A Novel and Interactive Industrial Control System Honeypot for Critical Smart Grid Infrastructure

Dimitrios Pliatsios, Panagiotis Sarigiannidis, Thanasis Liatifis, Konstantinos Rompolos, Ilias Siniosoglou
Department of Informatics and Telecommunications Engineering
University of Western Macedonia
Kozani, Greece
{dpliatsios, psarigiannidis, aliatifis, krobolos, isiniosoglou}@uowm.gr

***Abstract*—The Industrial Control Systems (ICS) are the underlying monitoring and control components of critical infrastructures, which consist of a number of distributed field devices, such as Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs) and Human Machine Interfaces (HMIs). As modern ICS are connected to the Internet, in the context of their digitalization as a part of the Internet of Things (IoT) domain, a number of security threats are introduced, whose exploitation can lead to severe consequences. Honeypots and honeynets are promising countermeasures that attract attackers and mislead them from hacking the real infrastructure, while gaining valuable information about the attack patterns as well as the source of the attack. In this work, we implement an interactive, proof-ofconcept ICS honeypot, which is based on Conpot, that is able to emulate a physical ICS device, by replicating realistic traffic from the real device. As the honeypot runs inside a Virtual Machine, it is possible to emulate the entire organization's ICS infrastructure, a fact that is very important for the security of the modern critical infrastructure. In order to assess the proposed honeypot, a real-life demonstration scenario was designed, which involves a hydro power plant. The honeypot architecture is provided, while the structural components are presented in detail.**

I. Introduction

The Industrial Control Systems (ICS) are the underlying monitoring and control components of critical infrastructures such as telecommunications, transportation, and power grid. The typical ICS architecture consists of a number of distributed field devices, such as actuators and sensors, a number of Programmable Logic Controllers (PLCs) and Remote Terminal Units (RTUs) that control the field devices, and one or more Human Machine Interfaces (HMIs), which gather the data received from the field devices and presents them to the operator in an appropriate manner.

Contrary to the legacy ICS, which operate in isolated enviroments, the current ICS are connected to the Internet. This enables operators to access the system remotely, while the administration of the underlying network use common computer network protocols for controlling and managing the underlying systems. The connection of ICS to the Internet introduces a number of security vulnerabilities [1]. An adversary can exploit these vulnerabilities in order to launch a cyberattack against an ICS, resulting in severe consequences. For example, an adversary can compromise an ICS controlling power and water services or destroy military infrastructure. A classification of a number of security incidents against critical ICS infrastructure is shown in [2].

A large number of methods have been proposed as effective countermeasures against ICS cyberattacks [3]. Honeypots and honeynets are promising countermeasures that are leveraged in both common and industrial networks in order to attract attackers and mislead them from hacking the real infrastructure, while gaining valuable information, such as the origin of the attacker, the employed methods and the attack patterns [4].

In this paper a novel, interactive and proof-of-concept ICS honeypot is introduced, which is based on the Conpot framework. The Conpot framework is able to emulate a physical device by replicating realistic traffic from the real devices. As the proposed honeypot runs inside a Virtual Machine (VM), it is possible to emulate the entire organization's ICS infrastructure. The introduced proof-of-concept ICS honeypot is applied in a real-world hydro power plant, which uses an RTU-based control center to manage all the underlying systems of the plant. The installed ICS honeypot is able to emulate the hydro power plant RTU device by interactively communicating to the HMI, which is located at the control center of the plant. The architecture of the ICS honeypot is presented in details, while the capabilities of the honeypot are also demonstrated. The contributions of this paper are summarized as follows:

- A novel, interactive and proof-of-concept ICS honeypot, which is able to attract various external attacks by emulating any smart grid device using Modbus.
- An efficient ICS architecture, where the proposed proof-of-concept ICS honeypot is applied.
- A real-world use case scenario, in the context of the SPEAR project, where one or more ICS honeypots are able to emulate various IoT-enabled control and management deices.

A. Related Work

The notion of honeypots is quite popular in the literature. A survey of recent advances and future trends in honeypot research is carried out in [5]. The survey suggests that honeypot research is on the rise due to the increasing number of connected devices. Moreover, research honeypots generate valuable data that are used to improve and develop new

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), 2019, doi: 10.1109/camad.2019.8858431.

