Towards agnostic Operational Technology (OT) honeypot fingerprinting

Arthur Cordeiro
Technical University of Denmark
Kgs. Lyngby, Denmark
arur@dtu.dk

Emmanouil Vasilomanolakis
Technical University of Denmark
Kgs. Lyngby, Denmark
emmva@dtu.dk

***Abstract*—Honeypots are versatile cyber-deception tools used to detect and analyze malicious activity across various environments, including operational technology (OT) systems that support critical infrastructure. Their effectiveness, however, depends on remaining undetectable to increasingly sophisticated attackers who employ fingerprinting techniques. This work introduces Aletheia, a framework for fingerprinting OT honeypots agnostically, by reversing the TCP/IP stack. We conduct an Internet-wide scan targeting the Modbus and S7comm protocols, identifying approximately 6 million responsive IPv4 addresses. Applying only a subset of our methodology, we uncover around 7,000 potential honeypot instances—including custom implementations—demonstrating, even at this preliminary stage, how our holistic framework combines established techniques with novel fingerprinting methods to reveal previously unidentified honeypots.**

***Index Terms*—fingerprinting, internet scan, honeypot, OT, cyber-defense**

I. Introduction

In today's increasingly complex cyber-security landscape, deception techniques play a crucial role in defending critical systems against sophisticated attacks [1]. Honeypots, as a key component of cyber-deception strategies, are designed to simulate legitimate systems and attract malicious actors [2], [3]. Their effectiveness hinges on their ability to remain undetected, making the evaluation of honeypots a vital area of research [4], [5]. Notable incidents—such as the Ukraine power grid attacks—have underscored the vulnerability of operational technology (OT) systems and the pressing need for robust cybersecurity defenses [6]–[9]. When carefully crafted to appear authentic, honeypots not only provide early warning of attacks but also yield valuable intelligence on adversary tactics and techniques [10], [11].Yet, attackers are becoming increasingly skilled at fingerprinting and identifying honeypots by exploiting subtle discrepancies in protocol implementations, system behavior, and network characteristics [5], [12], [13].

Honeypot fingerprinting, traditionally associated with banner grabbing, has evolved to include network-level features. While several frameworks exist for detecting honeypots, they often overlook the specific challenges of operational technology (OT) environments [4], [12]. In these unique settings—characterized by legacy systems, multi-protocol support, and highly sensitive workflows—such oversights can lead to inaccurate device simulations and expose OT-specific vulnerabilities [14]–[16]. Addressing these gaps is essential to maintaining the credibility and effectiveness of OT honeypots [17], [18].

This work examines the simulation behavior of Modbus over TCP/IP by analyzing several widely used Python libraries in OT honeypots, including modbus-tk, modsim, pyModbusTCP, pymodbus, and the consolidated honeypot framework Conpot. We study TCP-level protocol fields—such as window size, scaling factors, and TTL values—across various environments to identify patterns that distinguish honeypots from real OT devices. Our analysis reveals specific anomalies that can be exploited for honeypot detection. Building on these findings, we introduce *Aletheia*, a framework that employs protocol-agnostic fingerprinting criteria and integrates tools like ZMap within an OT-oriented methodology [19] to systematically detect OT honeypots. Our contributions include the identification of protocol-level discrepancies, the definition of agnostic detection features, and the development of a reproducible framework aimed at enhancing the reliability of cyber-deception in critical OT infrastructure [3], [20].

II. Related Work

Related work on the field of honeypot fingerprinting has flourished over the last years [12], [21]. Some studies incorporate features like TTL values, providing limited insight into OT environments due to their narrow scope [4]. Others propose detection frameworks referencing OT protocols but lack focus on OT-specific challenges, often using multistage fingerprinting without addressing protocol intricacies or operational constraints [12]. Improvements include modular (hybrid/high)-interaction designs enabling flexible protocol emulation and adaptable detection, enhancing attack capture and realism [14], [22], [23]. However, gaps remain in capturing protocol-level discrepancies and TCP/IP stack behaviors critical for OT honeypot detection. Large-scale scanning tools like ZMap support extensive data collection [19] but often omit OT-specific considerations. This paper presents *Aletheia*, a reproducible, agnostic framework combining context-based and static features tailored to OT environments. By focusing on Modbus simulation in diverse deployments, it reveals protocol-

978-3-903176-74-4

level anomalies and challenges in sustaining effective OT cyber-deception.

