HoneyVP: A Cost-Effective Hybrid Honeypot Architecture for Industrial Control Systems

Jianzhou You*†, Shichao Lv*†, Yue Sun*†, Hui Wen*†, Limin Sun*†
*Beijing Key Laboratory of IoT Information Security Technology, IIE CAS, Beijing 100093, China.
†School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China.
Emails: {youjianzhou,lvshichao,sunyue0205,wenhui,sunlimin}@iie.ac.cn

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500567

***Abstract*—As a decoy for hackers, honeypots have been proved to be a very valuable tool for collecting real data. However, due to closed source and vendor-specific firmware, there are significant limitations in cost for researchers to design an easy-to-use and high-interaction honeypot for industrial control systems (ICSs). To solve this problem, it's necessary to find a cost-effective solution. In this paper, we propose a novel honeypot architecture termed HoneyVP to support a semi-virtual and semi-physical honeypot design and implementation to enable high cost performance. Specially, we first analyze cyber-attacks on ICS devices in view of different interaction levels. Then, in order to deal with these attacks, our HoneyVP architecture clearly defines three basic independent and cooperative components, namely, the virtual component, the physical component, and the coordinator. Finally, a local-remote cooperative ICS honeypot system is implemented to validate its feasibility and effectiveness. Our experimental results show the advantages of using the proposed architecture compared with the previous honeypot solutions. HoneyVP provides a cost-effective solution for ICS security researchers, making ICS honeypots more attractive and making it possible to capture physical interactions.**

***Index Terms*—honeypot, high-interaction, ICS**

I. Introduction

Today, industrial control systems (ICSs) have become an essential part of the country's critical infrastructure, such as power grids, gas pipelines, and even aerospace. Computer network modernizes conventional industry by ICS devices, like Remote Terminal Units (RTUs), Programmable Logic Controllers (PLCs) and Intelligent Electronic Devices (IEDs). ICS devices use the Transport Control Protocol and Internet Protocol (TCP/IP) stack to exchange data, but additional security concerns may rise because of this convergence [1]. With the networking trend for the industry, large amounts of ICS devices appear online with little security measures to provide features such as real-time remote monitoring, remote plant maintenance and control, as well as the collection of cyber-physical data for analysis purposes in third-party and cloud-based systems. Shodan [2], a popular Internet scanning engine, lists more than 2000 endpoints running a variety of the S7comm protocol. This exposure has sparked the interest of malicious actors and researchers alike in performing scanning, reconnaissance and in some cases attacks targeted at ICS devices [3].

Defending ICSs in the cyber domain is both helped and hindered by bespoke systems integrating heterogeneous devices. Because of this fragmentation, how to capture more various and sophisticated cyber-attacks has become a challenge in the context of the ICS security. In recent years, cyber-attacks on ICS devices go very vigorously and cause a high amount of damage. It is difficult to prevent ICS from different malicious activities as the components of ICS will not be able to update or patch due to its demand. If attackers compromise an ICS device, they can disrupt the normal control of a physical process and cause a catastrophe, such as environmental damage and loss of life.

Honeypot is a vulnerable system that is set up with the intent to be probed and compromised by attackers. In order to build heterogeneity ICS services as decoys for attackers, each industrial protocol or firmware used in ICS devices should be analyzed using reverse engineering. However, the development cost of ICS honeypots constrained the further increases in fidelity and scalability. This dilemma forced us to seek an innovative way to design honeypot for ICS devices. Thus, we propose a semi-virtual and semi-physical ICS honeypot architecture enabling high cost efficiency. Overall, the major contributions of this work are summarised as follows:

- We reveal the cost limits in both virtual and physical honeypot design and describe two types of cyber-attacks on ICS devices.
- HoneyVP, an cost-effective honeypot architecture consisting of three components, is proposed to rationalize costs and improve effectiveness for the purpose of designing an easy-to-use and high-interaction ICS honeypot.
- A proof-of-concept system is implemented in terms of the local-remote cooperative honeypot design, which responds at two different interaction levels and maximizes the hardware utilization.
- We conduct experiments on the proof-of-concept system to verify the fidelity, scalability and cost of the honeypot architecture, and to evaluate the cost efficiency.