The published version is available at IEEE Xplore Digital Library: https://ieeexplore.ieee.org/document/8858431

TABLE I
SUMMARY OF RELATED WORK

Reference	Aim	ICS communication protocols	Method/Implementation	Results/Findings
[5]	Investigate emerging trends regarding honeypot research and detect knowledge gaps in honeypot environments by surveying the state of art in honeypot deployments	N/A	Survey on honeypot state of the art	1)Honeypot research is rising as the number of connected devices is growing. 2)Researchers aim to improve and develop new honeypots based on the data obtained from research honeypots, 3)The honeypot must be configured in a way that attracts attackers and keeps them engaged, 4)The legal and ethical concerns is an important research area
[6]	Discuss how the different approaches to security of typical information systems and industrial control systems lead to the need of specialized SCADA honeypots for process control network, and propose a reference architecture for ICS honeypots, and discuss implementation and deployment strategies	Modbus, SNMP, FTP	Implementation and comparison of two approaches. In the first approach the honeypot is hosted on a low-cost physical machine, while in the second one the honeypot is hosted on virtual machines	1)Low-cost machines such as Raspberry Pi provide more than enough resources. 2)In cases where the location of the honeypot is irrelevant to the security point of view, the virtual honeypots are more flexible and cost-effective
[7]	Present a novel ICS honeypot architecture that can be globally deployed in an automated manner, and implement a proof-of-concept ICS honeynet	Modbus, IEC 60870-5-104	Deployment of a modular and scalable honeynet architecture on Amazon EC2 cloud platform. The honeynet consists of multiple distributed honeypot nodes that are managed by a central entity.	The real-world experiments conducted on a cloud infrastructure show the feasibility of the proposed approach. The results highlight the fundamental requirement of having proper security mechanisms in ICSs environments
[8]	Present the design of a high-interaction ICS honeypot that aims to address the main challenges related to ICS requirements	N/A	Implemented they proposed honeypot based on the MiniCPS framework	The deployed a water treatment testbed in the Capture The Flag competition, highlighted the importance of an automated and reliable crash recovery mechanism
[9]	Present DiPot, a distributed honeypot system in order to capture and analyze suspicious behaviors against ICS. DiPot emphasizes on the coverage and diversity for ICS devices, aiming to collect useful data without being identified by attackers	Modbus, Kamstrup, SNMP, BACnet, S7comm	DiPot consists of three nodes, the Honeypot Node which emulates an ICS device, the Data Processing Node which periodically pulls the raw log files from the HNs in order to analyze them, and the Management Node which manages the user interaction and data visualization	Over the span of 6 months, DiPot captured and analyzed large amounts of legitimate and malicious network traffic. The results indicate the effectiveness of DiPot as a distributed honeypot system
[10]	Design, develop and deploy a honeypot, representing a water treatment plant, that feeds intelligence to real-world ICS security services	Modbus, S7comm	The iHoney ICS honeypot consists of three modules: the ICS system module, which includes an HMI and a network of PLC devices, the simulation system which evaluates the process status variables in real-time and interacts with the ICS system, and the cybersecurity monitoring infrastructure that obtains information about the behavior of cyber attackers.	The proposed honeypot is continuously providing security intelligence and insights, regarding IDS signatures, correlation rules, and general awareness of the cyber threat landscape to real-world ICS security services

honeypots. Finally, the legal and ethical concerns of honeypot usage is an important research area.

The utilization of honeypots in ICS environments along with implementation and deployment strategies are investigated in [6]. Additionally, two schemes of an ICS honeypot system were implemented and compared. In the first scheme the honeypot is hosted on a physical device, whereas in the second one, the honeypot is hosted on virtual machines. The findings suggest that low-cost machines can provide enough computational resources, and in cases where the location of the honeypot is irrelevant, the virtual honeypots are more flexible and cost-effective.

The authors in [7] presented the architecture of a novel honeypot, tailored to the requirements of industrial applications. They also deployed a modular and scalable honeynet architecture on the Amazon EC2 cloud platform. The real-world experiments support the feasibility of the proposed approach and highlight the fundamental requirement of having proper security mechanisms in ICS environments.

