ICSpot: A High-Interaction Honeypot for Industrial Control Systems

Mauro Conti
Department of Mathematics
University of Padova
Padova, Italy
conti@math.unipd.it

Francesco Trolese
Department of Mathematics
University of Padova
Padova, Italy
francesco.trolese.1@studenti.unipd.it

Federico Turrin
Department of Mathematics
University of Padova
Padova, Italy
turrin@math.unipd.it

2022 International Symposium on Networks, Computers and Communications (ISNCC) | 978-1-6654-8544-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISNCC55209.2022.9851732

***Abstract*—Honeypots represent one of the most common solutions to study the adversaries' movements and develop ad-hoc protection strategies. An effective honeypot can mimic a real system behavior and can be used to deceive the attacker and collect data related to his actions. However, current honeypots for Industrial Control Systems (ICSs) still lack realistic physical process simulation of the industrial network. Simulating an industrial process accurately while also enabling interaction with it is a complicated task.**

In this paper, we present ICSpot, the first ICS honeypot that addresses the current state-of-the-art limitations by integrating a physical process interaction. We developed our honeypot by leveraging different ad-hoc ICS tools resulting in a more completed and realistic solution. Then, we installed our honeypot on a local Internet Exchange Point and an AWS server, and we collected the interaction for 30 days. Finally, we report the finding related to the interaction collection and compare the results on the two installation points. Our results show that the physical process port we implemented is highly attractive to attackers.

***Index Terms*—CPS, ICS, Measurement, Honeypot**

I. Introduction

Most of the Industrial Control Systems (ICSs) were designed decades ago to operate in an air-gapped environment, as recommended by the Purdue Mode [1]. Therefore, security practices, such as Encryption and Authentication, were not considered. However, due to the increasing digitalization and "smartification" of the processes, ICSs have been integrated with Internet connections to allow remote control or remote diagnosis operations. Different works measuring the current exposition and implemented protocols showed dramatic security lack. In [2] the authors identify that 75.6% of the industrial protocols implemented to communicate over the Internet do not implement any security feature. Among the various security solutions proposed in the literature, honeypots represent an important protection mechanism that still needs contribution in the ICS field [3]. The goal of a honeypot is twofold: it can be used to fool the attacker, making them think they are interacting with the real system and collecting data about typical adversarial attack actions. However, developing an ICS honeypot is challenging since it should simulate both the network and the physical process with high fidelity. Honeypots are classified based on the level of interaction they offer to an attacker. *High-Interaction Honeypots* simulate all the services of the emulated machine and allow a high level of interaction with the emulated system. Instead, *Low-Interaction Honeypots* emulates the operating system and services provided by the simulated device, but due to the lower engagement it provides, it makes it possible to capture less information.

In this paper, we present ICSpot, the first *High-Interaction* ICS honeypot that addresses the limitations of the physical process interaction of other ICS honeypots. Since ICSs are characterized by physical processes, the lack of such features can lead to an incomplete emulation of an industrial system. Once we developed the honeypot, we hosted it in an AWS server and a local Internet Exchange Point (IXP). We then analyze the results after one month of exposition, highlighting the differences between the two installation points. The source code of the honeypot is available on Github[1]. We summarize our contribution as follows:

- We present ICSpot, the first honeypot implementing an interactive physical process.
- We installed the honeypot in two different hosts and collected data for one month. We then provide an analysis of the received interaction. This analysis confirms that the physical process port is the most interacted one.

II. Related Works Limitations

The main challenge in designing an ICS honeypot is integrating a reliable physical process simulation to capture an attacker's attention. The majority of the proposed works lack the implementation of such a feature. ICSpot is built on top of the recent open-source HoneyPLC [4], which to the best of our knowledge, represents the most advanced ICS honeypot available. However, HoneyPLC does not include a physical process simulation. Starting from HoneyPLC, we extended its framework to include additional services and interaction with a MiniCPS-based [5] physical process together with an interactive Human Machine Interface (HMI) representing the physical process evolution.

