A Real Time Physics-Based Industrial Control System Honeynet Architecture for the Smart Grid

Gabriel De Pace, Hui Lin, Yan (Lindsay) Sun
Department of Electrical, Computer and Biomedical Engineering
University of Rhode Island
Kingston, Rhode Island
{gdepace, huilin, yansun}@uri.edu

***Abstract*—Many details of cyber attacks occurred to Industrial Control Systems (ICS) such as power grids remain as a secret. To reveal those secrets and benefit a broad research community, we propose an original cyber-physical honeypot architecture that can seamlessly couple IT and OT components in an power grid. Specifically, we will provide high interaction with adversaries as physical processes of an ICS move forward, revealing attack strategies according to adversaries' knowledge on physical trajectories of the target system. Our preliminary evaluations demonstrate that the proposed CPS honeypot can present realistic device fingerprints and application-layer payload that piggybacks meaningful measurements conforming to the physical model of a power grid. In addition, by leveraging runtime high-fidelity OPAL-RT simulator, our CPS honeypot causes small latency, at least one order of magnitude smaller compared to the existing work.**

***Index Terms*—industrial control systems, honeynet, smart grid**

I. Introduction

Critical Infrastructures such as water supplies, telecommunications, transportation systems, and power grids heavily rely on Industrial Control Systems (ICS). The increased connectivity opens the door not only for efficiency improvement, but also for malicious actors, such as advanced persistent threats (APT) [1], [2]. *APTs present unique features compared to the attacks happened in general information technology (IT) infrastructures*; adversaries can remain undetected for a extended period until they launch attacks leading to irreversible damage.

Despite of such unique features, APTs keep many of their secrets hidden, since research communities have limited accesses to the details of a large number of real incidents. Using attacks on Ukrainian power plants as an example, we can learn from the public-available report that the attacks are coordinated in three distribution systems in sequence to cause a 8-hour blackout [3]. But why these three distribution systems and why each attack is separated around thirty minutes? Even though we intuitively know the answers are closely related to the operational technology (OT) components of the power grid, many details are restricted to protect companies' privacy. *Without obtaining the in-depth knowledge of ICS attacks, especially how they leverage physical processes, research communities significantly fall behind to design effective security solutions.*

Honeypots are useful tools to provide direct and interactive opportunities with real attacks [4]. By interacting with real attacks, honeypots can capture their activities and learn about their aims and strategies. However, honeypots adaption to ICSs are slow and limited. Current honeypot projects mainly focused on IT infrastructures, such as improving the simulation fidelity of proprietary devices from various vendors and tracking their activities occurred within specific devices. Even though these approaches can help to reveal the visibility of adversaries' early-stage activities with IT environments, they fail to capture and characterize their interactions with OT environments and how they leverage physical processes to drive their malicious activities in a long term.

In this paper, we propose an original cyber-physical honeypot architecture that can seamlessly couple IT and OT components in an ICS. Leveraging this feature that are missing in state-of-the-art ICS honeypots, we will provide high interaction with adversaries as physical processes of an ICS move forward, revealing attack strategies according to adversaries' knowledge on ICS' physical trajectories. The objective of this project is not to replace existing ICS honeypots. Instead, we position our work as a complementary service, enhancing their capabilities with interfaces that can automatically and transparently retrieve physical measurements from a high-fidelity ICS simulation environment and update its physical configurations according to adversaries' interactions at runtime.

Specifically, to implement our CPS honeypot, we first enhance Conpot with HoneyD, enabling high-interaction with adversaries with various network stack implementations profiled from real devices, e.g., Schweitzer Engineering Lab (SEL) Protection Relay 751A. Then, we build a small "update engine" based on socket programming interfaces, connecting Conpot with a high-fidelity hardware simulator OPAL-RT simulating realistic physical processes. The "update engine" asynchronously retrieves measurements from OPAL-RT simulators and triggers a new round of simulation according to the interactions received by Conpot. Our evaluations successfully demonstrate that the proposed CPS honeypot can (i) interact with potential adversaries, (ii) spawn multiple sessions of network communication corresponding to physical devices at different locations, and (iii) introduce negligible overhead in terms of network latency and throughput.