III. Detecting OT honeypots

This section outlines our methodology for detecting OT honeypots, focusing on two complementary feature sets. We first examine context-based indicators (Section III-A1) for spotting suspicious deployments, followed by static network features (Section III-A2) that offer environment-independent detection. Section III-B highlights TCP/IP behaviors and protocol-specific traits that produce distinctive signatures, enabling differentiation between real and emulated industrial systems.

A. Detection features

1) Context-based features: Context-based features are primarily logical and highly context-dependent. For instance, an *ASN Check* can reveal whether a device is operating within an expected network boundary or if it is anomalously associated with external or cloud networks —an uncommon trait for field-deployed OT devices. Similarly, the presence of a *cloud environment* indicator strongly suggests suspicious behavior, as genuine OT devices are rarely hosted in virtualized or cloud infrastructures. Additionally, *MAC address* signatures offer valuable manufacturer information, allowing the identification of devices that deviate from expected hardware origins.

2) Static features: While context-based features are protocol-specific, static features provide universal identification metrics applicable across network environments. These foundational elements include *Banner Grabbing* for service identification, *Cross-port validation* for response consistency analysis, and comprehensive *TCP Fingerprinting*. The latter encompasses critical parameters including *Time-to-Live (TTL), Window Size*, and *Window Scaling Size*-TCP/IP stack attributes that yield statistical signatures for device differentiation. Our analysis reveals that Conpot implementations supporting both Modbus and S7Comm protocols present distinctive fingerprinting opportunities. The Modbus specification's ambiguity regarding TCP field handling creates detection vectors, while devices operating both protocols warrant particular scrutiny of TTL values, as genuine Siemens equipment exhibits model-specific TTL patterns. This combination of protocol behaviors and network-layer characteristics constitutes a robust framework for honeypot identification.

The encapsulation relationship between TCP and Modbus protocols is particularly significant for detection purposes. The industrial context introduces unique constraints, many operational OT devices remain unmodified due to cost barriers or operational criticality. Consequently, numerous "unpatched" OT systems persist with legacy TCP connection requirements [24], creating distinctive fingerprints that differentiate authentic systems from emulated environments.

A notable limitation is that many OT devices restrict window scaling values to 65,535 bytes maximum, with most systems strictly adhering to this threshold. This constraint provides a valuable filtering mechanism for device classification. The true TCP window size calculation follows:

$$ScalingWindowSize = WindowSize * 2^S \quad (1)$$

where: $WindowSize$ can vary from [0, $2^{16} - 1$], and 2^S is the scaling factor, which S ranges from 0 to 14.

This mathematical relationship creates distinctive fingerprinting opportunities, as honeypot implementations will often fail to accurately replicate these protocol-specific TCP stack behaviors. The window scaling parameter's implementation varies significantly between virtualized environments and physical OT hardware, offering a reliable differentiator when combined with other TCP/IP characteristics.

B. Understanding Modbus Encapsulation in TCP/IP

We studied the TCP/IP stack to analyze encapsulation behaviors when TCP transports Modbus traffic, revealing implementation inconsistencies critical for OT honeypot identification. Using Wireshark, we captured and analyzed packet-level interactions across multiple Modbus simulations, establishing distinctive TCP fingerprints that differentiate authentic industrial systems from emulated environments.

Our TCP/IP stack analysis, conducted in a controlled Linux environment without external network interference, revealed that Python-based Modbus implementations (*Conpot*, *modbus-tk*, *modsim*, *pyModbusTCP*, and *pymodbus*) consistently show window sizes between 6370-6379 bytes, scaling window sizes around 408,000 bytes, and scaling factors of 64. The TTL value of 64 observed in these tests represents the Linux default, providing a consistent baseline. These characteristics reflect modern TCP implementations with window scaling support. Legacy OT devices typically lack window scaling and remain limited to 65,535-bytes, creating clear differentiation.

