Towards a High-interaction Physics-aware Honeynet for Industrial Control Systems

Marco Lucchese
University of Verona
marco.lucchese@univr.it

Massimo Merro
University of Verona
massimo.merro@univr.it

Federica Paci
University of Verona
federicamariafrancesca.paci@univr.it

Nicola Zannone
Eindhoven University of Technology
n.zannone@tue.nl

ABSTRACT

Industrial control systems (ICSs) play a crucial role in modern society, controlling and automating processes in industries ranging from manufacturing to energy production. The increasing connectivity of ICSs with corporate networks has made them vulnerable to cyber attacks that can compromise the controlled physical processes. We present the architecture of HoneyICS, a high-interaction, physics-aware, scalable, reconfigurable, and extensible honeynet for ICSs, facing most of the limitation of current honeypots for ICSs.

CCS CONCEPTS

• **Security and privacy** → *Intrusion/anomaly detection and malware mitigation*; *Network security*;

KEYWORDS

Cyber-physical systems security, Honeypot, cyber-physical attack

ACM Reference Format:
Marco Lucchese, Massimo Merro, Federica Paci, and Nicola Zannone. 2023. Towards a High-interaction Physics-aware Honeynet for Industrial Control Systems. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), March 27-March 31, 2023, Tallinn, Estonia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3555776.3577803

1 INTRODUCTION

Industrial Control Systems (ICSs) are physical and engineered systems whose operations are monitored, coordinated, controlled, and integrated by a computing and communication core [27]. They often represent the backbone of *Critical Infrastructures* for safety-critical applications such as electric power distribution, nuclear power production, and water supply.

ICSs are increasingly exposed to sophisticated *cyber-physical attacks* [14], i.e., security breaches in cyberspace that adversely affect the physical processes. To defend ICSs from these attacks, it is important to monitor and log remote connections to the *Operational Technology* (OT) network, linking controllers, interfaces and plants.

Honeypots are computer security systems that emulate hardware and software devices and can be used to detect attacks in their initial phase and to collect information about the techniques used by attackers in order to select appropriate mitigations [21]. Although honeypots for ICSs [3, 6, 8–11, 19, 20, 23, 28, 30, 31, 33] have done several progresses in the last years, they still have limitations regarding the following features that we deem important.

- *Level of interaction*: ICS honeypots should be able: (i) to return accurate fingerprints of the devices and the underlying industrial network (low-interaction), and (ii) to allow the attacker to interact with the honeypot (high-interaction) providing consistent simulation of physical feedback to attackers' actions.
- *Configurability*: including *extensibility*, to emulate different models of PLCs, and the possibility to adopt different industrial network protocols, depending on the context of use.
- *Scalability:* ICS honeypots should be able to simulate real-world ICSs, which often comprise hundreds of devices.
- *Entry point*: to support attacks that may gain access to the honeypot either by compromising the VPN to which it is connected or by exploiting devices directly exposed to the Internet. In the former case, ARP poisoning techniques allows us to mount non-trivial MITM attacks on the OT network.

Contribution. We propose the architecture of a new honeypot framework for ICS, called HoneyICS. HoneyICS is a high-interaction and physics-aware honeynet emulating an OT network of PLC and HMI honeypots rather than just a single PLC honeypot, like in most existing frameworks [17]. HoneyICS supports high extensibility as it is able to emulate different brands of PLCs. HoneyICS is a high-interaction honeypot as it allows the attacker to modify PLC registers, HMI interfaces, and the user program executed by the PLCs. Moreover, HoneyICS emulates physical plants connected to PLCs and it is thus able to provide realistic feedback to attackers' commands. Last but not least, the attacker has full control of the network connecting PLCs and HMIs to mount MITM attacks.

2 REQUIREMENTS FOR ICS HONEYPOTS

Honeypots are technical countermeasures that can support a multi-layered approach to ICS security. They are computer security systems that emulate hardware and software devices and can be used to decoy attackers away from the real system, to educate staff, and to study attack patterns [21]. A system consisting of two or more honeypots is called *honeynet*.

