Improved ICS Honeypot Techniques

United Kingdom

Primary author: David McColm

Organization: Dounreay

Email: david.mccolm@dounreay.com

Co-authors: Richard Macfarlane

Organizations: Edinburgh Napier University

Emails r.macfarlane@napier.ac.uk

Abstract

As work continues to advance the security posture of ICS systems across the UKNDA estate, opportunities arise to consider the deployment of deception technologies. With high-profile attacks on ICS occurring more frequently, and increasing numbers of adversaries developing ever more sophisticated techniques, strategies to try and stay ahead of the curve become increasingly necessary. Honeypots are an important research tool for discovering both new threat actors and any new techniques they are developing before they can cause harm. Outside of research, Honeypots are deployed internally as a tool to be used during defensively where they act as a distraction or early warning. This paper will examine current state of ICS Honeypots, and propose a new high-interaction honeypot technique using common industry tools. It is this new honeypot is made cheap and simple to deploy by making use of Siemens PLCSIM software, already in wide use in the nuclear industry. Offline validation testing and live internet deployment will be used to test and compare directly with other existing low and high interactivity honeypots. The results from the honeypots will be compared to examine scanning activity, reconnaissance activity and attacks to look for differences in both type and amount of activity seen.

1. Introduction

The historic lack of attention paid to security during the design and implementation of ICS systems, coupled with the increasing connectivity of previously isolated systems, equates to a burgeoning problem of systems being left vulnerable to remote attacks [1]. With increasing OT/IT connectivity forming a key part of the Nuclear Decommissioning Authority's Digital Transformation Strategy (a posture likely to be reflected throughout the UK nuclear estate), securing these items of Critical National Infrastructure (CNI) properly is paramount. While systems being open to remote availability leaves them more and more vulnerable, attackers continue to develop new capabilities both for ransomware and for state based APT usage [2], [3]. For now, attacks are highly manual, with automation only used for initial reconnaissance scanning, but development appears to be moving towards increasing automation/simplification of attacks similar to the effect of Metasploit on IT attacks [4].

Honeypots are deliberately vulnerable system exposed to attackers either for research or as an internal defensive deployment. Research honeypots are deployed to the internet to attract and record attacker activity to generate threat intelligence on current or new attacks/techniques. Defensive deployments of honeypots are performed on organisations internal networks and made to look like attractive targets for an attacker but with no actual function. This allows early detection of a breach as any traffic directed to them can be assumed to come from an attacker. They also function to distract an attacker, giving defenders more time to react with more advanced, realistic and interactive honeypots providing a longer distraction. ICS honeypots are designed to mimic common ICS systems typically Programmable Logic Controllers (PLC) or Human Machine Interface (HMI) devices which are both used to control industrial processes. ICS Honeypots can be used in both roles; deployed internally they can distract and delay an attacker while providing early warning to defenders, and external deployments can provide useful insight into the current threat environment and trends in attacker behavior and interest. To gain the most meaningful conclusions, not only requires attracting attackers, but then deceiving them throughout the full range of activities in the cyber kill-chain, allowing details of the attempted attacks to be analyzed and turned into useful threat intelligence. This creates the challenge of producing a high-fidelity simulation of the target devices, a non-trivial problem due to the unusual functions and proprietary protocols used by these devices.

This paper proposes a simple-to-deploy honeypot using a commercial simulation tool (Siemens PLCSIM), likely already owned by many Nuclear organisations due to it’s bundling with Siemens TIA programming software, providing a high fidelity simulation of a Siemens PLC capable of providing code upload and download, and of running uploaded/modified code. Details of penetration testing activities and data gathered from a internet facing test run will be discussed.

2. Related Work

There are a number of papers proposing and demonstrating low, medium and high interactivity ICS honeypots, primarily for research purposes using internet facing deployments. Low-interactivity ICS honeypots only implement the most basic surface-level appearance of an ICS system, medium-interactivity honeypots implement some of the actual functionality, and high-interactivity honeypots aim to implement the full range of interactions, sometimes including a simulation of the underlying industrial process.

Conpot [5] is probably the best known ICS honeypot and is commonly deployed for research. It is a low-interactivity honeypot which will only respond correctly to basic scanning commands, but can be customised and extended to have other functionality added. The default settings contain a number of errors and specific strings which make it very easy to detect, and Shodan Honeyscore (an online tool for assessing the likely-hood a device is a honeypot) can automatically analyse and detect default Conpot deployments [6]. Most related work using Conpot, either as a low-interaction honeypot, or with modifications to increase interactivity, have not seen valuable results [7]. Typically, only attracting automated scanning traffic [8] with rare attempts to read and write variables [9].