The remainder of the paper is outlined as follows: Section II introduces types of honeypots and cyber-attacks on ICS devices. Section III describes the architecture of HoneyVP in detail. Section IV describes a proof-of-concept implementation of HoneyVP and demonstrates the system evaluation and the experimental results. Furthermore, Section V discusses the difference between our approach and existing works, and Section VI summarizes this paper.

Shichao Lv*† is the corresponding author.

II. Background

A. Types of Honeypots

Honeypots can be classified as virtual honeypots and physical honeypots. Virtual honeypots are not limited by specific physical hardware, which means they can be designed to be compatible with any platform, like cloud servers, personal computers or even a Raspberry Pi. The main concern of these honeypots is to imitate the original services as far as possible. Virtual honeypots are quite popular in traditional computer security due to large amounts of available software packages and mature virtualization technology. But in ICS's context, the key challenges for virtual honeypots are summarised as follows:

- High development cost. In order to build heterogeneity ICS services, each proprietary protocol or firmware used in ICS should be analyzed using reverse engineering, which consumes huge manpower and time.
- Hard to achieve high-interaction. Because these protocols or firmware are often closed source and vendor-specific, there will still remain a huge gap between a non-firmware-driven simulator and a physical ICS device no matter how hard researchers work.
- Easy to identify. It's easy for adversaries to identify a virtual honeypot through unimplemented interaction or default configuration information.

Physical honeypots rely on physical devices to complete the perfect interaction [4]. In ICS's context, the key challenges for physical honeypots are summarised as follows:

- High hardware cost. Compared to regular computers, ICS devices are often heterogeneous and far more expensive. It also means we need to purchase different ICS devices to improve compatibility for different manufacturers and versions.
- Restricted deployment scope. On the one hand, for local address space, it is a huge cost to increase the number of physical honeypots. On the other hand, for remote address space, it is impractical to set up a physical honeypot.
- High requirement for self-protection. Physical honeypots often imply high-interaction, thus allowing the system to be compromised completely. Researchers may have to establish additional security measures to protect physical devices.

Neither physical honeypots nor virtual honeypots seem to be a good solution for ICS security. For one thing, it was considered not practical to use physical ICS honeypots due to the limited locations and financial restrictions, not to mention the risks of being abused. For another, a virtual honeypot is software that emulates a vulnerable system or network. But the fact is that, unlike operating systems (e.g. Android, Windows), the majority of ICS do not have any emulator available. It led to the low interaction capability for most virtual honeypots. In this paper, we propose a semi-virtual and semi-physical ICS honeypot architecture enabling high cost efficiency.

B. Cyber-attacks on ICS devices

As an industrial digital computer, ICS devices have been ruggedized and adapted both for the control of manufacturing processes and ethernet network connections. Therefore, research of ICS devices can be conducted from two different perspectives. In the control space, ICS devices like general-purpose controllers use data taken from a collection of inputs, process them through a series of pre-programmed logic, and send a physical output based on the results of that logic. In cyber space, ICS devices with IP addresses are accessible via Ethernet to provide features such as real-time remote monitoring, remote plant maintenance and control. These two spaces also imply the two focal points in recent ICS cyberattack research, which are summarised as follows:

- Reconnaissance. Research [5] in the cyber space often indicates the ICS cyber-attacks as reconnaissance in the wild. Such research characteristically puts less emphasis on the control system operating state by essentially focusing on the cyber (e.g., communication networks, protocols, data, etc.) perspective. ICS cyber-attacks are summarized as some probing activities based on different industrial protocols, like Modbus, S7comm, Bacnet. Device search engines (e.g., Shodan) and potential attackers regularly use specific payload in the application protocol data unit to acquire the device's status in the wild.
- Physical Interactions. Research [6] in control space often considers the influence of the system operating state from a physical point of view to perform their analysis. Their goal is to analyse some high-risk ICS cyber-attacks that could use control programs to affect the control flow. If successful, they may cause a catastrophe.

III. HoneyVP Architecture

In this section, we first define the three advanced ICS honeypot features as our requirements, then we will present the HoneyVP architecture and explain how it facilitates these three features. Finally, we will highlight the novelty of our work by comparing it with literature in terms of the contribution to the three features.