ICS Honeypot Comparison. Table I reports a comparison between the different ICS honeypots. This table is an extension

978-1-6654-8544-9/22/$31.00 ©2022 IEEE

[1] "ICSPot" on Github, github.com/ftrole/ICSpot

Honepot/Features	Open Source	Network Simulation	Physical Interaction	Log	HMI
CryPLH [6]	○	◐	○	○	○
SHaPe [7]	●	○	○	●	○
Gaspot [8]	●	○	◐	●	○
GridPot [9]	●	○	◐	●	○
Conpot [10]	●	○	○	●	○
Antonioli et al. [11]	◐	◐	◐	●	○
HoneyPhy [12]	○	○	◐	○	○
DiPot [13]	○	○	○	●	○
S7commTrace [14]	○	○	○	●	○
Mimepot [15]	●	◐	●	○	○
HoneyNet [16]	●	●	○	●	○
HoneyPLC [4]	●	●	○	●	○
ICSpot	●	●	●	●	●

TABLE I: Comparison among honeypots in literature. ● the feature is included. ◐ the feature is implemented but has limitations. ○ the feature is not implemented or not documented.

of the comparison presented in [4]. To the best of our knowledge, ICSpot is the first open-source honeypot that exposes an effective and interactive physical process. We also included an HMI representing the evolution of the process so that the attacker can interact with the physical process and look at the physics modification in real-time.

III. ICSpot Honeypot

We developed ICSpot by leveraging different existing ICS tools to obtain a complete ICS honeypot and address the current related works limitations. In particular, we built the ICSpot on the top of HoneyPLC [4]. HoneyPLC is a recent and effective honeypot which, as explained by the authors, implemented a complete set of functionalities, including multiple Programmable Logic Controllers (PLCs) simulation, ladder logic capture capabilities, and low Honeyscore. We extended HoneyPLC to include an interactive physical process, additional industrial exposed services, and a log analysis interface.

A. Architecture

Figure 1 represents the components of ICSpot and how they interact with each other. The core tool we used to simulate the TCP/IP stack is Honeyd. Honeyd listens for requests addressed to the honeypots, simulates, and responds by imitating a list of services. Packets sent by Honeyd are modified by its personality engine: a component that allows simulating the behavior of a predefined operating system or a device chosen from the list of Nmap fingerprints. A configuration file lists the parameters defining the machine to be simulated, such as its MAC address, IP address, operating system, and services. Thanks to these features, Honeyd allows configuring and simulating complex honeypots without exposing the host machine to risks. ICSpot implements different industrial services described in Section III-C. The most innovative is the S7comm server, and the integration of the physical process (Section III-B) has made it possible to create a highly interactive honeypot that aims to overcome the limitations of the work described in the previous chapter. Among the different PLCs offered by HoneyPLC, we decided to emulate Siemens Simatic S7-300. This choice is motivated by the lower Honeyscore

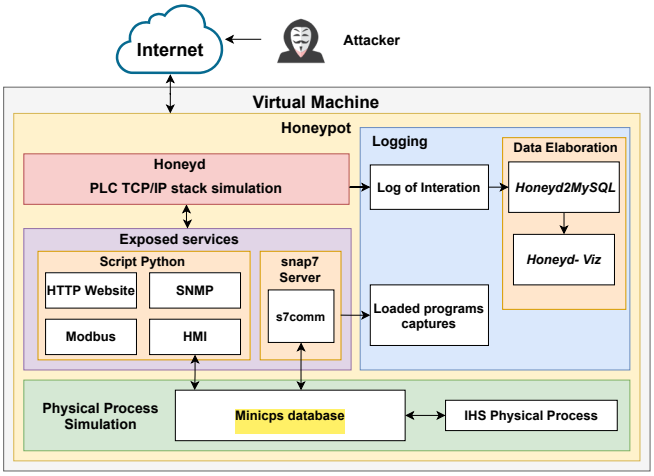


Fig. 1: ICSpot Schema

obtained in [4] allowing, therefore, the best level of simulation and likelihood. The Shodan Honeyscore is a mechanism that flags a device as a honeypot with a confidentiality score. The score is calculated by analyzing features such as open network ports or default known honeypot settings. All the interactions that occurred with ICSpot are recorded, stored, and processed in a Logging module (Section III-D).

