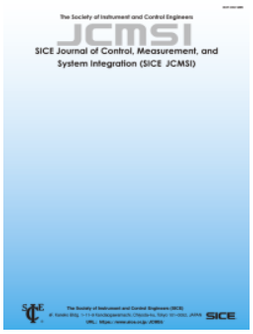

SICE Journal of Control, Measurement, and System Integration

ISSN: 1882-4889 (Print) 1884-9970 (Online) Journal homepage: www.tandfonline.com/journals/tmsi20

Developing Deception Network System with Traceback Honeypot in ICS Network

Shingo Abe, Yohei Tanaka, Yukako Uchida & Shinichi Horata

To cite this article: Shingo Abe, Yohei Tanaka, Yukako Uchida & Shinichi Horata (2018) Developing Deception Network System with Traceback Honeypot in ICS Network, SICE Journal of Control, Measurement, and System Integration, 11:4, 372-379, DOI: 10.9746/jcmsi.11.372

To link to this article: https://doi.org/10.9746/jcmsi.11.372

Published online: 18 Jan 2021.

Submit your article to this journal

Article views: 359

View related articles

View Crossmark data

Citing articles: 6 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tmsi20

SICE Journal of Control, Measurement, and System Integration, Vol. 11, No. 4, pp. 372–379, July 2018

Developing Deception Network System with Traceback Honeypot in ICS Network

Shingo ABE *, Yohei TANAKA **, Yukako UCHIDA ***, and Shinichi HORATA ****

Abstract : In industrial control system (ICS) network, communication is often conducted using custom protocols. Methods for analysis and protection from cyber threats that are specific to ICS network need to be discussed in line with each device and system specification. In this research, the honeypot technology, which is already practiced in IT networks, was further improved for ICS networks so that it responds to packets reaching the honeypots and even conducts counter-scan to collect information of the attack method and its sources. It has been already presented that machines infected with some known malware (e.g. Havex RAT) in ICS networks conduct scan activities against certain devices. For this type of attack, interaction honeypot is considered effective in identifying infected devices out of such scans. In the simulation based on Modbus Stager, which affects programmable logic controller (PLC) operation and connected PCs, the suggested interaction honeypot, namely "traceback honeypot system (THS)" successfully collected payload that is actually sent in the attacks by emulating responses to commands on Modbus protocols. Information obtained from THS-based observation can be used for proactive purposes as in separating infected devices from the operating network and restricting access to certain devices to prevent further infection in the ICS network. This paper discusses methods of tracking attack sources using the THS and preventing further infection within the network based on the search result.

Key Words : honeypot, deception network system, industrial control system, security, malware

1. Introduction

Response to cyber security incidents in industrial control system (ICS) is not significantly different from that in information technology (IT) [1]. Likewise, the importance of analysing event logs is the same in both areas regardless of the scale or cause of incidents [2]. For instance, logs recorded in PCs, servers, and other devices are used for identifying infected areas and isolating affected devices in dealing with threats on networks including malware infection, packets sent from infected devices, and devices that are remotely controlled by attackers [3]. However, such logs are not originally intended for cyber security aspects — they are rather designed to record events for the purpose of system monitoring, debugging, and troubleshooting. Therefore, even in IT systems, logging of events related to cyber security, which is required for incident response, is not always practised. Especially in ICS, devices are designed to collect logs to simply monitor operation history and behaviour as a requirement for maintaining the system, rather than the purpose of detecting security threats. As a consequence, when handling incidents, event logs which can be utilised for incident investigation may not be recorded in each device. While it is still possible to detect system failure from the limited logs, it may not be sufficient for identifying its cause. In fact, it is suggested that analysing such log trends [4] potentially presents significant hints in detecting incidents which are brought up by cyber security and other aspects. There are some suggestions on requirements and standards for event logging for incident analysis [5], however, the discussion is yet to be done especially in ICS area. In fact, ICS-CERT has pointed out several problems at users' side including insufficient log configuration for incident response [6]. This may be attributed to the fact that safe and stable operation of ICS is prioritised.

On the other hand, in the sense of detecting anomalies out of normal operations, there are means other than log analysis to observe trace of cyber security incidents. For example, by creating a whitelist on network communication to only allow actions that are predefined as normal operations, it is possible to detect and block any undefined actions. However, if any malicious actions happen to be included in the predefined, whitelisted actions or if a series of legitimate actions are leveraged, then these will not be detected properly since the rule cannot distinguish a normal and malicious action correctly. As an instance, malware can be embedded in a holding register or its device configuration can be altered by leveraging legitimate Modbus protocols or a device specification [7]. In this case, whitelisting can restrict commands to be executed, however, it is not practical to restrict input values since normal range of input values for each command need to be examined beforehand. This issue implies that protection through whitelisting is not practical for detection or as a countermeasure.