HoneyPLC, which utilizes ’Honeyd’ (a framework for creating honeypots) to handle initial interaction and comms, and the Snap7 library to handle S7Comm interaction (Siemens’ proprietary PLC communications protocol) appears to be the most advanced ICS honeypot freely available [10]. The Snap7 library allows HoneyPLC to emulate many of the functions of Siemens S7-300, S7-1200 and S71500 PLCs, including code upload and download, but does not simulate or run the code deployed to it making it a medium-interactivity honeypot. To verify its functionality and test how well it emulated a real PLC, the creators performed testing using NMap, Shodan Honeyscore, Siemens Step 7 Manager, PLCInject [11] and PLCScan [12]. The results compared favorably against those given by a real Siemens PLC. Notably, this honeypot is good enough to fool Siemens own PLC management software (Step 7 Manager and Totally Integrated Automation) enough to allow upload/download of Ladder Logic code. During a five month cloud-based internet-facing deployment it received a large volume of scanning traffic, a mixture of protocol aware scans and scans aimed at discovering ordinary services like HTTP on unusual ports. However, they did receive a small amount of very interesting traffic in the form of 4 CPU Stop commands. These commands are used to halt execution of the user supplied program on the PLC, and would be disruptive and potentially dangerous or damaging to the physical process and equipment if performed on a real system. These are highly likely to have been a manual, targeted attack as this was not something seen by other honeypot papers. It is highly likely that HoneyPLC fooled at least one attacker into making a manual interaction, either for reconnaissance, to probe if it was a real device, or a deliberate attack.

3. Design/Implementation

It was decided to implement the core PLC functions identified lacking in available honeypots. Siemens PLC-SIM, which is included with their TIA Portal management software, provides a simulation of almost all the functions of a real PLC device and became the basis for the development of a high-interactivity honeypot.

As PLC-SIM does not provide network accessible services by itself, an additional piece of software was required to remedy this. NetToPLCSIM [13] is a freely available tool which provides this function, making a PLC-SIM instance available to a local network. Tests conducted on a local network using Virtual Machines were successful so a VPN tunnel was used to expose an instance to the internet by forwarding packets from a cloud based Virtual Private Server(VPS).

Testing this using a second VPS to simulate an external actor were also successful. Wireshark was used to capture all network traffic with captures from local tests used for comparison to prove all traffic was being captured. This combination of PLC-SIM, NetToPLCSIM and Wireshark forms a full honeypot design capable of receiving requests and giving appropriate responses while capturing all traffic for analysis.

An internet-facing instance was used to test the concept, alongside other existing honeypots with different levels of interactivity/realism to provide a comparison. This has the added advantage of helping filtering out indiscriminate automated scanning versus attacks that are manually targeted at specific systems as the low level honeypots are less likely to be targeted for more advanced attacks/reconnaissance. Some Conpot instances and a HoneyPLC instance were used along side the new honeypot developed for this work.

The honeypots were deployed on Virtual Machines based on a local hypervisor server. Outside connectivity and exposed IP addresses were provided using minimal-specification Virtual Private Servers (VPS) in the cloud with packet forwarding through VPN tunnels providing the path between them. Siemens TIA V15.1 and PLCSIM V5.4 were installed on the host and NetToPLCSIM was downloaded from SourceForge [13] and configured as per instructions available online [14]. A project was setup in TIA and configured with an S7-300 CPU 315-2 PN/DP (catalog number 6ES7 315-2EH14-0AB0) PLC. PLCSIM was started up from TIA with this configuration, and linked to NetToPLCSIM making the PLCSIM instance accessible on the VPN tunnel, allowing it to respond to forwarded traffic from the cloud server. Figure 1 shows a simplified network diagram.

A basic program was developed in TIA and preinstalled, for an attacker to download, along with a basic physics simulation to provide values which change dynamically and respond to manipulation by an attacker. The program was designed to simulate a basic cooling cycle where a measured temperature value gradually increases until it hits a threshold, which causes the PLC to enable a cooling pump, causing the temperature to decrease while it is running. This will continue until a low temperature threshold is reached at which point the cooling pump will stop and the cycle will restart. State changes on the simulated plant were provided using internal memory Data Blocks and separate ladder logic code using timers to periodically increment/decrement the temperature I/O value depending on the state of the simulated cooling pump.