SAC '23, March 27-March 31, 2023, Tallinn, Estonia

ACM ISBN 978-1-4503-9517-5/23/03.
https://doi.org/10.1145/3555776.3577803

Table 1: Comparison with other ICS honeypots

✗= Not supported ◒= Partially supported ✓= Fully supported

	Level of Interaction						Configurability		Scalability	Honeypot
Honeypot	ICS Network simulation	Physics-aware	PLC registers	Code inj.	HMI	MITM	ICS protocols	Extensibility		Entry Point
SCADA Honeynet [33]	◒	✗	✗	✗	✗	✗	Modbus	◒	◒	Internet
Conpot [20]	◒	✗	✗	✗	✗	✗	Modbus, S7comm, BACnet, EtherNet/IP	◒	◒	Internet
Dipot [16]	◒	✗	✗	✗	✗	✗	Modbus, S7comm, BACnet	◒	◒	Internet
HosTaGe [10]	◒	✗	◒	✗	✗	✗	Modbus, S7comm	✗	◒	Internet
Pliatsios et al. [7]	◒	✗	◒	✗	✓	◒	Modbus	◒	◒	VPN
Honeyd+ [23]	◒	✗	✗	✗	✗	✗	EtherNet/IP	◒	✓	Internet
THS [28]	◒	✗	◒	✗	✗	✗	Modbus, S7comm, BACNet	◒	◒	Internet
CryPLH [8]	◒	✗	✗	✗	◒	✗	S7comm	✗	◒	Internet
HoneyPhy [31]	◒	✓	✗	✗	◒	◒	DNP3	✗	✗	Internet + VPN
GasPot [19]	✗	✗	✗	✗	✗	✗	-	✗	◒	Internet
Antonioli et al. [6]	✓	✓	✗	✗	✓	✓	EtherNet/IP	✗	◒	VPN
Murillo et al. [3]	◒	✓	✗	✗	✗	◒	EtherNet/IP	✗	◒	VPN
MimePot [11]	◒	✓	✓	✗	✗	◒	Modbus	✗	◒	VPN
HoneyPLC [9]	◒	✗	✗	◒	✗	✗	S7comm	✓	◒	Internet

Operational Technology (OT) networks of ICSs usually include devices, systems, networks, and controllers used to operate and/or automate industrial processes. They consists of field devices such as sensors and actuators that monitor and control the evolution of a physical process overtime, programmable logic controllers (PLCs) that control the field devices, and one or more human machine interfaces (HMIs), which allow human operators to interact with PLCs and display status information and historical data gathered by the field devices in the ICS environment. OT networks include two main sub-networks: the *supervisory control network* to connect PLCs and HMIs, and the *field communications network* to link the PLCs with the associated field devices. Modern ICS systems are interconnected through a variety of network industrial protocols, such as Modbus [4], DNP3 [12], EtherNet/IP [26], OPC UA [29], and S7comm [2].

Thus, honeypots for ICSs, which effectively emulate OT networks, have to face a number of non-trivial challenges to provide the following features that we deem essential to deceive attackers.

Level of interaction. Honeypots and honeynets are usually classified based on the *level of interaction* that they allow to the attacker. Low-interaction honeypots emulate one or more services with simple functions. While high-interaction ones emulate the behavior of real devices and are thus suitable to collect information about attacker's actions. An ICS honeypot should be able to simulate an industrial network connecting an arbitrary number of communicating PLCs, possibly supervised via HMI interfaces, and supporting an observable and accessible network traffic involving PLCs and HMIs. Thus, an ICS honeypot should be not only able to return accurate fingerprints of the involved devices and ICS networks (*device and network simulation*) as is the case for low-level interaction ICS honeypots [8, 28], but it should also allow the attacker to interact with the honeypot, for example, inspecting and modifying PLC registers, uploading malicious PLC code, inspecting and exploiting HMI interfaces, and basically gaining full control over the OT network. In addition, as pointed out in [22, 31], *physics-awareness* is a crucial ingredient to realize convincing and deceiving ICS honeypots; this means the attacker should receive consistent feedback from a (possibly simulated) manipulated physical process.