In [8] the authors present the design of a high-interaction ICS honeypot that aims to address the main challenges related to ICS requirements. They also propose an attacker model for the honeypot that captures the goals, skills resources and entry points of the attacker. In addition, the authors present

an implementation of the proposed ICS honeypot based on the MiniCPS framework. For the evaluation of the honeypot, they deployed a water treatment testbed in a Capture The Flag (CTF) competition hosted by Singapore University of Technology and design. Apart from the captured cyber attack traces, the importance of an automated and reliable crash recovery mechanism was highlighted.

The authors in [9] proposed DiPot which is a distributed industrial honeypot system that provides deep data analytics and advanced visualization techniques. DiPot is a modular honeypot that consists of three nodes, namely the honeypot node which emulates an ICS device, the data processing node that periodically analyses raw log files, and the management node which facilitates user interaction and data visualization. During the evaluation of Dipot, large amounts of legitimate and malicious network traffic were captured and analyzed, thus proving its effectiveness.

Navarro et al. [10] designed an ICS honeypot that is used to collect and feed intelligence to real-world ICS cyber security monitoring services. The honeypot consists of the ICS system module which emulates the HMI and the PLC devices, the simulation system that evaluates the process status variables in real time, and the cybersecurity monitoring infrastructure that obtains information about the behavior of cyber attackers. The proposed honeypot continuously provides security intelligence and insights regarding DIS signatures, correlation rules, and general awareness of the cyber threat landscape to real-world ICS security services.

The leverage of honeypots is an effective asset in the protection of ICS environments. The honeypots are constantly improving in order to attract attackers and keep them engaged long enough to analyze their behavior and motives. Honeypots can be realized on low-cost physical machines (such as a Raspberry Pi), as well as virtual ones, which are more flexible and cost-effective. The evaluation results suggest that the honeypots achieve good performance in terms of detection accuracy and detectability. However, the aforementioned works feature honeypot systems that emulate ICS devices, while no particular attention was paid to the usage of real device data. Our proposed honeypot system replicates the data from the physical devices, thus improving its chances to attract potential attackers.

The rest of the paper is organized as follows. Section II introduces the required background of honeypots and presents the Conpot honeypot, which is used in this paper. Section III describes the architecture and the configuration of its components. Section IV describes the demonstration scenario. Finally, Section V concludes the paper.

II. Background

A. Definition and classification of honeypots

A honeypot can be considered as a closely monitored network decoy that attracts attackers in order to prevent them from attacking real infrastructure, provides an early alarm about the attack, and collects information that can be analyzed in order to gain valuable insights about novel attacks and exploitation trends [11]. It should be highlighted that honeypots have minimal effectiveness regarding the prevention of a cyberattack, so they should be better deployed alongside an IDS [12].

Honeypots can be classified based on their purpose and level of interaction. Depending on their purpose, honeypots can be classified into production honeypots and research honeypots. Production honeypots are primarily used in corporate and industrial organizations, aiming to increase security and mitigate the risks. They usually mirror the organization's network infrastructure, inviting attackers to interact with it in order to expose currently unknown vulnerabilities of the network. Contrary to the research honeypots, the production honeypots are easier to build and install.

Research honeypots aim to gather information about the motives and tactics of cyber-attackers and do not add any direct value to an organization. Their primary aim is to investigate the attack patterns and understand the attacker's motives and behavior. Research honeypots provide significant contribution as a platform to study cyber-attacks. As they include additional functionality compared to production honeypots they are more complex and difficult to deploy and manage.

Depending on their interaction level honeypots can be categorized into low-interaction, medium-interaction, and high-interaction. A low-interaction honeypot only simulates certain services and there is no operating system that the attacker can exploit to gain access to it. This minimizes the risk of the honeypot being compromised but restricts the functionality of honeypot. Low-interaction honeypots are easy to deploy and manage.

Similarly to low-interaction honeypots, medium-interaction honeypots do not include an underlying operating system. However, they simulate more advanced application and network services, providing the attacker with a better illusion of interacting with a real system. This enables the collection and analysis of more sophisticated cyber-attacks.

Finally, the high-interaction honeypots are the most advanced ones as they also incorporate an operating system. Due to their high-interaction functionalities, they manage to act as a real system, capable of attracting attackers and collecting large amounts of information. This type of honeypots has increased difficulty in the deployment as they require higher storage and processing resources. Additionally, they require continuous monitoring to ensure that the honeypot is not compromised, hence becoming a vulnerability to the whole network.