A. Three Key Features

As aforementioned, the features named fidelity, scalability and cost are vitally important to the ICS honeypot for the purpose of catching high quality attack data on a large scale. In this part, we provide their definitions in the context of ICS honeypot system as follows:

- **Fidelity.** Only high-interaction honeypots will convince skilled attackers to use advanced attack methods while exploiting zero-day vulnerabilities. Conventional honeypots use virtualization or simulation approaches to replicate the device under attack and attract attackers. In the context of ICS, physical honeypot ought to be the best way to achieve high-interaction capability.
- **Scalability.** There are often various versions of the ICS device from one manufacturer. For attackers, different

versions devices mean different profile information. Because physical devices are embedded in the physical world, physical honeypots are often unscalable to change their profile information, which is very useful for researchers to study different attack vectors.

- **Cost.** The cost for ICS honeypots can be divided into hardware cost, development cost and time cost. Industrial technology hardware (eg. PLCs) is much more expensive than regular computers. There are also very few available software resources which cause the high development cost and long development period in developing ICS honeypots. Therefore, it is necessary to propose a suitable scheduling mechanism to make the best use of limited resources and improve cost efficiency.

B. HoneyVP

The HoneyVP decouples the virtual component and physical component from the architectural point of view, and by using the coordinator to coordinate them, it can efficiently enable high cost-efficiency design. As Figure 1 shows, we propose a novel honeypot system architecture that decouples different interactions into two separate components, where they take responsibility for handling different level interactions, and they are scheduled by the coordinator so as to work them together in a more flexible way. In the following subsections, each component is described in detail respectively.

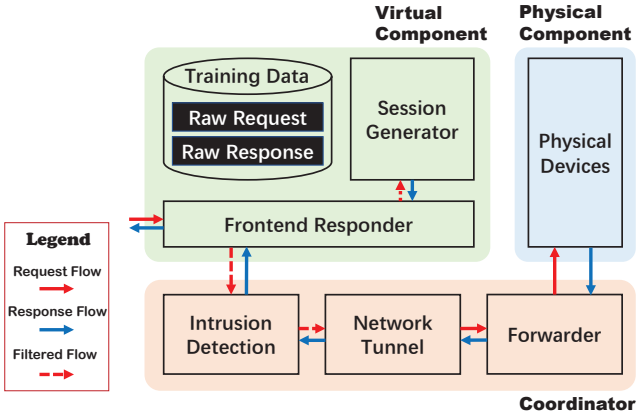

Fig. 1. An overview of the HoneyVP architecture

1) Virtual component: The virtual component is responsible for interacting with attackers by handling all incoming requests. There are 3 major parts running in the virtual component. First, a database of training data stored raw traffic we obtained regarding real ICS cyber-attacks. Second, a session generator extracts the valid request/response pairs and stores each pair in a session table. Third, a frontend responder matches the response in the session table according to the incoming request and redirects the unmatched one to the coordinator.

2) Physical component: The physical component is in charge of providing support for physical interactions and unknown requests. The physical component is composed of physical ICS devices. Each of them can be a normal commercial off-the-shelf ICS device. It will expose one or more services and connect with the frontend responder via network tunnel. These devices are able to maintain a number of concurrent connections at any point in time. The original design of these off-the-shelf devices may limit the number of simultaneous connections.

3) Coordinator: The coordinator takes the responsibility of coordinating the other two components to work together efficiently. It consists of intrusion detection, network tunnel and forwarder. Intrusion detection leverages multiple heuristics and existing detecting tools to detect the exploit code in matched requests to filter out dangerous ones. Network tunnel ensures that the traffic between the virtual component and the physical component is re-written in real-time to hide the fact that devices are physically located somewhere else, or communicating with many other tunnels. The forwarder is used by the honeypot operator to control the data distribution mechanism in order to coordinate the resources of the virtual component and the physical component.

IV. Evaluation

In this section, a proof-of-concept system is presented in order to conduct the validation experiments. Then we will present the validation of fidelity, scalability and cost. All components of the system are implemented by using specialized tools (see following subsections), which are not limited to those used in this prototype.