Configurability. The honeypot should allow to change the attack surface exposed to an attacker in order to adapt to evolving attackers' exploit tools and techniques as well as to the ICS network to be protected. Thus, possibly, the honeypot should be able to support different *industrial network protocols*, depending on the context of use. It should also be *extensible* to support the simulation of PLCs of different brands and models.

Scalability. To simulate real-world ICSs, the honeypot should *scale* to middle-size ICSs with hundreds of PLCs and HMIs of different kinds without affecting the performance. Although the adoption of virtual resources (rather than physical devices) is a necessary condition for scalability, it does not necessarily ensure scalability. To ensure scalability, ICS honeypots should be empirically tested (at varying number of PLCs and HMIs) to assess their response time.

Honeypot Entry Point. A convincing ICS honeypot should support attacks to availability and integrity of the target system. The attacker might gain access to the honeypot and its components either by compromising the VPN to which the honeypot is connected or by exploiting devices directly exposed to the Internet. The entry point has an impact on the attacker's capabilities. In case the attacker is able to access the honeypot via the Internet, she can try tamper the exposed PLCs and HMI interfaces (eventually after a brute-force attack on their authentication). On the other hand, in case the attacker is able to compromise the VPN under which the honeypot runs, she can do ARP poisoning in order to sniff network traffic on the supervisory control network, and to set up MITM attacks between two PLCs or between a PLC and the associated HMI.

3 LIMITATIONS OF ICS HONEYPOTS

Several honeypots and honeynets have been proposed in the literature. Table 1 provides an overview and a comparison of related work, highlighting their limitations, according to the desired requirements for an ideal ICS honeypot, as discussed in Section 2.

Level of interaction. Current approaches mostly provide limited functionality when it comes to TCP/IP stack simulations, as well as native ICS network protocols. This poses serious limitations in the actions an attacker can perform within the honeypot and, thus, in the understanding of adversarial interactions and malware. The exhibition of a convincing *traffic* in the network connecting PLCs with supervising HMIs is crucial to convince an attacker on the authenticity of the targeted ICS. In this respect, Antonioli et al. [6] provide the possibility to build up a communication network between PLCs and/or HMIs, while HoneyPhy [31] only propose an ideal architecture where such communication is possible. As a consequence, only Antonioli et al. [6] support non-trivial MITM attacks between PLCs and/or HMIs; more limited forms of MITM attacks, between PLCs and their plant, can be simulated in [3, 7, 11, 31]. Among the reviewed honeypots, only [7, 10, 11, 28] explicitly support some form of register manipulation, and only [6–8, 31] explicitly support some form of HMI manipulation. As regards *physics-awareness*, only the works in [3, 6, 11, 31] provide some form of simulation of the underlying physical industrial processes. Moreover, only HoneyPLC [9] is able to simulate the upload of malicious user programs, although the injected code is only stored by the honeypot but not executed. While capturing the code is a first important step to support PLC malware analysis, the execution of the injected code together with consistent physical feedback is crucial to deceive the attacker.

Configurability. All honeypots discussed in Section 3, but [9, 20, 28], support only limited extensibility because they can impersonate only one or two PLC models returning the corresponding fingerprints. Similarly, most works, except for [10, 16, 20, 23], only support a limited number of ICS network protocols. This may significantly limit the ICS network they can emulate and, thus, context in which they can be deployed.

Scalability. Most of the reviewed ICS honeypots have scalable designs because they adopt virtual resources and/or lightweight virtualization techniques such as Docker containers [5]. However, only Honeyd+ [23] provides explicit evaluation of the proposed honeypot, in terms of the number of supported virtual PLCs.

Attacker Entry Point. Our analysis of the literature has shown that existing honeypots have been either exposed on the Internet [8, 10, 16, 20, 23, 28, 33] or protected via a VPN [3, 6, 7, 11]. Although exposing the honeypot on the Internet can provide the attacker with an easier access to the honeypot, it limits the type of interactions that the attacker can perform with the honeypot (cf. Section 2). On the other hand, while providing more useful information on how the attacker can attempt to compromise the ICS network, the employment of a VPN might discourage the attacker as she has to go through an additional line of defense. We advocate that a honeypot should support both entry points to capture a larger spectrum of adversarial interactions. In this respect, only HoneyPhy [31] has been designed to supports both kinds of entry points.