Cloud connections differ significantly from local connections in TCP field patterns. Tests across cloud platforms revealed consistent TCP field values regardless of Modbus request type, providing clear differentiation evidence. Cloud-hosted Conpot instances show distinct characteristics: window sizes of 502-509 bytes, calculated window sizes of 64,240-65,152 bytes based on firewall settings, scaling factors of 128-256, and TTL values of 40-47. These anomalies result from cloud-specific network management. Firewalls create network overhead uncommon in OT environments due to their diversity and sensitivity. OT systems support numerous protocols, making additional overhead critical to evaluate.

TCP/IP stack discrepancies open spaces for honeypot detection biases. Local Python implementations share similar characteristics (window sizes 6370-6379 bytes, scaling factors of 64, TTL values of 64), while cloud-hosted variants show different patterns (window sizes 502-509 bytes, scaling factors of 128-256, TTL values of 40-47). The consistent scaling factor and TTL values across all Python implementations-including Conpot-reveals these simulations neglect TCP protocol fidelity for OT devices.

This oversight highlights the honeypots' focus on deceiving attackers at the application layer (Modbus) while ignoring critical lower-layer features integral to authentic Modbus services. These discrepancies effectively identify non-legitimate

OT devices, as genuine legacy systems lack window scaling support and maintain distinctive TCP fingerprints. While not definitive on their own, such indicators substantially enhance our ability to identify emulated environments within operational networks.

IV. Preliminary results

A. Ethical Considerations

Before presenting our scanning results, we address potential ethical implications of our methodology. Network scanning generates traffic that could potentially impact target systems [25]–[27]. To minimize disruption, we implemented several precautionary measures (following considerations from [28]).

First, we utilize ZMap's IP address randomization feature to distribute scan traffic across different network blocks, reducing the probability of overwhelming specific segments. Additionally, we utilize a comprehensive blocklist of public sensitive IPv4 addresses and honored opt-out requests from network administrators who contact us. Furthermore, our scanning protocol adhered to responsible research practices by limiting probe frequency and intensity, ensuring we extracted necessary information while minimizing impact on target systems. These considerations reflect our commitment to balancing knowledge acquisition with network infrastructure integrity.

B. Honeypot Detection

We conducted a comprehensive scan of the IPv4 address space using ZMap, focusing on Modbus (ports 502, 1502, 5020) and S7comm (ports 102, 1102, 10201). The TTL (Time-to-Live) value served as a strong filter when combined with Modbus TCP stack features, creating an effective dual-layer approach [4]. This combination is particularly powerful because Modbus lacks clear specifications for Transport Layer data handling, and most OT honeypots fail to accurately simulate TCP/IP stack behaviors [14]. These implementation gaps create detectable fingerprints that converge to our targets.

It is worth noting that we deliberately included alternative ports beyond the official standards in our scans, as many honeypots use high port numbers by default. This approach aligns with our focus on identifying OT honeypots while studying protocols within the OT scenario.

By applying the filters described in Section III-A, we identified 6,591 IP addresses as potential honeypots across all scanned ports, with exactly 2,084 IP addresses on the standard ports 502 and 102 at the same time (see Figure 1). Our broader scanning revealed approximately 6 million responsive IPv4 addresses, with nearly 1.3 million on port 502 and another 1.3 million on port 102. Interestingly, 488,761 addresses hosted both ports simultaneously. We refined our results by cross-referencing common port numbers and applying the TTL and TCP values outlined in Section III-B.

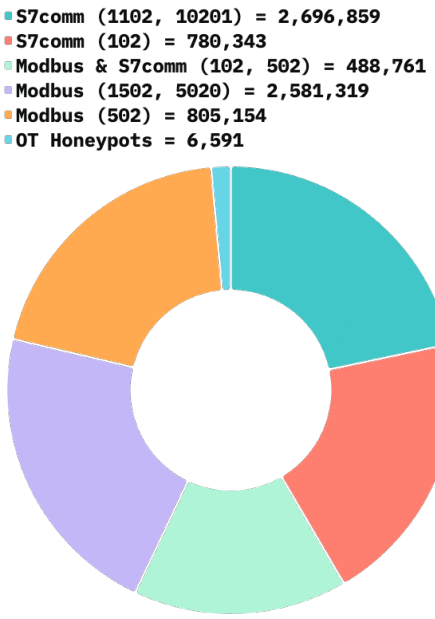

Fig. 1. Preliminary results of our IPv4 scan for Modbus and S7comm along with the identified honeypots