B. Conpot Honeypot

Conpot is a low-interactive ICS honeypot, which is easily deployed, modified, and extended [13]. It supports a range of well-known ICS communication protocols, thus enable users to build and emulate complex realistic infrastructures that can attract cyber-attackers. Emphasis has been given to the response times of services in order to mimic the behavior of a complex system under constant load. Conpot monitors any intruder traces by utilizing a logging system, offering basic tracking information, such as source addresses and

request types. Finally, as the Conpot is equipped with complete stacks of the supported communication protocols, it is able to interconnect real RTUs, PLCs, and HMIs. Supported industrial communication protocols include the Modbus, the Distributed Network Protocol Version 3 (DNP3) [14], the IEC 60870-5-104 [15], and BACNET [16].

C. Modbus Communication Protocol

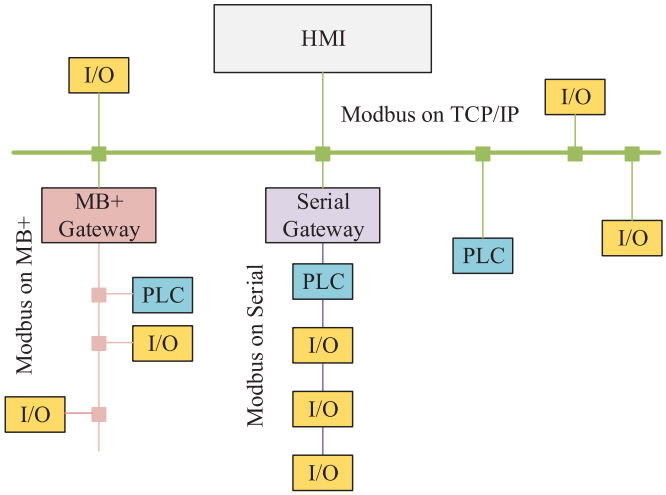

Fig. 1. Modbus Network Architecture

Modbus [17] is a widely used industrial communication protocol, due to its open specifications, easy deployment and maintenance procedures. Fig. 1 shows a reference architecture of a Modbus network. Different types of field devices (e.g., PLC, HMI, and I/O devices) can connect to the network by using different Modbus variants such as Modbus+, Serial, and TCP/IP. The MB+ and Serial Gateways are used as converters between the Modbus variants. Serial Modbus enables communication among master and slave devices using serial communication mode. The master device coordinates the communication, while the slave devices monitor the channel for requests from the master and respond accordingly.

The structure of a Modbus frame is illustrated in Fig. 2. The Application Data Unit (ADU), includes a Protocol Data Unit (PDU) along with fields reserved for device addressing and error checking. The PDU consists of the function code field, which is used to select the operation, while the data field size depends on the selected function. In the Modbus serial variant, the addressing field contains the slave ID and utilizes CRC for error detection. In the TCP/IP variant of Modbus, the addressing field is replaced by the Modbus Application Protocol (MBAP) header, while the error check field is removed. The MBAP header includes the following fields: the transaction identifier, which is used for logically pairing the transactions that are carried out in the same TCP stream. The protocol identifier is always set to 0, while the length field indicates the size of the remaining fields. Finally, the unit identifier is used to identify hosts that belong to networks, for example in case of bridging TCP/IP and serial