4 OUR PROPOSAL

In this section, we propose HoneyICS, a high-interaction ICS honeynet supporting a non-trivial simulation of OT industrial networks. In the following, we describe the underlying architecture of our honeynet framework and the supported attacker model; then we provide guidelines for its implementation in the next section.

4.1 Honeynet architecture

HoneyICS can emulate the key components of OT networks: PLCs, HMIs, communication networks, and physical plant. The whole honeynet framework is managed and supervised via a management dashboard. Fig. 1 presents the architecture of our honeynet along with its components and entry points.

PLCs. To support a realistic interaction with PLC devices, HonyICS combines the capabilities of *both* low-interaction and high-interaction physics-aware honeypots. Specifically, network simulation (i.e., *low-level interaction*) is achieved using a *personality engine* returning accurate fingerprints matching the profiles of the target PLC. At the same time, the *high-interaction* physics-aware honeypot implements ICS network protocols supporting specific commands to modify PLC registers. This allows attackers to compromise the underlying physical process by reading/writing the value of PLC registers and/or manipulating the user program of PLCs.

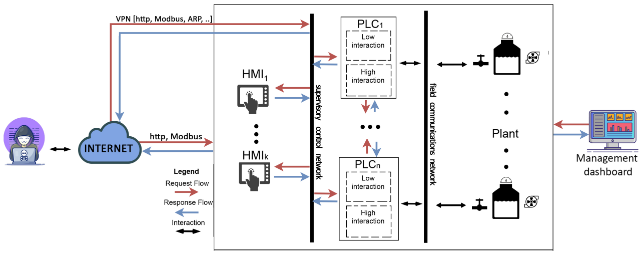

Figure 1: HoneyICS architecture

HMIs. The presence of HMIs allows us to support attacks where the attacker gains control of the HMI, for example via password brute-force attacks or via phishing, and is able to send actuator commands directly to the PLCs.

Physical plant. We assume in our honeynet architecture the presence of physical processes to realize *physics-awareness*, i.e., a convincing physical evolution of the compromised system that can be observed by the attacker. Such processes could consist in either real physical devices or simulations done via classical tools.

Communication network. The previous components are connected via a communication network, which is divided into *supervisory control network*, connecting PLCs among them and PLCs with HMIs (this network is devoted to exhibit a realistic network traffic), and *field communication network*, connecting the PLCs to the physical plant (usually transparent to the attacker).

Management Dashboard. The honeynet is managed via a web dashboard to simplify its deployment and configuration. It enables the user to see what honeynets are currently running, add new honeynets, stop old ones and see/analyze the logs for each of the honeynets running. It also allows to compare what is displayed by an HMI interface with the real state of the physical process. This allows an operator to detect MITM attacks where an attacker feeds the HMIs with faked data which do not correspond with the actual evolution of the plant.

4.2 Attacker model

As shown in Fig. 1, our honeynet supports both Internet exposure and VPN protection. In particular, the attacker can access our honeynet either via the Internet, by finding specific devices (PLCs and HMIs) via search engines, such as Google, Shodan [18], etc., or by gaining access to the VPN to which the honeypot is connected. In the latter case, the attacker can take full control of the supervisor control network connecting PLCs and HMIs, although she does not have direct access to the field communications networks.

Once she has gained access, the attacker may fingerprint the target PLCs, using tools such as Nmap [13], to obtains basic system information (e.g., PLC model and brand, open and filtered ports, the services running on those ports, and the supported industrial protocol). In a subsequent step, the attacker may attempt to read and write PLC memory registers and to upload and execute a malicious PLC user program. The attacker may be able to observe the effect of the program execution either by examining the value of the PLC registers or via a compromised HMI. Similarly, another possible attack vector is to fingerprint and then brute-force HMI interfaces to tamper with the physical state of the system by directly sending commands through the HMI interfaces.

In case the attacker is able to compromise the VPN under which the honeypot runs, then she will also be able: (i) to sniff network traffic on the supervisory control network; (ii) to set up MITM attacks between two PLCs or between a PLC and the associated HMI. In the latter case the attacker may achieve a two-fold objective: on one hand she can manipulate PLC registers (such as those used to store actuator commands or sensor measurements) to bring the physical process into a compromised state; on the other hand the attacker may transmit fake measurements to the corresponding HMI; these measurements may possibly come from previous recordings made by the attacker on the genuine target system during an eavesdrop phase.

