, proprietary protocols, and extensive legacy deployments make security upgrades difficult and often infeasible without disrupting critical processes. These limitations increase the value of modern defensive techniques, such as early detection and deception measures that achieve visibility into adversarial behavior without modifying operational systems. Existing honeypots for ICS rarely deliver convincing and physically consistent responses that protect against real attackers. This paper introduces a hybrid deception architecture that combines real process data, LLM-generated sensor values, and deterministic protocol handling validated against physical constraints. Two key insights emerge from our evaluation. LLMs still require strong supervision, real data, constraints, and validation to operate safely in ICS contexts. Moreover, high-quality contextualization matters more than model scale, enabling effective on-premise deployment with moderately sized models. Beyond demonstrating feasibility, the system offers a practical path for organizations to deploy realistic ICS honeypots for early warning and threat intelligence collection. Future developments should extend this foundation toward richer physical models, multi-protocol support, and adaptive behavior informed by attacker interaction patterns, as well as integration with existing defensive tooling. As ICS environments grow more connected and adversaries become more capable, LLMs represent a promising direction for next-generation cyber detection and deception. This work provides an initial architecture, implementation, and methodology to support replication, evaluation, and further advancement within the ICS cybersecurity community.

Acknowledgments

This work was partially supported by the project SERICS (PE000000014) under the MUR National Recovery and Re-silience Plan funded by the European Union - NextGenera-tionEU and by the project C4SI funded by the PR-FESR ER 2021-2027.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Cybersecurity, I. S. Agency, The attack on colonial pipeline: What we've learned and what we've done over the past two years, 2023. URL: https://www.cisa.gov/news-events/news/attack-colonial-pipeline-what-weve-learned-what-weve-done-over-past-two-years, accessed: 2024-11-10.

[2] E-ISAC, Analysis of the Cyber Attack on the Ukrainian Power Grid, Technical Report, Electricity Information Sharing and Analysis Center, 2016.

[3] Dragos Inc., TRITON: Analysis of a Targeted Attack on Industrial Safety Systems, Technical Report, 2017.

[4] G. Murray, M. N. Johnstone, C. Valli, The convergence of it and ot in critical infrastructure (2017).

[5] Cisco Systems, Inc., IT/OT Convergence in Critical Infrastructure and Industrials, Technical Report, Cisco, 2022. URL: https://www.cisco.com/c/en/us/solutions/collateral/industries/manufacturing/itot-convergence-wp.html, white Paper.

[6] C. Zanasi, F. Magnanini, S. Russo, M. Colajanni, A zero trust approach for the cybersecurity of industrial control systems, in: 2022 IEEE Conference on Communications and Network Security (CNS), 2022, pp. 304–312. doi:10.1109/CNS56114.2022.10013559.

[7] N. Jiang, H. Lin, Z. Yin, L. Zheng, Performance research on industrial demilitarized zone in defense-in-depth architecture, in: 2018 IEEE International Conference on Information and Automation (ICIA), 2018, pp. 534–537. doi:10.1109/ICInfA.2018.8812486.

[8] S. Russo, Industrial Demilitarized Zone and Zero Trust cybersecurity models for Industrial Control Systems, Ph.D. thesis, Alma Mater Studiorum, University of Bologna, 2022. URL: https://amslaurea.unibo.it/id/eprint/26737/.

[9] N. Provos, et al., A virtual honeypot framework., in: USENIX Security Symposium, volume 173, 2004, pp. 1–14.

[10] L. Rist, Conpot: An ICS Honeypot, Master's thesis, Applied Security GmbH, 2013.

[11] A. Vetterl, R. Clayton, Bitter harvest: Systematically fingerprinting low-and medium-interaction honeypots at internet scale, in: 12th USENIX Workshop on Offensive Technologies (WOOT 18), USENIX Association, Baltimore, MD, 2018. URL: https://www.usenix.org/conference/woot18/presentation/vetterl.

[12] C. Vasilatos, D. J. Mahboobeh, H. Lamri, M. Alam, M. Maniatakos, Llmpot: Dynamically configured llm-based honeypot for industrial protocol and physical process emulation, 2025. URL: https://arxiv.org/abs/2405.05999. arXiv:2405.05999.

[13] F. Xiao, E. Chen, Q. Xu, S7commtrace: A high interactive honeypot for industrial control system based on s7 protocol, in: S. Qing, C. Mitchell, L. Chen, D. Liu (Eds.), Information and Communications Security, Springer International Publishing, Cham, 2018, pp. 412–423.

[14] M. Sladić, V. Valeros, C. Catania, S. Garcia, Llm in the shell: Generative honeypots, in: 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2024, p. 430–435. URL: http://dx.doi.org/10.1109/EuroSPW61312.2024.00054. doi:10.1109/eurospw61312.2024.00054.

[15] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, 2024. URL: https://arxiv.org/abs/2312.10997. arXiv:2312.10997.

[16] D. I. Buza, F. Juhász, G. Miru, M. Félegyházi, T. Holczer, Cryplh: Protecting smart energy systems from targeted attacks with a plc honeypot, in: International Workshop on Smart Grid Security, 2014, pp. 181–192.

[17] K. Wilhoit, Who's really attacking your ics equipment?, in: Black Hat USA, 2013.

[18] J. Franco, A. Aris, B. Canberk, A. S. Uluagac, A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems, IEEE Communications Surveys & Tutorials 23 (2021) 2351–2383. doi:10.1109/COMST.2021.3106669.

[19] P. Malhotra, Llmhoney: A real-time ssh honeypot with large language model-driven dynamic response generation, 2025. URL: https://arxiv.org/abs/2509.01463. arXiv:2509.01463.

[20] A. P. Mathur, N. O. Tippenhauer, Swat: A water treatment testbed for research and training on ics security, 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater) (2016) 31–36.

[21] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL: https://arxiv.org/abs/2005.11401. arXiv:2005.11401.