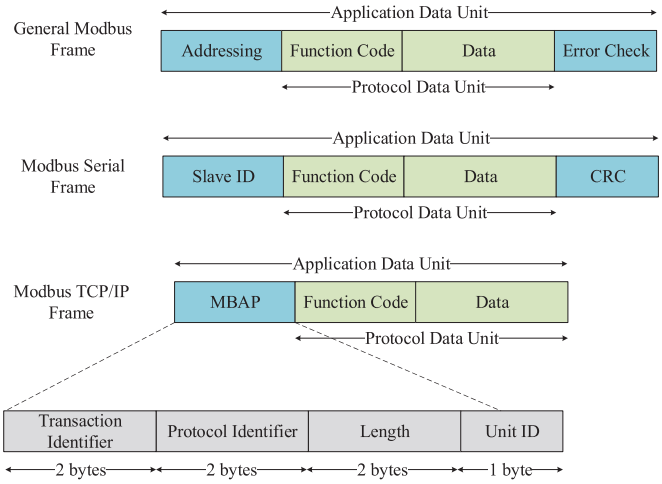

Fig. 2. Modbus Frame Structure

TABLE II
FUNCTION CODE DESCRIPTION

Function Code	Size	Operation
01	1 bit	Read Memory Bit
02	1 bit	Read Device Input Bit
03	16 bits	Read Memory Words
04	16 bits	Read Device Input Word
05	1 bit	Write Memory Bit
06	16 bits	Write Memory Word
15	1 bit	Write Multiple Memory Bit
16	16 bits	Write Multiple Memory Words
23	16 bits	Read/Write Multiple Memory Words

networks. The complete Modbus ADU is encapsulated into the data field of a standard TCP/IP frame.

A description of the Modbus supported function codes is shown in Table II. The function code column lists the corresponding number of each code, while the Size column lists the size of each payload. Finally, the function of each code is described in the operation column.

III. Design and Implementation

The architecture of the proposed ICS is presented subject to the implemented testbed, as shown in Fig. 3. It is composed of the following components: a) the real HMI, which is used to gather and present the collected data to the operator, b) the Schneider Electric Saitel HU_AF (RTU), which reads the analog outputs from the smart meters, and c) a Virtual Machine (VM) Manager that manages two virtual machines: the Conpot-VM that emulates the Saitel RTU in order to attract potential attackers, and the Virtual HMI that generates requests to the Conpot VM. Finally, a switch is used to enable the interconnection of the aforementioned components. In order to improve the illusion that the Conpot is a real device, we configured it to utilize real traffic collected from the Saitel RTU.

In order to generate realistic traffic, the Virtual HMI generates requests to the Conpot-VM. The Python code of the

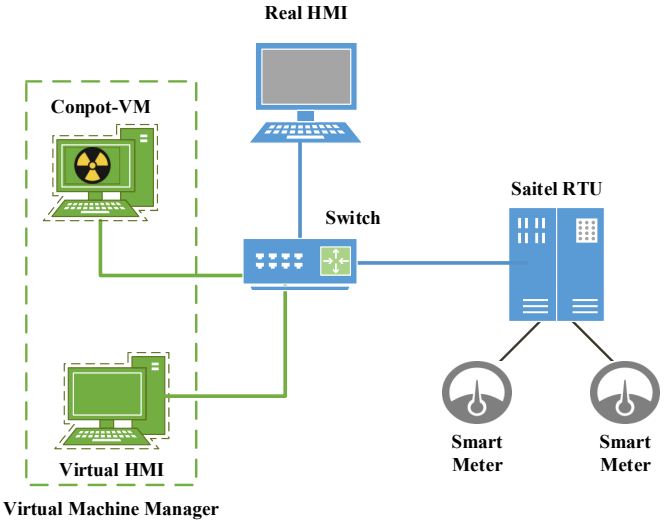

Fig. 3. Testbed Architecture

Virtual HMI is shown in Listing 1. The Modubs server address (line 2) is set to the Conpot's IP. The captured traffic from the real devices is exported to a csv file using Tshark. An instance of the file is shown in Table III. For each data frame, a socket connecting to the Conpt is initialized and a PDU structure is created (e.g., line 15). The PDU structure is included in a Modbus TCP query (line 21) and transmitted to the Conpot (line 22).

```
1  parser = argp.ArgumentParser()
2  parser.add_argument("—modbusip", help="The modbus
       server ip address",required=True)
3  parser.add_argument("—trafficfile", help="The csv
       file that will be used to mimic the traffic",
       required=False)
4  arguments= vars(parser.parse_args())
5
6  dataframe= pandas.read_csv(arguments["trafficfile"
       ])
7  while True:
8      for i in range (0, dataframe.shape[0]):
9          start_address= dataframe.iloc[i]["
       StartAddr"]
10         quantity= dataframe.iloc[i]["Quantity"]
11         slave_id= dataframe.iloc[i]["SlaveID"]
12         modbus_server_socket= modbus_tcp.TcpMaster
       (host=arguments["modbusip"])
13         query= modbus_tcp.TcpQuery()
14         if dataframe.iloc[i]['Function Code']==1:
15             pdu= struct.pack(">BHH",1,
       start_address,quantity)
16         elif dataframe.iloc[i]['Function Code'
       ]==2:
17             pdu= struct.pack(">BHH",2,
       start_address,quantity)
18         elif dataframe.iloc[i]['Function Code'
       ]==3:
19             pdu= struct.pack(">BHH",3,
       start_address,quantity)
20         #... The rest of the function codes are
       not shown in this example ...#
21         request=query.build_request(pdu,slave_id)
22         modbus_server_socket._send(request)
23         response= modbus_server_socket._recv()
24 modbus_server_socket.close()
```