V. Conclusion

We investigate the detection of OT honeypots by identifying implementation flaws that allow attackers to distinguish them from legitimate OT systems. Our analysis shows that while honeypots such as Conpot effectively emulate application-layer protocols, they often overlook critical network-layer characteristics, resulting in detectable fingerprints. Discrepancies in TCP/IP stack parameters—particularly window sizes, scaling factors, and TTL values—serve as reliable indicators for honeypot detection. Local Python-based implementations exhibit consistent patterns (e.g., window sizes of 6370–6379 bytes, scaling factor of 64, TTL of 64) that diverge from those of genuine industrial devices. Cloud-hosted variants display different but equally detectable signatures (window sizes 502-509 bytes, scaling factor 128-256, TTL 40-47).

To address these challenges, we developed *Aletheia*, a framework that systematically combines context-based features (ASN checks, cloud environment indicators, MAC address signatures) with static features (banner grabbing, cross-port validation, TCP fingerprinting) to identify honeypot instances. Our scan of the IPv4 address space, targeting Modbus and S7comm protocols across standard and alternative ports, identified 6,591 potential honeypots, with 2,084 simultaneously hosting both standard ports 502 and 102. This dual-layer approach proved effective because most OT honeypots prioritize protocol simulation over network stack fidelity.

Our analysis improves detection accuracy by cross-referencing common port numbers and applying filters based on TTL and TCP field values, successfully identifying fake OT systems across the IPv4 space. While effective, our methodology remains a work in progress.

Future work will incorporate remaining detection features in strategic combinations, employing intersection-based filtering to enhance precision and reduce false positives. This approach will strengthen honeypot detection while informing the development of more resilient cyber-deception technologies.

ACKNOWLEDGMENT

The work presented here is supported by the Carlsberg Foundation, grant CF23-0843.

REFERENCES

[1] A. Bushby, "How deception can change cyber security defences," *Computer Fraud & Security*, vol. 2019, no. 1, pp. 12–14, 2019.

[2] Spanish National Cybersecurity Institute (INCIBE), *Industrial Honeypot Implementation Guide*, INCIBE-CERT, October 2019, version 1. [Online]. Available: https://www.incibe.es/

[3] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, "Guide to operational technology (ot) security," National Institute of Standards and Technology, Gaithersburg, MD, NIST Special Publication 800-82 Revision 3, September 2023, approved by the NIST Editorial Review Board on 2023-09-20. Supersedes NIST SP 800-82 Rev. 2 (May 2015). [Online]. Available: https://doi.org/10.6028/NIST.SP.800-82r3

[4] J. Williams, M. Edwards, and J. Gardiner, "Time-to-lie: Identifying industrial control system honeypots using the internet control message protocol," *arXiv preprint arXiv:2410.17731*, 2024.

[5] S. Morishita, T. Hoizumi, W. Ueno, R. Tanabe, C. Gañán, 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. 134–143.

[6] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 ukraine blackout: Implications for false data injection attacks," *IEEE transactions on power systems*, vol. 32, no. 4, pp. 3317–3318, 2016.

[7] R. Singla, S. Srinivasa, N. Reddy, J. M. Pedersen, E. Vasilomanolakis, and R. Bettati, "An analysis of war impact on ukrainian critical infrastructure through network measurements," in *2023 7th Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 2023, pp. 1–10.

[8] J. E. Sullivan and D. Kamensky, "How cyber-attacks in ukraine show the vulnerability of the us power grid," *The Electricity Journal*, vol. 30, no. 3, pp. 30–35, 2017.

[9] C.-C. Sun, A. Hahn, and C.-C. Liu, "Cyber security of a power grid: State-of-the-art," *International Journal of Electrical Power & Energy Systems*, vol. 99, pp. 45–56, 2018.