A. Proof-of-Concept Implementation and Setup

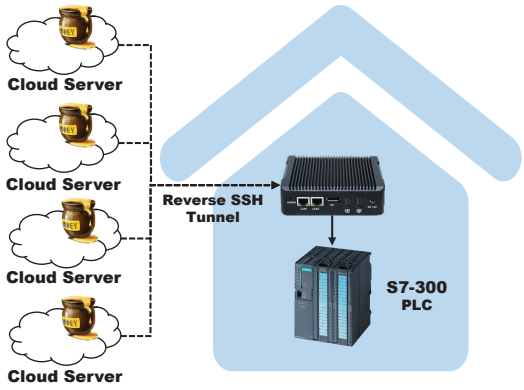

Fig. 2. An Implementation of Proof-of-concept HoneyVP System

As described in Figure 2, we present a local-remote cooperative honeypot system, of which virtual components are running on cloud servers while a physical Siemens S7-300 PLC and a coordinator machine are hosted locally. The remote part contains 4 cloud servers (512MB memory, 1 CPU, 20G SSD) from Aliyun in different network segments (China, Germany, India, Indonesia). We use a coordinator machine(4GB of RAM, running Ubuntu 18.04) in our lab which manages the TCP connections from cloud servers and monitors the

incoming traffic. An off-the-shelf S7-300 PLC is enough for the implementation of programbility and physical interactions.

TABLE I
COVERAGE OF SYSTEM STATUS LISTS

SZL-ID	SZL Name
0x0011	Module identification
0x0111	Module identification (single record)
0x001c	Component Identification
0x0112	CPU characteristics
0x0113	User memory areas
0x0115	Block types
0x0117	List of the permitted SDBs (single record)
0x0f17	List of the permitted SDBs (partial list)
0x0132	Communication status data
0x0018	Maximum S7-300 I/O configuration (all)
0x0118	Maximum S7-300 I/O configuration (single record)
0x001c	Component Identification (all)
0x011c	Component Identification (single record)
0x0591	Module status information (submodules)
0x0a91	Module status information (DP master systems)
0x01a0	Diagnostic buffer of the CPU

The virtual component hosted on the cloud servers is responsible for the interactions with reconnaissance activities, which involves the common requests in the wild. Table I shows our supported response capability for identification entries in the system status lists ("System-ZustandsListen" in German, SZL), most of which originated from Shodan and our previous honeypot-captured data. Additionally, we use OsChameleon[1], a tool that hides the fingerprint of modern Linux kernels from tools such as nmap[2], to provide TCP/IP Stack simulations. Any other requests beyond the SZL list will be redirected to the local physical component. The connection between the coordinator machine and the cloud instances is established with reverse ssh tunnels that redirect traffic of the specific port on the cloud server to a port in the local machine.

B. Validation of Fidelity

Fidelity of HoneyVP can be divided into TCP/IP stack simulation, programbility and physical interactions. Refer to HoneyPLC, we generate a detailed TCP/IP Stack fingerprint for Siemens PLC. And all the requests for progrability and physical interactions will be forwarded to the physical part. Our experiment conducted the Nmap scan with OS dection enabled to test the TCP/IP stack fingerprint and Siemens Step7 Manager [7] to test the programbility and physical interactions. First, we installed Nmap 7.80 and public Siemens Step7 Manager. Second, we perform an OS detection with the flag -O. Third, we use Step7 Manager to perform the following operations: listing memory blocks, uploading a memory block and downloading the contents of a memory block. Finally, we attempt to modify ladder logic programs and debug them. The results show that our proof-of-concept implementation is capable of handling all of the functionality. Thanks to the physical component, the operational capacity can be infinitely close to the real device. But it is important to notice that an intrusion detection system should be implemented to filter uncontrollable malicious traffic in a real attack scenario.

C. Validation of Scalability

Scalability of HoneyVP can be divided into the extensibility in device version, device manufacturer and deployment scope. Our experiment conducted two sets of device identification by Nmap Script[3] toward ICS device from Siemens and Schneider. One set for HoneyVP proof-of-concept implementation and another set for the real device. The devices used for testing are Siemens S7-300 (CPU: 6ES7 313-5BG04-0AB0, 315-2AH14-0AB0), S7-400 (CPU: 6ES7 412-2EK06-0AB0), Schneider Quantum (CPU: 140 CPU 311 10), Schneider (CPU: BMXP342020). The results show that all information acquired from honeypot by Nmap script is the same as the real device. Because this information is often static, related response can be easily generated from the preset session table. Due to the lack of Schneider programming software, only Siemens has proved to be effective as mentioned in section IV-B.