Figure 2 shows a condensed version of the code, with the first set of rungs taken from the ”Main” program loop and the remaining rungs taken from a separate ”ReadInputs” Function Block (FB). It was intended that the ”Main” Organisational Block (OB) would look realistic with the simulation code separated into a separate Block. This is a simple method which removes the need for additional software or hardware however as it is implemented in the same user code area on the simulated PLC it will be provided to an attacker who reads the code from the PLC (code Upload). This means an attacker with an understanding of ladder logic could be expected to realise the PLC was a honeypot if they view the code beyond the Main OB.

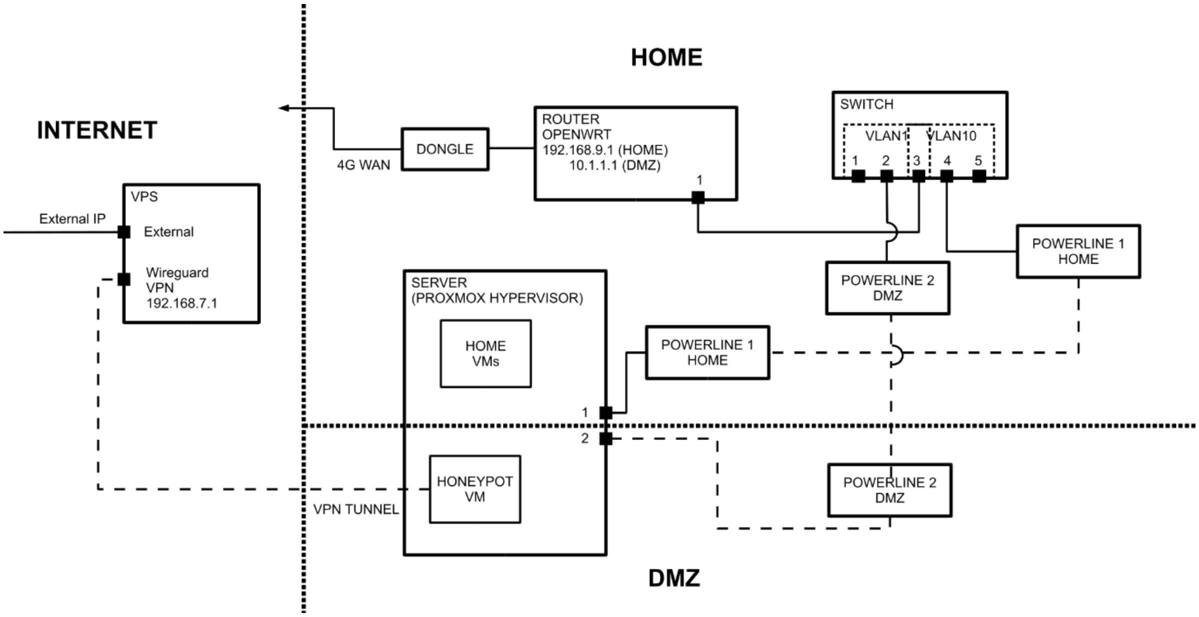


Figure 1: Network Diagram

During initial checking it was noticed that there was one flaw in the CPU scan data provided by PLCSIM, it did not include a serial number. Two attempts were made to correct this using on the fly packet manipulation but neither were successful and there was not enough time to make further attempts.

4. Simulated Attacks

Some simulated attacks carried out, attempting to simulate an attacker exploring the honeypot. During this process, both the response received by the testing programs and packet captures of the network traffic were examined and compared with the same activities performed against a real PLC. Basic scans were performed with NMap and PLCScan showing that all the honeypots were working properly and that the responses contained the expected banner info, with all but a default Conpot instance producing convincing (but not perfect) banners. Deeper probes, using the Python-Snap7 library to perform S7Comm protocol interrogation, quickly found the limitations of the Conpot instances, with incorrect responses given on the initial tests querying values for digital and analog variables. This gave mixed results for HoneyPLC; with optional Datablock variables giving a valid response (although the values are always 0) but memory variables (which should always be present) giving an ”Item Not Available” response, normally seen when an invalid Datablock is requested. PLCSIM gave responses consistent with a real PLC; Datablocks implemented