4.3 Implementation guidelines

We plan to realize a prototype implementation of HoneyICS based on Modbus [4], DNP3 [12] and other ICS network protocols. To support configurability and scalability, we aim to rely upon existing simulation frameworks such as *Honeyd* [25], *HoneyPLC* [9], *OpenPLC* [32], and *Simulink* [24], rather than physical ICS hardware/devices, where each component of our architecture will be deployed in a dedicated Docker container [5].

To simulate a PLC, we can leverage existing low-interaction and physics-aware high-interaction honeypots. For example, the realization of the low-interaction honeypot can rely upon Honeyd [25] to provide a *personality engine* able to simulate the TCP/IP stack of target devices such as PLCs. On the other hand, OpenPLC [32], an open source software PLC compliant with the IEC 61,131-3 standard [15], can be used to implement the physics-aware high-interaction honeypot. This allows us to leverage the *network layer* of OpenPLC to establish and maintain network connections over ICS network protocols. To route network requests coming either from scanning tools, such as Nmap, or via ICS network protocols to the proper honeypot, OpenPLC can be integrated into Honeyd using Honeyd's *subsystem virtualization* feature [25]. HMI components can be implemented using some open source drag-and-drop SCADA interface that can interact with several PLC brands, such as *ScadaBR* [1]. On the other hand, the physical process of the plant can be simulated in *Simulink* [24], a framework to model, simulate and analyze cyber-physical systems, widely adopted in industry and research.

Finally, the *supervisory control network* of the honeynet can be simulated using a *broker* that connects PLCs with each other through the ICS network protocols. Similarly, a second broker can be used to emulate the *field communications network* connecting the PLCs with the physical plant.

ACKNOWLEDGMENTS

We thank Daniele Antonioli and Nils O. Tippenhauer for comments on an early draft of the paper, and the anonymous reviewers for their useful reviews. The authors have been partially supported by the project "Dipartimenti di Eccellenza 2018–2022" funded by the Italian Ministry of Education, Universities and Research (MIUR).

REFERENCES

[1] 2009. The ScadaBR project. https://www.scadabr.com.br/

[2] 2016. S7comm - The Wireshark Wiki. https://wiki.wireshark.org/S7comm/ Accessed: 2022-05-14.

[3] A. Murillo, L. Combita Alfonso, A. Gonzalez, S. Rueda, A. Cardenas, and N. Quijano. 2018. A Virtual Environment for Industrial Control Systems: A Nonlinear Use-Case in Attack Detection, Identification, and Response. In *ICSS*. 25–32.

[4] A. Swales. 1999. Open Modbus/TCP specification. *Schneider Electric* 29 (1999), 3–19.

[5] C. Boettiger. 2015. An Introduction to Docker for Reproducible Research. *SIGOPS Oper. Syst. Rev. ACM* (2015), 71–79.

[6] D. Antonioli, A. Agrawal, and N.O. Tippenhauer. 2016. Towards High-Interaction Virtual ICS Honeypots-in-a-Box. In *CPS-SPC*. ACM, 13–22.

[7] D. Pliatsios, P.G. Sarigiannidis, T. Liatifis, K. Rompolos, and I. Siniosoglou. 2019. A Novel and Interactive Industrial Control System Honeypot for Critical Smart Grid Infrastructure. In *IEEE CAMAD*. 1–6.

[8] D.I. Buza, F. Juhász, G. Miru, M. Félegyházi, and T. Holczer. 2014. CryPLH: Protecting Smart Energy Systems from Targeted Attacks with a PLC Honeypot. In *Smart Grid Security*. Springer, 181–192.

[9] E. López Morales, C. Rubio, A. Doupé, Y. Shoshitaishvili, R. Wang, T. Bao, and G-J. Ahn. 2020. *HoneyPLC: A Next-Generation Honeypot for Industrial Control Systems.* ACM SIGSAC, 279–291.