Besides, to obtain more valid profile information for HoneyVP, we can leverage Shodan search engine to gather data of Internet-facing real PLCs' profile and their tag. As Table II shows, we totally extract 18 different identification entries for S7comm protocol by using Shodan API. There are 2766 profiles, among which 1299 are unique and 387 are marked as honeypot. Together with the profile, we also got the raw response data with the help of Shodan API. All these profiles can be loaded into the session table of virtual components and respond to attackers once they match the specific query. And during our experiments all 4 cloud instances were indexed by Shodan with the tag "Industrial Control System".

D. Validation of Cost

Only one developer is enough to take charge of the development, deployment, operation and maintenance. The main cost of HoneyVP prototype comes from the hardware cost of ICS devices. To resolve this issue, an one-to-many mechanism has been used to share the hardware cost. Generally, ICS devices often support concurrency and a S7-300 PLC support at least 16 concurrent queries. Therefore, HoneyVP has achieved the feature of short R&D cycle and medium hardware cost. It is very friendly to ICS security researchers who are suffering from the high cost of ICS honeypots implementation. Although in some cases some attack flow may be disturbed by other attackers, actually in real ICS environment PLCs also are under the control of operators, who may also disturb attackers.

E. Performance Tests

As shown in Figure 3, we use a test case (Sequence 1-5: read SZL, 6-10: write or download) to perform a comparative analysis of response time with respect to virtual honeypots and online physical PLCs. Virtual honeypots always respond in a shorter time while online physical PLC in a longer time. As for HoneyVP, the response time of the first five requests is similar to virtual honeypots. And there will be a leap at the

[1]https://github.com/mushorg/oschameleon

[2]https://github.com/nmap/nmap

[3]https://github.com/digitalbond/Redpoint

TABLE II
PROFILE INFORMATION OF S7COMM EXTRACTED FROM SHODAN

Identification Entries	Counts (Unique)	Value (Top)
Serial number of memory card	66	MMC 267FF11F
Location designation of a module	2	Blank
Copyright	3	Original Siemens Equipment
Manufacturer and profile of a CPU module	3	*
PLC name	323	SIMATIC 300(1)
Module type	40	CPU 315-2 PN/DP
Unknown (129)	91	Boot Loader A
Module	301	6ES7 315-2EH14-0AB0 v.0.4
Basic Firmware	218	v.3.2.6
OEM ID of a module	1	Blank
Module name	116	CPU 315-2 PN/DP
Serial number of module	1067	S C-C2UR28922012
Plant identification	31	Blank
Basic Hardware	300	6ES7 315-2EH14-0AB0 v.0.4
Unknown (130)	15	OS V
Reserved for operating system	1	Blank
Unknown (0)	1	IM151-8 PN/DP
Unknown (128)	3	6ES7954-8LF03-0AA0
⋆ Blank means the response value is blank		

sixth request because of the detection and decision-making which need some time. The response time can be superficially inflated depending on the deployment settings. We adjusted the response time of other requests to be consistent with that of the sixth requests. Though it seems that an attacker can use the response time to differentiate a honeypot from a physical PLC, in a real deployment scenario, overall response time depends on different factors such as network bandwidth, client's processing power, etc. Merely using response time as a differentiating factor is not enough.

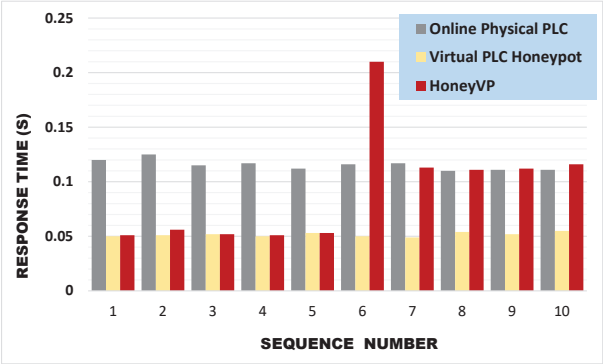

Fig. 3. Response Latency under Different Interaction Levels

F. Live Attack Capture

From January 7 to February 20 in 2020, our honeypots totally captured 9332 S7comm sessions from the Internet. Among these sessions, we observed full S7comm handshakes and queries for typical identification entries and some malformed requests. Beyond that we did not observe suspicious high-risk operation activities. These results are similar to virtual honeypot cases, which means queries at a low interaction level are still the main ICS threat in the wild. On another dimension, physical interactions towards online ICS devices may well be sparse and goal-oriented.