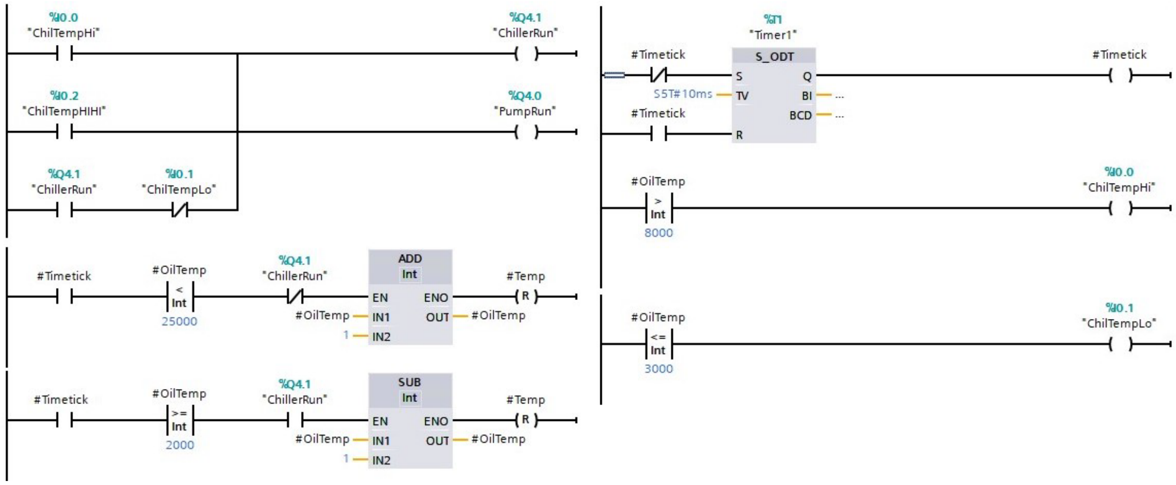


Figure 2: Physics Simulation Code

in the user code and all memory variables can be queried and will give varied responses consistent with the state of the physics simulation and unimplemented Datablocks produce ”Item Not Available”. Both HoneyPLC and PLCSIM gave realistic and consistent responses to requests to list the program blocks and both give correct responses to requests to upload (copy the code from the PLC to the programmer) and download (write code from the programmer to the PLC) code, however, the Datablocks received from HoneyPLC do not contain any data and have a last changed date stamp reading 1984. Blocks received from PLCSIM contain valid data, and code downloaded to the PLC will run on the Honeypots simulation. It should be noted that, as the physics simulation in PLCSIM runs in the user code, it will be obvious as a simulation to an attacker who uploads code from the Honeypot and analyses it. Both HoneyPLC and PLCSIM gave correct responses to requests from the script to change the operating mode of the CPUs from Run to Stop and back again.

Probes were also performed using Siemens TIA programming software, with HoneyPLC and PLCSIM managing to fool this software into connecting to them. The same range of probes as with the Snap7 library were performed using this software; finding the same results for both honeypots with variable reads and code upload/download succeeding. One notable difference was that although HoneyPLC appeared to accept CPU Start/Stop requests, the Running state that should have been changed to ’Stop’ was not recognised by TIA. These tests show PLCSIM to be almost indistinguishable from a real PLC with the only exceptions being the lack of a serial number presented in the banner info and the presence of the physics simulation code in the ladder logic program if it is examined.