Listing 1. Virtual HMI

The traces of the traffic between the real RTU and the real HMI are collected using Wireshark [18] and stored into pcap files. A custom developed tool is used to parse pcap files in order to feed them to the Conpot engine. In this way, Conpot manages to better imitate the behavior of its real counterpart.

TABLE III
TRAFFIC FILE EXAMPLE

Data Frame	Slave ID	Function Code	Start Address	Quantity
1	2	3	0	2
2	1	1	0	2
3	1	3	4	1
4	2	2	4	1
5	1	2	5	1

Fig. 4 shows an instance of the captured traffic between the Virtual HMI and Conpot. The communication session begins with the TCP handshake as shown in step 1. Then the virtual HMI sends a "Read Input Registers" Modbus/TCP packet, followed by the corresponding TCP Acknowledgment (ACK) packet from Conpot (step 2). Conpot replies with the appropriate response packet, followed by the corresponding TCP ACK packet from the HMI (step 3). Finally, the communication session terminates.

IV. DEMONSTRATION & ASSESSMENT

The demonstration scenario aims to address one of the critical infrastructures defined by the European Program for Critical infrastructure Protection [19]. Within the context of SPEAR H2020 project, a real-world hydro power plant was selected in order to test and evaluate our honeypot. The hydro power plant control center features some unique characteristics such as a) it represents an example of renewable energy utility, b) it requires supreme technical skills and knowledge, thus it constitutes a high-technology plant where any vulnerabilities could harm major components of the smart grid infrastructure, c) it enables a high-cost way of generating hydro-electric power, hence the scenario has increased impact in terms of hardware failures and d) it constitutes a roadmap in order to validate the SPEAR architecture towards securing renewable energy smart grid utilities.

Fig. 5 illustrates a hydro power plant scenario. The power generation department includes the main power plant equipment that is related to the generation of electricity out of the river. The infrastructure includes the turbines, the generator, the electrical systems, and the transformer. Smart devices and smart meters govern this equipment by using IP interfaces and API systems. The Conpot system is installed in a VM and emulates the behavior of the RTU controlling the smart meters. In addition, a second VM emulates an HMI and generates realistic traffic in order to attract potential attackers.

0.127335829	192.168.1.4	192.168.1.5	TCP	} 1	76 56284 → 502 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1037592608 TSecr=0 WS=128
0.127923412	192.168.1.5	192.168.1.4	TCP		76 502 → 56284 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=894660909 TSecr=1037592608 WS=128
0.127931994	192.168.1.4	192.168.1.5	TCP		68 56284 → 502 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1037592609 TSecr=894660909
0.128039468	192.168.1.4	192.168.1.5	Modbus/TCP	} 2	80 Query: Trans: 1; Unit: 1, Func: 4: Read Input Registers
0.128233345	192.168.1.5	192.168.1.4	TCP		68 502 → 56284 [ACK] Seq=1 Ack=13 Win=29056 Len=0 TSval=894660910 TSecr=1037592609
0.131040136	192.168.1.5	192.168.1.4	Modbus/TCP	} 3	81 Response: Trans: 1; Unit: 1, Func: 4: Read Input Registers
0.131046369	192.168.1.4	192.168.1.5	TCP		68 56284 → 502 [ACK] Seq=13 Ack=14 Win=29312 Len=0 TSval=1037592612 TSecr=894660913
5.134934237	192.168.1.5	192.168.1.4	TCP		68 502 → 56284 [RST, ACK] Seq=14 Ack=25 Win=29056 Len=0 TSval=894665914 TSecr=1037597611