V. DISCUSSION

In this section, we will discuss the difference between our research and existing work.

A. ICS Honeypots

The first ICS Honeypot project [12] was developed by the Cisco Critical Infrastructure Assurance Group (CIAG). Released in March of 2004, it uses Honeyd [13] to simulate the Modus, FTP and HTTP services from a PLC. Conpot, an open-source ICS honeypot, is one of the most famous ICS honeypots products, having the ability to emulate Modbus and Simens S7 server. Most of the researches regard Conpot as a baseline, either to increase the interaction level of specific protocols [14] [15] or optimize network deployment structure [16] [17] [18]. Snap7 is an open source project that provides Ethernet communication with Siemens S7 PLCs and can be used in ICS honeypots [19]. HoneyPhy [8] provides a novel physics-aware model to simulate a generic analog thermostat and the DNP3 protocol. HoneyPLC [9] includes advanced simulations of the most common network protocols found in PLCs, namely, the TCP/IP Stack, S7comm, HTTP, and SNMP, addressing the challenges introduced by inadequate simulations and protocol closeness. However, from the perspective of interaction level, all of these ICS honeypots are virtual honeypots with limited interaction capability, which are caused by the inadequate understanding of private industrial protocols. On the one hand, it will be a huge cost to research, develop and maintain a high-interaction virtual ICS honeypot on a case by case basis. On the other hand, it is too difficult for ICS honeypot to achieve high-interaction under a virtual honeypot architecture.

B. Honeypot Achitecture

As described in the previous section, ICS honeypots mostly adopt the virtual architecture, which is independent and tight coupling in the whole interactive process. Similar to virtual architecture, the interactive process in physical architecture should be tight coupling and all requests are handled by physical devices. Juan Guarnizo et al. proposed a high-interaction physical honeypot framework called SIPHON [10]. This architecture redirects all incoming traffic to the physical devices directly, which may cause the heavy load in physical devices and limited extensibility for an ICS honeypot.

Unlike the virtual or physical architecture, the interactive process under a hybrid honeypot architecture is loose coupling and separated. HoneyDOC [11] is a SDN-enabled architecture that decouple the honeypot to supplies high programmability for technically sustaining the features for controling incoming traffic. Essentially, HoneyDOC has flexible capability in flow control, but the interaction capability is redundancy under a multi-decoy architecture. So the cost is also high.

TABLE III
COMPARATION OF EXISTING HONEYPOT ARCHITECTURES IN THE LITERATURE AND HONEYVP

Architecture Type	Representative	Fidelity			Scalability			Cost	
		TCP/IP Stack Simulation	Programbility	Physics Interactions	Extensibility in Device Version	Extensibility in Device Manufacturer	Extensibility in Deployment Scope	R&D Cycle	Hardware Cost
Virtual	HoneyPHY [8]	×	×	1/2	1/2	1/2	1/2	Long	Medium
Virtual	HoneyPLC [9]	✓	✓	×	✓	1/2	1/2	Long	Medium
Physical	Siphon [10]	×	-	-	✓	✓	✓	Short	High
Hybrid	HoneyDOC [11]	×	-	-	1/2	1/2	1/2	Long	High
Hybrid	**HoneyVP**	✓	✓	✓	✓	✓	✓	Short	Medium

Keys: - = Unkown; × = No Coverage; 1/2 = Limited Coverage; ✓= Optimal Coverage

We summarise the significant advancement of our work by comparing it with the literature work in terms of the three features proposed in this work. Table III presents the comparison, which illustrates the improvement created by HoneyVP over fidelity, scalability and cost.

The main difference between proposed architecture and existing works is summarised as follow:

- In this paper, we emphasize the special requirements of ICS honeypots in three features.
- The virtual component can respond to TCP/IP Stack OS detection and basic Read SZL requests in the wild. The physical component is responsible for the interaction with sensitive operation.
- The main concern of HoneyVP is to provide a low cost design for high-interaction ICS honeypots.