II. Related Work

Current studies categorize honeypots into low-interaction and high-interaction according to their involvement or interactions with actual adversaries. A low interaction honeypot provides a limited number of services based on software emulation. For example, Gaspot emulates a virtualized Guardian AST gas tank, allowing adversaries to modify some basic physical values such as temperature and volume [5]. Conpot is another low-interaction honeypot used in several previous studies, supporting various ICS devices and allowing adding new ondes based on their XML-based interface [6]–[8]. The low-interaction honeypots can detect simple attacks, especially the ones automated in scripts (e.g., scanning for open ports), with low-cost of set up and maintenance. However, advanced attacks can easily detect the fake environments and render the attack tracking impossible.

A high interaction honeypot, on the other hand, involves more comprehensive services. The objective is to provide realistic responses to adversaries' probing continuously, allowing us to track their long-term activities. The most recent example of high-interaction honeypot designed for ICS environments is HoneyPLC [9]. It aims to support a broad range of PLC models, by enhancing HoneyD project with custom device fingerprints and ladder logic logging capabilities. However, current high-interaction honeypot are high-interaction regarding to ICS' IT components, including proprietary network protocols to custom system software. They leaves users to provide meaningful measurements from OT components, which is not a trivial task.

There are several studies emphasizing that cyber-physical honeypots should integrate the simulation of physical processes with IT equivalence [10]–[12]. Litchfield et al. presented HoneyPhy in [10], a proof-of-concept prototype integrating a simple heating system and their customized network stack simulator. Antonioli et al. focused on the network simulation based on the current Ethernet/IP protocols [11]; the network simulation is connected to the simulation of a simple water treatment plan based on their MiniCPS testbed. In stead of a simple and local physical process, Mashima et al. designed an electrical substation honeypot integrating more complicated physical processes in power grids substations [12]. However, their network simulations are based on virtual machines, which can be costly and make the honeypot unscalable.

TABLE I
COMPARE OUR CONTRIBUTION TO RELATED WORK.

	IT Components Interactions	Device Fingerprints	OT Component Interactions
Conpot	low	✗	✗
HoneyPLC	high	✓	✗
HoneyPhy	high	◪	✓(low)
Honeypot in a box	high	✗	✓(low)
Substation Honeypot	high	◪	✓(low)
Our work	high	✓	✓(high)

Our Contribution. In Table I, we position our work against the important related work. We use "✓" and "✗" to directly specify whether an approach is or is not equipped with the corresponding capability. When the work does not explicitly specify a capability , we use "◪" to reflect the fact. Our work will use HoneyD's capability to achieve high-interaction regarding network communications on behalf of real physical devices. Meanwhile, using OPAL-RT simulators allows us to modify configuration and retrieve data from simulation without stopping it, achieving high-interaction regarding to OT components as well.

III. CPS Honeypot Architecture

As shown in Figure 1, we design a CPS honeypot by including four components: a front-end, a Conpot instance, a update engine, and the simulation of a power system in the OPAL-RT simulator. In our current setup, we implement these components in four different machines that can be addressed based on IP addresses in a internal network. However, our designs are not restricted by the detailed network configuration, as long as machines can communicate with each other based on IP-based network links.

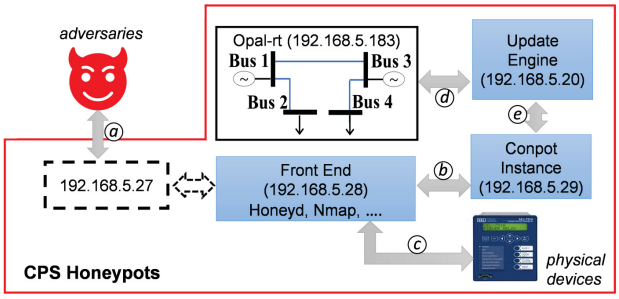

Fig. 1. **Architecture of the CPS Honeypot.**

A. Front End

The front-end directly interacts with external communication, potentially from adversaries, as indicated by ⓐ in the figure. To construct a realistic response, the front end restrofits HoneyD to collect the following information: (i) device fingerprints that are reflected in the communication pattern of lower layer of networks (e.g., TCP/IP protocol), (ii) proprietary application layer protocols used by power grids to exchange information (e.g., Modbus or DNP3), and (iii) meaningful application-layer payload piggybacking measurements that conform to the physical models of a power grid. HoneyD is an open-source software daemon that creates virtual hosts on a network on behalf of a non-existing identities (e.g., shown as dashed box in the Figure). In addition, because HoneyD creates such virtual hosts without dedicating specific computational resources, we can create virtual hosts to represent different physical devices found in a power grid. These features enable our honeypot to achieve high-interaction even though we use services provided by a low-interaction Conpot instance.