B. Physical Process

from HoneyPLC paper:
HoneyPLC does not provide support for modeling physical interactions as depicted by PLCs in practice.

The main contribution of ICSpot compared to its predecessors is integrating physical process simulation and the possibility of interacting with it. This is achieved by running a MiniCPS [5] simulation in the host machine. To this end, we leveraged the simulation proposed in the IHS project[2] by running it on the host machine of the honeypot in the background. IHS project is based on a simplified simulation of the SWaT water treatment process, provided by MiniCPS [5]. Two PLCs act on two pumps, which regulate the incoming and outgoing water flow in a tank, turning them on or off to maintain a constant water level. We integrated with ICSpot the possibility of using the S7comm protocol to write or read data in the MiniCPS database. In particular, we included the option of reading the PLC data blocks that include the current values on the MiniCPS database. Furthermore, the attacker can modify the state of the valves controlling the water flow. This allows interacting with the real-time undergoing water process. To avoid the crash of the process after disruptive interactions, we integrated a process adjustment when the water reaches too low or too high levels. The IHS project also offers a web interface showing data coming from the simulation. We modified such an interface to obtain a specific industrial HMI aspect and exposed it to deceive the attackers.

C. Services and Interaction

ICSpot includes a PLC website and a web interface reporting data about the simulated physical process, the interaction with the PLC memory exploiting the industrial protocols

[2]Industrial Hacking Simulator (IHS), github.com/CarlosLannister/IHS

ask professor to have a look for this with u.
the difference between this physical process, and THI physical process.

I think the main difference is that here they use IHS which is a full simulation system, instead of just reading from simple files

S7comm and Modbus, and device monitoring via SNMP protocol. In the following, we reported the details of the different services exposed.

HTTP. Generally, PLCs integrate HTTP web services to provide a list of functionalities. ICSpot implements a copy of the Siemens Simatic S7-300 website on port 80. On port 8000, ICSpot exposes an HMI interface that shows details of the physical process simulated: the tank water level, the status of the two valves, and the water ingoing and outgoing volume.

SNMP. The SNMP protocol, used to monitor devices connected to the network, is listening on port 161 UDP. The SNMP agent, when queried, responds with the Management Information Base (MIB), which reports information related to the PLC. This protocol is implemented with a python script from the SCADA Honeynet project [16] and contains the MIB data of a real Simatic S7-300 PLC.

Modbus. Modbus/TCP protocol enables the communication between industrial devices and supervisor computers in a master-slave paradigm. ICSpot implements Modbus/TCP using a python script from the SCADA Honeynet project [16]. This implementation is considered low-interaction but allows connecting to the honeypot, which takes the role of slave, and reading and writing data, exactly as with a real PLC.

S7comm. The S7comm protocol was challenging to implement, mainly because Siemens does not release the official documentation. Following the example of HoneyPLC, we employed the snap7 library to implement an S7comm server listening on port 102 TCP. Based on HoneyPLC implementation, the S7comm server enables different functions. First, it simulates a Siemens Simatic S7-300 PLC, offering the possibility to read and write PLC memory blocks. Second, the S7comm server allows the honeypot to fool the Siemens Step7 Manager proprietary software. Since this proprietary software cannot detect the honeypot, it is reasonable to assume that no other application which can connect to it through the S7comm protocol can do it. Third, the S7comm server allows storing the programs injected by attackers in the virtual machine file system on which the honeypot is installed. This allows an accurate analysis of potential malware. These three functionalities derive from the HoneyPLC server implementation. Finally, we innovate the implementation of the S7comm server by offering the attacker data related to the physical process. The attacker can read the value of the volume of water present in the simulated tank by accessing the data blocks. The attacker can also modify the state of the valves in the process. This interaction allows the attacker to communicate with the physical process through the S7comm protocol. The S7comm server runs locally and is connected to the honeypot via the Honeyd configuration file.