VI. CONCLUSION AND OUTLOOK

Researchers are not willing spend much time and money on improving the interaction capability of ICS honeypots, or just don't want to make them public. Therefore, ICS honeypot design is suffering from the lack of a cost-effective way to attract attackers. To tackle this challenge, our work analyzed the structure of the industrial protocols and proposed a novel honeypot architecture enabling high cost efficiency. Under HoneyVP architecture, an effective combination between the virtual component and physical component can greatly reduce the cost of a flexible high-interaction ICS honeypot system. A proof-of-concept system was implemented to validate its feasibility and effectiveness. The experimental results showed the benefits of using the proposed architecture compared to previous honeypot solutions.

In subsequent studies, we would improve the interaction capability of the virtual component of HoneyVP by learning the real attack traffic and strengthen the cooperation between the virtual component and the physical component.

ACKNOWLEDGMENT

The research presented in this paper is supported by Guangdong Province Key Area R&D Program of China (Grant No.2019B010137004), National Natural Science Foundation of China (Grant No.Y710291104), Science and Technology Project of State Grid Corporation of China (No.521304190004), Industrial Internet Innovation and Development Project (No.TC190H3WU).

REFERENCES

[1] F. P. D. Rocha, "Cybersecurity analysis of a scada system under current standards, client requisites, and penetration testing," 2019.

[2] R. Bodenheim, J. Butts, S. Dunlap, and B. Mullins, "Evaluation of the ability of the shodan search engine to identify internet-facing industrial control devices," *International Journal of Critical Infrastructure Protection*, vol. 7, no. 2, pp. 114–123, 2014.

[3] M. Dodson, A. R. Beresford, and M. Vingaard, "Using global honeypot networks to detect targeted ics attacks," in *2020 12th International Conference on Cyber Conflict (CyCon)*, vol. 1300. IEEE, 2020, pp. 275–291.

[4] C. K. Ng, L. Pan, and Y. Xiang, *Honeypot Frameworks and Their Applications: A New Framework*. Springer, 2018.

[5] K. Li, J. You, H. Wen, H. Li, and L. Sun, "Collaborative intelligence analysis for industrial control systems threat profiling," in *Proceedings of the Future Technologies Conference*. Springer, 2018, pp. 94–106.

[6] S. Kalle, N. Ameen, H. Yoo, and I. Ahmed, "Clik on plcs! attacking control logic with decompilation and virtual plc," in *Binary Analysis Research (BAR) Workshop, Network and Distributed System Security Symposium (NDSS)*, 2019.

[7] H. Berger, *Automating with STEP7 in STL and SCL: programmable controllers Simatic S7-300/400*. Publicis, 2006.

[8] S. L. Litchfield, "Honeyphy: A physics-aware cps honeypot framework," Ph.D. dissertation, Georgia Institute of Technology, 2017.

[9] E. D. L. Morales, "Honeyplc: A next-generation honeypot for industrial control systems," Ph.D. dissertation, Arizona State University, 2020.

[10] J. D. Guarnizo, A. Tambe, S. S. Bunia, M. Ochoa, and Y. Elovici, "Siphon: Towards scalable high-interaction physical honeypots," in *Acm Workshop on Cyber-physical System Security*, 2017.

[11] W. Fan, Z. Du, M. Smith-Creasey, and D. Fernández, "Honeydoc: An efficient honeypot architecture enabling all-round design," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 683–697, 2019.

[12] V. Pothamsetty and M. Franz, "Scada honeynet project: Building honeypots for industrial networks," *SCADA Honeynet Project*, vol. 15, 2005.

[13] N. Provos, "Developments of the honeyd virtual honeypot," *Internet: http://honeyd. org, Accessed*, 2005.

[14] 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.

[15] C. Zhao and S. Qin, "A research for high interactive honepot based on industrial service," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 2935–2939.

[16] 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.

[17] 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.

[18] J. You, S. Lv, Y. Hao, X. Feng, M. Zhou, and L. Sun, "Characterizing internet-scale ics automated attacks through long-term honeypot data," in *International Conference on Information and Communications Security*. Springer, 2019, pp. 71–88.

[19] C. Ding, J. Zhai, and Y. Dai, "An improved ics honeypot based on snap7 and imunes," in *International Conference on Cloud Computing and Security*. Springer, 2018, pp. 303–313.