Asynchronous from adversaries' interaction, the front end leverages Nmap to periodically scan physical devices in the control network, specified by ⓒ in the figure. Nmap is an

open-source network mapping tools, determining key system characteristics such as opened ports and version of operating systems by analyzing responses of probes sent to those devices [13]. However, the default Nmap does not include fingerprints belonging to physical devices commonly found in ICSs like power grids. Consequently, we use Nmap to probe existing physical devices (e.g., SEL 751A) and build our own profiles. These profiles can be built in advance and loaded at runtime by HoneyD to construct responses to adversaries.

B. Conpot Instance

We mainly leverage Conpot to construct application layer payload according to the proprietary protocol syntax. When the front end receives a request from adversaries, it forwards the request to a Conpot instance. Conpot's capability to interact with adversaries is limited by the fact that it is unable to construct realistic lower-layer network contents, especially if they are based on the TCP protocol. To overcome this drawback, we use the front end to extract application payload of the responses sent back from the Conpot instance (indicated by ⓑ in the figure) and concatenate the payload with the low-layer network contents constructed in HoneyD.

C. Update Engine

Conpot uses random and static measurements encoded in a XML file to construct application-layer payload based on proprietary network protocols used by ICSs. These fake measurements can easily reveal the fake environment. To insert meaningful physical measurements conforming to the physical model of a real power grid, we rely on physical simulation performed in OPAL-RT, a high-fidelity runtime simulation that can analyze power system runtime states on the granularity of sub-microseconds.

For this end, we build a update engine based on a common socket interface, periodically retrieving all measurements from OPAL-RT (specified by ⓓ in the figure). OPAL-RT includes an web server, facilitating external entity to obtain simulation states through various network protocols, e.g., Modbus, DNP3, and a proprietary variant of the UDP protocol. When a Conpot needs some measurements as requested by adversaries, the update engine deliver them accordingly without consulting to the OPAL-RT simulation (specified by ⓔ). Note that the communication specified by ⓓ is performed asynchronously from the communication specified by ⓔ. Consequently, the update engine serves as a "cache" to store runtime simulation states, reducing the latency of constructing responses to adversaries at runtime.

D. OPAL-RT Simulation

In addition to retrieving measurement from OPAL-RT simulation, we leverage the runtime simulation feature of Opal-RT to reflect the changes that adversaries attempt to make. In other words, when a change is made on a physical model, OPAL-RT can instantly apply the change in the simulation without restarting the model. When an adversary initiate a control command (e.g., opening a circuit breaker to destabilize a power grid as in [3]), our HoneyPot can instantly apply the change and respond with the updated measurement reflecting those changes to the adversaries at real time (see details in Section IV-C). These measurement changed based on adversaries action can further convince adversaries that they interact with actual power grids and continue further activities. In addition to logging activities adversaries performed with IT components, we can log the sequence of the control commands as well as the corresponding trajectory of physical state to reflect the attack strategy.

IV. Evaluation

A. Implementation

We implemented four components of the CPS honeypot in off-the-shelf operating systems either running on a virtual machine or a workstation, increasing its portability of deployment. In the preliminary evaluation, we implemented a Modbus server client based on uModbus library as an adversary [14], which sends probing messages to collect system statuses and control messages to open circuit breakers as attack actions. To simulate a physical model of a power grid, we implemented IEEE 9-bus system in the OPAL-RT simulator. Specifically, an encoder of a circuit breaker was attached to Bus 4 of the 9-bus system; it serves as a web server to take remote commands and change the topology of the system.