D. Log Data Visualization

In order to make more usable and organized the data on the log generated by Honeyd, we leveraged Honeyd2MySQL[3] and Honeyd-viz[4] open-source tools. Honeyd2MySQL allows to extract all the information from the Honeyd log and import them into a MySQL database. Then HoneydViz exploits the database created with Honeyd2MySQL to create a web interface showing useful statistics related to the collected data, such as the number of connections divided by ports, the IP addresses of the main attackers, their origin, and the connections established per day.

IV. Results

We installed two instances of ICSpot in two different environments. The first honeypot was installed in an AWS EC2 virtual machine, while the second honeypot was installed in a virtual machine in a network segment owned by VSIX[5] IXP. *VSIX* is a local IXP that manages the traffic circulating in the North East of Italy. VSIX allows us to integrate many interesting features to the honeypot exposed, thanks to their infrastructure. In particular, the IP dedicated to the honeypot has been announced worldwide thanks to the Transit Provider, allowing, therefore, high visibility worldwide.

We installed and exposed the virtual machines for one month. We report the data collected in the following.

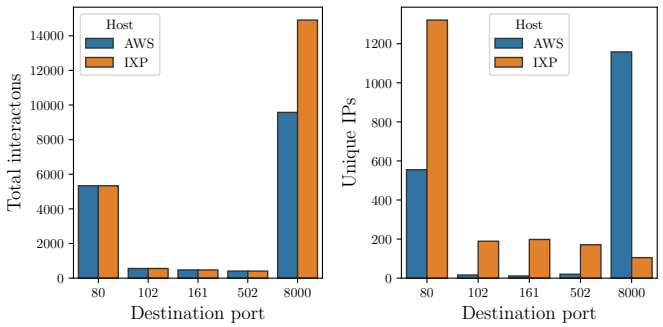
A. Interaction Analysis

Figure 2a compares the distribution of the interactions with the different exposed services for the two honeypot instances. We can see that the most requested were the website (port 80) and the HMI (port 8000). In particular, as reported in Figure 2b, we registered 2185 different IPs interacting with the port 8000 in the AWS instance and 105 in the VSIX case. Instead, we registered for port 80, 555 unique IPSs in AWS, and 1321 in VSIX. While such a high number of interactions was expected for port 80 since it is one of the most famous service ports, the high requests in port 8000 confirm the effectiveness of the HMI in attracting and increasing the attacker's engagement. Another interesting insight concerns the interactions through s7comm, the second most exploited protocol by attackers to interact with ICSpot. These data indicate the effectiveness of ICSpot in implementing a service that faithfully reproduces that of an original PLC. We also analyzed the origin of the IPs used for interaction. We note that the scanning sources were identically distributed for both honeypot instances. This indicates that most malicious hosts that scan the Internet for vulnerable machines do not discriminate against associations or geographic areas but perform systemic network crawling. The nations from which the highest number of scans have been recorded are the United States and China, followed by Russia and India. Note that the original IP source represented may be hidden using VPN. In this case, the IP identified can represent the last VPN hop.

B. Interactions Origin

To analyze the presence of malicious actors among the IP sources, we leveraged GreyNoise[6]. Greynoise is a company that collects, labels, and diagnoses data and provides access

[3]honeyd2mysql, https://github.com/ikoniaris/honeyd2mysql
[4]Honeyd-viz, https://github.com/ikoniaris/honeyd-viz
[5]VSIX Internet Exchange Point, https://www.vsix.it/
[6]Greynoise, https://greynoise.io/



(a) Services targeted (b) Number of Unique IPs

Fig. 2: Services analyzed in the two ICSpot instances and number of different IPs connecting to each service.