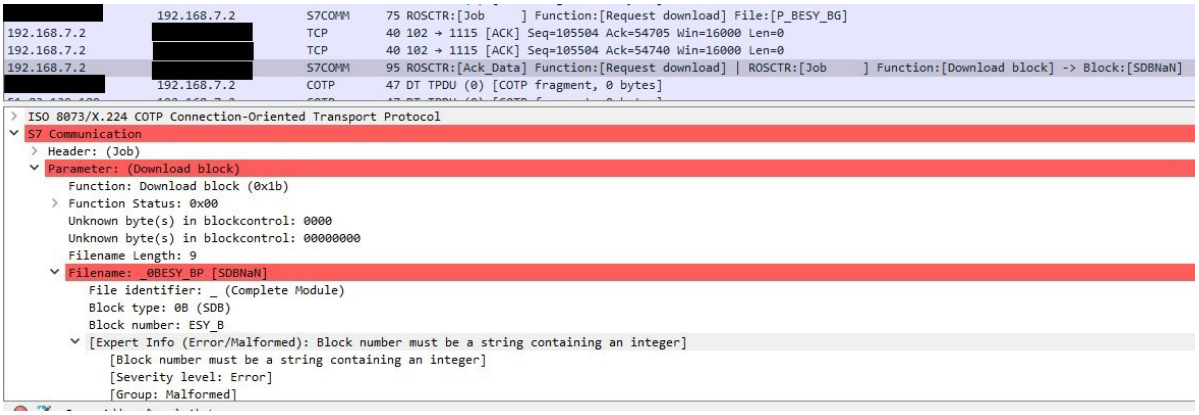
5. Results

Discounting spikes caused by some outliers, all four honeypots received similar levels of scanning traffic doing basic banner grabbing and a very small number of single variable read attempts each. This was the limit of the results for the low interactivity honeypots. As expected, the limited interactivity did not attract more advanced interactions. Both HoneyPLC and PLCSIM received a mixture of more advanced interactions, primarily appearing to be deeper reconnaissance activity but also some actions on the HoneyPLC instance that, even if they were only intended to probe the device, would be highly disruptive to operations making them best classified as attacks.

Both received a number of requests to list available program blocks from a handful of different IP addresses. There appear to be distinct patterns associated with these connections on each honeypot, indicating either a different actor or a different technique being used. HoneyPLC received connections which, when compared to the reference PCAPs taken using TIA Portal and a real PLC, appeared to be generated by the actor using TIA Portal to connect to the honeypot. These connections appear on PCAPS as a large quantity of CPU data requests mixed in with ”List Blocks” commands which are used to present basic information on the user program running on the PLC. PLCSIM received a different pattern of connections which also appeared to be from actor(s) using TIA Portal; these connections were code Upload connections i.e. copying the code on the PLC/Honeypot to the actors computer. These look likely to have been from a single actor as they are all concentrated on the 6th and 7th of December 2022 and many come from sequential IP addresses many of which are known TOR exit nodes. It’s assumed that this is a single actor using some form of VPN which cycles through IP addresses. This activity is significant is it is a crucial reconnaissance step for an attacker to become familiar with the process being controlled by the PLC, allowing them to intelligently manipulate the process. This type of connection has not been reported in previous work indicating a major increase in the threat environment.

Of the traffic connecting to Honey PLC, 83% came from a single IP address which attempted a mixture of variable reads (memory and DB reads) and PLC Start/Stop commands (825600 packets total) during a single long session (around 12 hours). These were likely probes to see whether variables would change when the PLC is in Run/Stop mode, (for a real device variables would stop changing in Stop mode), however, on HoneyPLC, variables do not change anyway. An intelligent actor should notice this along with the incorrect responses to memory variable reads. This is where the inclusion of a physics/process simulation is essential along with correct responses to all protocol queries, a PLCSIM based honeypot would maintain believability during this activity. Given the nature of the traffic it looks like a looped script left to run for a long period. This sort of activity would have been extremely disruptive to a PLC controlling a physical process and has the potential to cause real damage or harm.

HoneyPLC also saw a session with a handful of malformed download requests (attempts to load code onto the PLC) using a filename of ”P BESY BG” and targeting an invalid block number on the PLC (see Figure 3). This appears to have been an attempt by the actor to develop a script to alter code on a PLC, another previously unseen result which also shows the growth in the threat environment and the necessity of having honeypot systems which can correctly respond to these interactions to ensure the full capability of an attacker is seen.



```
[redacted]        192.168.7.2     S7COMM    75 ROSCTR:[Job     ] Function:[Request download] File:[P_BESY_BG]
192.168.7.2       [redacted]      TCP       40 102 → 1115 [ACK] Seq=105504 Ack=54705 Win=16000 Len=0
192.168.7.2       [redacted]      TCP       40 102 → 1115 [ACK] Seq=105504 Ack=54740 Win=16000 Len=0
192.168.7.2       [redacted]      S7COMM    95 ROSCTR:[Ack_Data] Function:[Request download] | ROSCTR:[Job     ] Function:[Download block] -> Block:[SDBNaN]
[redacted]        192.168.7.2     COTP      47 DT TPDU (0) [COTP fragment, 0 bytes]

> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
v S7 Communication
  > Header: (Job)
  v Parameter: (Download block)
      Function: Download block (0x1b)
    > Function Status: 0x00
      Unknown byte(s) in blockcontrol: 0000
      Unknown byte(s) in blockcontrol: 00000000
      Filename Length: 9
    v Filename: _0BESY_BP [SDBNaN]
        File identifier: _ (Complete Module)
        Block type: 0B (SDB)
        Block number: ESY_B
      v [Expert Info (Error/Malformed): Block number must be a string containing an integer]
          [Block number must be a string containing an integer]
          [Severity level: Error]
          [Group: Malformed]
```