B. Responses of Nmap Probing

To evaluate the capability of the front end component responding with realistic device fingerprints, we use the same Nmap to probe our honeypot and a SEL 751A device. The snippets of the responses are presented in Table II. By default, Nmap includes few profiles for ICS devices. Consequently, we can see that the Nmap identifies SEL 751A as a type of HP LaserJet printer, which are similar to the responses from the CPS honeypot. Note that some minor differences can be observed from details of the device fingerprint; these are mainly due to the variation of the target system and networks, which do not affect Nmap's decision.

C. Responses from OPAL-RT Simulation

Applying run-time physical changes and the capability of continuous simulation plays a critical role in our CPS honeypot. Without this capability, unexpected latency can occur due to the restart of the simulation. For example, the evaluations in [12] shows that using Power World to handle the physical changes requires approximately 500 milliseconds as Power World simulator needs to be restarted to re-calculate grid states after physical changes. Our honeypot significantly benefit from this feature from OPAL-RT. In Figure 2, we show the responses collected within OPAL-RT, when we open the circuit breaker. From the figure, we clearly see the changes of the system state between time 0.01 to 0.02, which does not involve any simulation rebooting.

TABLE II
NMAPS RESPONSES FROM THE CPS HONEYPOT AND SEL 751A.

Snippet of Nmap Responses from the CPS honeypot
PORT STATE SERVICE REASON 21/tcp open ftp syn-ack ttl 64 23/tcp open telnet syn-ack ttl 64 80/tcp open http syn-ack ttl 64 MAC Address: 00:30:A7:23:5B:53 (Schweitzer Engineering) Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port Device type: printer Running: HP embedded OS CPE: cpe:/h:hp:laserjet_cp4525 cpe:/h:hp:laserjet_m451dn OS details: HP LaserJet M451dn, CM1415fnw, or CP4525, HP LaserJet M476dw printer
Snippet of Nmap Responses from SEL 751A
PORT STATE SERVICE REASON 21/tcp open ftp syn-ack ttl 64 23/tcp open telnet syn-ack ttl 64 80/tcp open http syn-ack ttl 64 20000/tcp open dnp syn-ack ttl 64 MAC Address: 00:30:A7:23:5B:53 (Schweitzer Engineering) Device type: printer Running: HP embedded OS CPE: cpe:/h:hp:laserjet_cp4525 cpe:/h:hp:laserjet_m451dn OS details: HP LaserJet M451dn, CM1415fnw, or CP4525

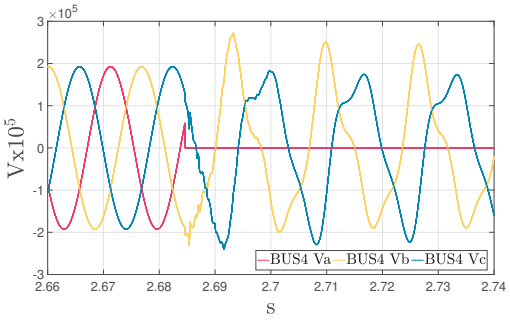

Fig. 2. **Responses from OPAL-RT simulation to physical changes.**

Because OPAL-RT can apply changes without restarting the simulation, adversaries observe minor differences from network communication. As shown in Table III, adversaries observe approximately 41.6 ms of round-trip time when they attempt to apply physical changes, which is at least one order of magnitude smaller compared to the evaluation in [12]. This level of latency is consistent with the latency observed in actual utility environments [15], making it difficult for adversaries to detect the fake environment.

D. Performance

In Table III, we present the performance of the CPS honeypot, quantified by the round-trip time and throughput observed by adversaries. This performance evaluation can estimate the scale of a coordinated attack that the honeypot can handle. Comparing the cases with and without OPAL-RT simulation, we can see that minor changes are observed, on the same of level of variations caused by network conditions. Also, the small latency caused by the CPS honeypot provides a great opportunities to intentionally add latency, manipulating communication patterns from more complicated network scheduling or congestion mechanisms.