Fig. 4. Captured Traffic Between Virtual HMI and Conpot

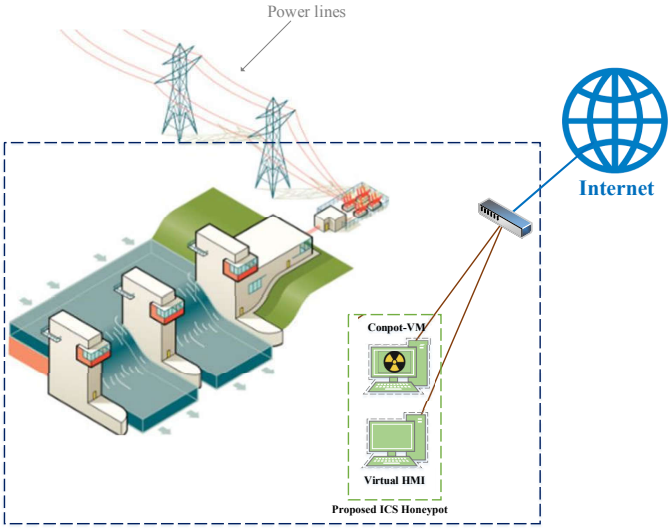

Fig. 5. VETS Demonstration Scenario

V. Conclusion

In this paper, we implemented a real-world, proof-of-concept ICS honeypot based on Conpot. The proposed honeypot emulates a Modbus-enabled physical device using real traffic from real devices. In order to generate network traffic a custom HMI emulator was developed that communicates with the honeypot. The ability to replicate the entire ICS infrastructure and the utilization of real traffic data increases the probability to attract cyber-attackers. As a future extension, we aim to design and execute a demonstration scenario in order to validate the effectiveness and accuracy of the proposed honeypot. The demonstration scenario involves running the proposed honeypot over a span of at least six months. The collected data will be stored into a repository of attacks, where further analysis will be carried out.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 787011 (SPEAR).

References

[1] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security issues in scada networks," *computers & security*, vol. 25, no. 7, pp. 498–506, 2006.

[2] R. I. Ogie, "Cyber security incidents on critical infrastructure and industrial networks," in *Proceedings of the 9th International Conference on Computer and Automation Engineering*. ACM, 2017, pp. 254–258.

[3] M. Korman, M. Välja, G. Björkman, M. Ekstedt, A. Vernotte, and R. Lagerström, "Analyzing the effectiveness of attack countermeasures in a scada system," in *Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*. ACM, 2017, pp. 73–78.

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

[5] R. M. Campbell, K. Padayachee, and T. Masombuka, "A survey of honeypot research: Trends and opportunities," in *2015 10th international conference for internet technology and secured transactions (ICITST)*. IEEE, 2015, pp. 208–212.

[6] P. Simões, T. Cruz, J. Proença, and E. Monteiro, "Specialized honeypots for scada systems," in *Cyber Security: Analytics, Technology and Automation*. Springer, 2015, pp. 251–269.

[7] A. V. Serbanescu, S. Obermeier, and D.-Y. Yu, "A flexible architecture for industrial control system honeypots," in *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*, vol. 4. IEEE, 2015, pp. 16–26.

[8] 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*. ACM, 2016, pp. 13–22.

[9] 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.

[10] Ó. Navarro, S. A. J. Balbastre, and S. Beyer, "Gathering intelligence through realistic industrial control system honeypots," in *International Conference on Critical Information Infrastructures Security*. Springer, 2018, pp. 143–153.

[11] I. Mokube and M. Adams, "Honeypots: concepts, approaches, and challenges," in *Proceedings of the 45th annual southeast regional conference*. ACM, 2007, pp. 321–326.

[12] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[13] 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.

[14] IEEE Power and Energy Society, "IEEE Standard for Electric Power Systems Communications - Distributed Network Protocol (DNP3)," Oct 2012.

[15] Y. Ju and H.-G. Zhang, "Design and application of iec 60870-5-104 telecontrol protocol." *Relay*, vol. 34, no. 17, pp. 55–58, 2006.

[16] H. M. Newman, *BACnet: The Global Standard for Building Automation and Control Networks*. Momentum Press, 2013.

[17] Schneider Automation Inc, "MODBUS Application Protocol Specification," Apr 2012, v1.1b3.

[18] A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.

[19] European Program for Critical Infrastructure Protection (EPCIP). Protection of critical infrastructure. [Online]. Available: https://ec.europa.eu/energy/en/topics/infrastructure/protection-critical-infrastructure