Figure 3: Malformed Download Request

Conclusion

The use of PLCSim is a useful new technique for honeypot deployment for the nuclear industry. These tests show it to be convincing, and more interactive, bringing attackers further down the cyber kill-chain than other available honeypots. Its main advantage lies in the adaption of currently available and in-use technology that can be found on most nuclear sites globally. By following some very simple steps, employing freely available software (NetToPLCSim & Wireshark), a high-fidelity honeypot can be easily and quickly deployed. Although this work gave insightful intelligence about the current threat environment, using a greater number of honeypots to give data about the background noise vs potentially significant traffic, more definitive conclusions could be reached. The analysis pipeline could be improved with more automation, this would also allow the addition of automated alerting for unusual activity, this would be especially useful for an internal deployment. Although detailed interactions by attackers are still rare, with traffic being dominated by automated scanners, a pattern of increasing sophistication and frequency of the outlier interactions can be seen by comparing previously reported results to these results. This is an important field for the nuclear industry to stay abreast of - the increasing attacks on OT systems pose a threat to sites with systems becoming more accessible each year as IT/OT connectivity increases. Honeypots allow the volume of threats to be monitored as well as giving an insight into the types of attacks that sites may face. It is likely that what is being seen by this sort of direct internet facing deployment is the tip of the threat iceberg; sophisticated attackers will be working in a more targeted manner and are likely to be using much more advanced methods than those used on random exposed PLCs. However, if the skills of these less sophisticated attackers are growing rapidly so will the already more advanced skills of high-level attackers. This is also true of the ability to find flaws in honeypot implementations; just in this small piece of research new flaws in existing implementations were found, meaning attackers will also be able to find them, highlighting the importance of continuing development.

References

[1] D. Pliatsios, P. Sarigiannidis, T. Liatifis, K. Rompolos, and I. Siniosoglou, “A novel and interactive industrial control system honeypot for critical smart grid infrastructure,” in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, IEEE, 2019, pp. 1–6.

[2] M. Sapir, U. Katz, N. Moshe, S. Brizinov, and A. Preminger, “White paper: Evil plc attack: Weaponizing plcs,” Claroty, Tech. Rep., Aug. 2022. [Online]. Available: https://claroty.com/team82/research/whitepapers/evil-plc-attack-weaponizing-plcs.

[3] J. Slowik, “Evolution of ICS attacks and the prospects for future disruptive events,” *Threat Intelligence Centre Dragos Inc*, 2019.

[4] Dragos, “White paper: Pipedream: Chernovite’s emerging malware targeting industrial control systems,” Dragos Incorporated, Tech. Rep., May 2022. [Online]. Available: https://hub.dragos.com/whitepaper/ chernovite-pipedream.

[5] mushorg, *Conpot github*, 2022. [Online]. Available: https://github.com/ mushorg/conpot.

[6] N. C. Rowe, T. D. Nguyen, M. M. Kendrick, Z. A. Rucker, D. Hyun, and J. C. Brown, “Creating effective industrial-control-system honeypots,” *American Journal of Management*, vol. 20, no. 2, pp. 112–123, 2020.

[7] 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.

[8] D. Hyun, “Collecting cyberattack data for industrial control systems using honeypots,” Ph.D. dissertation, Monterey, California: Naval Postgraduate School, 2018.

[9] 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.

[10] E. L´opez-Morales, C. Rubio-Medrano, A. Doup´e, *et al.*, “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.

[11] SCADACS, *Plcinject*, ”Online; accessed 22nd November 2022”, 2022. [Online]. Available: https://github.com/SCADACS/PLCinject.

[12] meeas, *Plcscan*, ”Online; accessed 22nd November 2022”, 2022. [Online]. Available: https://github.com/meeas/plcscan.

[13] thomas ₋v2, *Nettoplcsim download*, 2022. [Online]. Available: https:// sourceforge.net/projects/nettoplcsim/.

[14] M. Automation, *Nettoplcsim setup instructions*, 2015. [Online]. Available: https://www.mesta-automation.com/nettoplcsim-how-to-connectstep-7-plc-sim-to-scadas/.