to such information to users via API. Results between the two honeypot instances were slightly different. In both cases, 97% of the analyzed IP addresses are labeled in the GreyNoise database "scanner". However, VSIX honeypot received a higher amount of different malicious scanning according to Greynoise classification (i.e., about 49%). Instead, 22% of the scanning was labeled as malicious in the AWS honeypot according to the Greynoise classification. This difference can be due to Shodan, which labels the AWS instance as a honeypot, reducing the interest of malicious scanners in analyzing such IP. The top-3 organization with the highest number of malicious IP addresses in the VSIX case are Google LLC, Korea Telecom, and Chunghwa Telecom Co. Instead, in the AWS instance, the top-3 include DigitalOcean LLC, Google LLC, and WIND Telecom S.A. Both the lists are followed by numerous ISPs from all over the world. Finally, GreyNoise can identify the actor related to an IP address, i.e., the entity actually using the IP address. The actor can differ from the organization to which the address belongs since it commonly happens that ISPs and companies offering cloud computing services or IP addresses block which are used for malicious activities. All the actors analyzed belong to hosts categorized as "benign" and represent legitimate organizations scanning the network to identify exposed and vulnerable services, sometimes also notifying the owners about the dangers they incur. The most frequent benign scanners in both the honeypot include Stretchoid.com, Censys, Bitsight, ShadowServer.org, BinaryEdge.io, and Shodan.io. All the malicious scanner actors are instead unknown by Greynoise.

V. Conclusion

In this work, we presented ICSpot, an ICS honeypot that addresses the related works' current limitations by employing an interactive physical process. We compared the feature of our honeypot with the related works proving its effectiveness and contribution. The comparison highlight the contribution of ICSpot in terms of physical process simulation. We then exposed ICSpot on two different hosts, and we analyzed the interactions after a month of data collection. By leveraging Greynoise, we identified in the VSIX instance that 49% of the IPs interacting with the honeypots were labeled malicious, while in the AWS instance, 22%. The service installed on port 8000 has particularly attracted the attention of attackers labeled as malicious in the AWS instance. This demonstrates the interest of attackers in interacting with a physical process in an industrial system. We believe that ICSpot can represent a building block toward implementing future honeypots with a higher degree of fidelity. In future works, we will extend the functions of ICSPot, like the PLC profiles, and compare the interactions received by ICSPot with other existing solutions.

Acknowledgment

Federico Turrin is supported by a grant from the Cariparo Foundation and Yarix S.r.l. which we would like to thank. We want to thank *VSIX* for enabling us to install the honeypot and collect data at their IXP.

References

[1] T. J. Williams, "The purdue enterprise reference architecture," *Computers in industry*, vol. 24, no. 2-3, pp. 141–158, 1994.

[2] G. Barbieri, M. Conti, N. O. Tippenhauer, and F. Turrin, "Assessing the use of insecure ics protocols via ixp network traffic analysis," in *2021 International Conference on Computer Communications and Networks (ICCCN)*, 2021, pp. 1–9.

[3] J. Franco, A. Aris, B. Canberk, and 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*, vol. 23, no. 4, pp. 2351–2383, 2021.

[4] 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*, pp. 279–291.

[5] D. Antonioli and N. O. Tippenhauer, "Minicps: A toolkit for security research on cps networks," in *Proceedings of the First ACM workshop on cyber-physical systems-security and/or privacy*, 2015, pp. 91–100.

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

[7] K. Kołtyś and R. Gajewski, "Shape: A honeypot for electric power substation," *Journal of Telecommunications and Information Technology*, no. 4, pp. 37–43, 2015.

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

[9] W. O. Redwood, "Cyber physical system vulnerability research," Ph.D. dissertation, The Florida State University, 2016.

[10] A. Jicha, M. Patton, and H. Chen, "Scada honeypots: An in-depth analysis of conpot," in *2016 IEEE conference on intelligence and security informatics (ISI)*. IEEE, 2016, pp. 196–198.

[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] 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.

[13] J. Cao, W. Li, J. Li, and B. Li, "Dipot: A distributed industrial honeypot system," in *International Conference on Smart Computing and Communication*. Springer, 2017, pp. 300–309.

[14] F. Xiao, E. Chen, and Q. Xu, "S7commtrace: A high interactive honeypot for industrial control system based on s7 protocol," in *International Conference on Information and Communications Security*. Springer, 2017, pp. 412–423.

[15] G. Bernieri, M. Conti, and F. Pascucci, "Mimepot: a model-based honeypot for industrial control networks," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 433–438.

[16] V. Pothamsetty and M. Franz. (2005) SCADA HoneyNet Project: Building Honeypots for Industrial Networks. Accessed: 01-07-2021. [Online]. Available: http://scadahoneynet.sourceforge.net/