TABLE III
PERFORMANCE OF THE CPS HONEYPOT. (AVERAGE MEASUREMENT WITH 95% CONFIDENCE INTERVAL)

Evaluation Cases		Round-trip Time (ms)	Throughput (kbps)
With OPAL-RT	*Retrieve Data*	35.8 ± 0.1	29.5 ± 0.1
	Open/Close Breakers	41.6 ± 0.8	25.3 ± 0.5
Without OPAL-RT	*Retrieve Data*	35.8 ± 0.1	29.5 ± 0.1
	Open/Close Breakers	42.2 ± 0.3	25.0 ± 0.2

V. CONCLUSION AND FUTURE WORK

We present an architecture of CPS honeypot that can perform high interaction with adversaries with realistic responses. The responses integrate device fingerprints collected by Nmap, proprietary application-layer protocols enabled by Conpot, and meaningful application layer payload piggybacking measurements conforming to the physical model of a power grid. In a preliminary evaluation, we demonstrate that an adversary is able to apply physical changes and obtain the responses reflecting those changes. Compared to previous studies, our Honeypot presents much smaller latency, making it difficult for adversaries to identify the fake environment.

By spawning more virtual hosts from the front-end component, we hope to build a honeynet to present adversaries a complete picture of a power grid. After careful evaluation, the honeynet can be deployed on the public Internet to collect real malicious attacks.

VI. ACKNOWLEDGEMENT

This work was supported by the Office of Naval Research under Contract no. N00014-20-C-1096. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the Office of Naval Research.

This material is based upon work partially supported by the National Science Foundation under Award No. CNS-2144513. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] K. E. Hemsley, E. Fisher *et al.*, "History of industrial control system cyber incidents," Idaho National Lab.(INL), Idaho Falls, ID (United States), Tech. Rep., 2018.

[2] S. P. Sithungu and E. M. Ehlers, "Gaainet: A generative adversarial artificial immune network model for intrusion detection in industrial iot systems," *Advances in Information Technology*, vol. 13, pp. 456–461, October 2022.

[3] R. Lee, A. M., and T. Conway, "Analysis of the cyber attack on the ukrainian power grid," nerc.com/pa/CI/ESISAC/Documents/E-ISAC SANS Ukraine DUC 18Mar2016.pdf, March 2016.

[4] L. Spitzner, "Honeypots: Catching the insider threat," in *19th Annual Computer Security Applications Conference, 2003. Proceedings.* IEEE, 2003, pp. 170–179.

[5] K. Wilhoit and S. Hilt, "The gaspot experiment: Unexamined perils in using," *blackhat*, 2015.

[6] S. Maesschalck, V. Giotsas, and N. Race, "World wide ics honeypots: A study into the deployment of conpot honeypots," in *Industrial Control System Security Workshop*, 2021.
[7] P. Ferretti, M. Pogliani, and S. Zanero, "Characterizing background noise in ics traffic through a set of low interaction honeypots," in *Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy*, 2019, pp. 51–61.
[8] A. V. Serbanescu, S. Obermeier, and D.-Y. Yu, "Ics threat analysis using a large-scale honeynet," in *3rd International Symposium for ICS & SCADA Cyber Security Research 2015 (ICS-CSR 2015) 3*, 2015, pp. 20–30.
[9] 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.
[10] S. Litchfield, D. Formby, J. Rogers, S. Meliopoulos, and R. Beyah, "Rethinking the honeypot for cyber-physical systems," *IEEE Internet Computing*, vol. 20, no. 5, pp. 9–17, 2016.
[11] D. Antonioli, A. Agrawal, and N. O. Tippenhauer, "Towards high-interaction virtual ics honeypots-in-a-box," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, 2016, pp. 13–22.
[12] D. Mashima, B. Chen, P. Gunathilaka, and E. L. Tjiong, "Towards a grid-wide, high-fidelity electrical substation honeynet," in *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2017, pp. 89–95.
[13] G. Lyon, "Nmap: the network mapper - free security scanner," https://nmap.org, 2022, accessed: 2022-2-22.
[14] "umodbus - umodbus 1.0.0 documentation," https://umodbus.readthedocs.io/en/latest/, 2022, accessed: 2022-2-22.
[15] D. Formby, P. Srinivasan, A. Leonard, J. Rogers, and R. Beyah, "Who's in control of your control system? device fingerprinting for cyber-physical systems," in *Proceedings 2016 Network and Distributed System Security Symposium*. Internet Society, 2016.