[10] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the mirai botnet," in *26th USENIX security symposium (USENIX Security 17)*, 2017, pp. 1093–1110.

[11] A. Mangino, M. S. Pour, and E. Bou-Harb, "Internet-scale insecurity of consumer internet of things: An empirical measurements perspective," *ACM Transactions on Management Information Systems (TMIS)*, vol. 11, no. 4, pp. 1–24, 2020.

[12] S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, "Gotta catch'em all: a multistage framework for honeypot fingerprinting," *Digital Threats: Research and Practice*, vol. 4, no. 3, pp. 1–28, 2023.

[13] M.-R. Zamiri-Gourabi, A. R. Qalaei, and B. A. Azad, "Gas what? i can see your gaspots. studying the fingerprintability of ics honeypots in the wild," in *Proceedings of the fifth annual industrial control system security (icss) workshop*, 2019, pp. 30–37.

[14] E. López-Morales, C. Rubio-Medrano, A. Doupé, Y. Shoshitaishvili, R. Wang, T. Bao, and G.-J. Ahn, "Honeyplc: A next-generation honeypot for industrial control systems," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 279–291.

[15] M. Dahlmanns, J. Lohmöller, I. B. Fink, J. Pennekamp, K. Wehrle, and M. Henze, "Easing the conscience with opc ua: an internet-wide study on insecure deployments," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 101–110.

[16] A. Mirian, Z. Ma, D. Adrian, M. Tischer, T. Chuenchujit, T. Yardley, R. Berthier, J. Mason, Z. Durumeric, J. A. Halderman *et al.*, "An internet-wide view of ics devices," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2016, pp. 96–103.

[17] M. Conti, F. Trolese, and F. Turrin, "Icspot: A high-interaction honeypot for industrial control systems," in *2022 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2022, pp. 1–4.

[18] C. Vasilatos, D. J. Mahboobeh, H. Lamri, M. Alam, and M. Maniatakos, "Llmpot: Automated llm-based industrial protocol and physical process emulation for ics honeypots," *arXiv preprint arXiv:2405.05999*, 2024.

[19] Z. Durumeric, D. Adrian, P. Stephens, E. Wustrow, and J. A. Halderman, "Ten years of zmap," in *Proceedings of the 2024 ACM on Internet Measurement Conference*, 2024, pp. 139–148.

[20] M. Dodson, A. R. Beresford, and D. R. Thomas, "When will my plc support mirai? the security economics of large-scale attacks against internet-connected ics devices," in *2020 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2020, pp. 1–14.

[21] 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.

[22] S. Srinivasa, J. M. Pedersen, and Vasilomanolakis, Emmanouil, "Riotpot: a modular hybrid-interaction iot/ot honeypot," in *European Symposium on Research in Computer Security (ESORICS)*. Springer, 2021, pp. 745–751.

[23] S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, "Interaction matters: a comprehensive analysis and a dataset of hybrid iot/ot honeypots," in *Proceedings of the 38th Annual Computer Security Applications Conference*, 2022, pp. 742–755.

[24] *MODBUS Messaging on TCP/IP Implementation Guide V1.0b*, Modbus Organization, Inc., 2006, available online. [Online]. Available: https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf

[25] J. O'Hare, R. Macfarlane, and O. Lo, "Identifying vulnerabilities using internet-wide scanning data," in *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*. IEEE, 2019, pp. 1–10.

[26] T. Dai and H. Shulman, "Smap: Internet-wide scanning for spoofing," in *Proceedings of the 37th Annual Computer Security Applications Conference*, 2021, pp. 1039–1050.

[27] G. Wan, L. Izhikevich, D. Adrian, K. Yoshioka, R. Holz, C. Rossow, and Z. Durumeric, "On the origin of scanning: The impact of location on internet-wide scans," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 662–679.

[28] R. Yaben, N. Lundsgaard, J. August, and E. Vasilomanolakis, "Towards identifying neglected, obsolete, and abandoned iot and ot devices," in *2024 8th Network Traffic Measurement and Analysis Conference (TMA)*, 2024, pp. 1–10.