[10] E. Vasilomanolakis, S. Srinivasa, C.G. Cordero, and M. Mühlhäuser. 2016. Multi-stage attack detection and signature generation with ICS honeypots. In *NOMS IEEE*. 1227–1232.

[11] G. Bernieri, M. Conti, and F. Pascucci. 2019. MimePot: a Model-based Honeypot for Industrial Control Networks. In *IEEE SMC*. 433–438.

[12] G. Clarke, D. Reynders, and E. Wright. 2004. *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems.* Newnes, Elsevier.

[13] G. Lyon. 1997. Nmap. https://nmap.org/

[14] Y. Huang, A. A. Cárdenas, S. Amin, Z. Lin, H. Tsai, and S. Sastry. 2009. Understanding the physical and economic consequences of attacks on control systems. *Int. J. Crit. Infrastructure Prot.* 2, 3 (2009), 73–83.

[15] International Electrotechnical Commission. 1993. Programmable controllers-Part 3 : Programming languages. *IEC 61131-3* (1993).

[16] J. Cao, W. Li, J. Li, and B. Li. 2018. *DiPot: A Distributed Industrial Honeypot System.* Springer, 300–309.

[17] J. Franco, A. Aris, B. Canberk, and A. Selcuk Uluagac. 2021. A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems. *IEEE Commun. Surv. Tutorials* (2021), 2351–2383.

[18] J. Matherly. 2015. *Complete guide to Shodan.* Shodan LLC.

[19] K. Wilhoit and S. Hilt. 2015. The GasPot Experiment: Unexamined Perils in Using Gas-Tank-Monitoring Systems. In *Trend Micro*, Vol. 6. 3–13.

[20] L. Rist, J. Vestergaard, D. Haslinger, A. De Pasquale, and J. Smith. 2013. Conpot ICS/SCADA Honeypot. http://conpot.org/

[21] M. Dodson, A.R. Beresford, and M. Vingaard. 2020. Using Global Honeypot Networks to Detect Targeted ICS Attacks. In *CyCon*. 275–291.

[22] M. Krotofil, K. Kursawe, and D. Gollmann. 2019. Securing Industrial Control Systems. In *Security and Privacy Trends in the Industrial Internet of Things*, Cristina Alcaraz (Ed.). Springer, 3–27.

[23] M. M. Winn. 2015. *Constructing Cost-Effective and Targetable ICS Honeypots Suited for Production Networks.* Master's thesis. Air Force Institute of Technology.

[24] MATLAB. 2021. version R2021a.

[25] N. Provos. 2003. Honeyd: A Virtual Honeypot Daemon (Extended Abstract). *DFN-CERT* 2 (2003).

[26] P. Brooks. 2001. Ethernet/IP-industrial protocol. In *ETFA*, Vol. 2. 505–514.

[27] R. Rajkumar, L. Lee, I. Sha, and J. A. Stankovic. 2010. Cyber-physical systems: the next computing revolution. In *DAC*. ACM, 731–736.

[28] S. Abe, Y. Tanaka, Y. Uchida, and S. Horata. 2018. Developing Deception Network System with Traceback Honeypot in ICS Network. *SICE JCMSI* 11 (2018), 372–379.

[29] S-H. Leitner and W. Mahnke. 2006. OPC UA–service-oriented architecture for industrial applications. *ABB Corporate Research Center* 48, 61-66 (2006), 22.

[30] S. Arndt S. Lau, J. Klick and V. Roth. 2016. POSTER: Towards Highly Interactive Honeypots for Industrial Control Systems. In *CCS*. ACM, 1823–1825.

[31] S. Litchfield, D. Formby, J. D. Rogers, A. P. S. Meliopoulos, and R. A. Beyah. 2016. Rethinking the Honeypot for Cyber-Physical Systems. *IEEE Internet Comput.* 20 (2016), 9–17.

[32] T.R. Alves, M. Buratto, F.M. Souza, and T.V. Rodrigues. 2014. OpenPLC: An open source alternative to automation. In *IEEE GHTC*. 585–589.

[33] V. Pothamsetty and M. Franz. 2004. SCADA HoneyNet Project: Building Honeypots for Industrial Networks. (CIAG) Cisco Systems.