In this sense, as a means to detect threats in ICS network, technologies applying honeypots have been proposed [8]. A honeypot is a system imitating a vulnerable system, which attracts attackers and bypasses possible attacks against devices in use. It enables collecting logs of malicious activities and

* ICS Security Response Group, JPCERT Coordination Center (JPCERT/CC), Hirose bldg. 11F Kanda-Nishiki-Cho 3-17, Chiyoda, Tokyo 101-0054, Japan
** Analysis Center, JPCERT/CC, Japan
*** Global Coordination Division, JPCERT/CC, Japan
**** Watch and Warning Group, JPCERT/CC, Japan
E-mail: icsr@jpcert.or.jp, aa-info@jpcert.or.jp, global-cc@jpcert.or.jp, ww-info@jpcert.or.jp
(Received November 1, 2017)
(Revised January 7, 2018)

analysing the attack method based on the logs. This technique has been commonly practised in IT [9], and also some that are disguised as Internet of things (IoT) devices have been used for searching for malware such as Mirai [10]. Generally, honeypots are directly connected to the Internet and used for observing attack methods through receiving attacks. In detecting threats in ICS networks [8], it can also be connected to networks which have control devices (e.g. programmable logic controllers (PLCs)) to detect lateral movement. Even if ICS devices itself within the ICS network do not have adequate logging configuration, it is possible to track the attack source based on the logs obtained by the honeypot.

In fact, devices infected with Havex RAT [11], a type of malware used for sophisticated cyber attacks against ICS networks, conduct network scans to search for certain products. In lateral movement observed in IT, attackers expand malware infection by leveraging authentication systems that are commonly applied in enterprise IT systems to gain access to confidential files and conduct other activities in pursuit of their achievement. In ICS, however, infected device activities and attack purposes need to be analysed from a different viewpoint since authentication systems on ICS networks are different from those in IT. Some research has been already suggesting that malware that spreads infection through PLCs is considered as a realistic threat as well as ransomware attack against ICS [12],[13]. In order for malware to find a next target to infect, network scans can be conducted for checking the network structure and connected devices. In a situation where an ICS network is compromised resulting in information theft or malware infection within the network, it is suggested that anomalies can be detected by using a honeypot disguised as an ICS device.

Given the discussion, JPCERT/CC developed a honeypot which detects scans out of obtained logs and analyses the source host and attacking packets [14]. The system was also tested to identify attack sources, based on attack detection method in ICS networks using honeypot [8]. The details of the honeypot will be described in section 2. JPCERT/CC also evaluated the honeypot's function to detect various threats on ICS networks by using Havex RAT and other malware (section 3). As an example of attacker's reconnaissance activities by leveraging existing protocols and commands, attacks leveraging Modbus Stager was simulated to evaluate the traceback honeypot system (THS)'s behaviour. Even if an anomaly is detected based on the information collected by the honeypot, the system would not be practical without any means to notify the system admin. From this perspective, additional functions to the detection system will also be discussed in section 3. Section 4 presents the methods to collect information of the attack source using the honeypot and observations gained through the analysis.

2. Traceback Honeypot System (THS) on ICS

Honeypots in general classify received packets by protocol or port and send a response that is pre-set to each packet. The contents of a response would be either sent directly from the actual application in use, or honeypots can imitate such a response. Honeypots enable observing attack activities in a continuous manner by disguising as a device or server in use and collect information on attack methods through recording access logs from attackers. However, if the response from the honeypot is not provided or not corresponding to the packets sent from attackers, they may be able to realise that the device is a honeypot. Under the circumstances, attackers are likely to stop attacking the system, and attack observation could not be continued. It is necessary that honeypots are designed to deceive attackers and disguise as an ordinary device or server in use. Honeypots have been applied in analysis of threats against ICS devices. There was a research using honeypots that were configured to receive access to telnet (23/tcp) and secure shell (22/tcp) and respond by emulating IoT device console access, which detected some malware including Mirai [10]. Another research collected device-specific information contained in the response from devices in ICS networks (e.g. product name, model number, vendor information) and integrated them in the honeypot's response, which successfully observed some attacker activities against ICS networks [8].

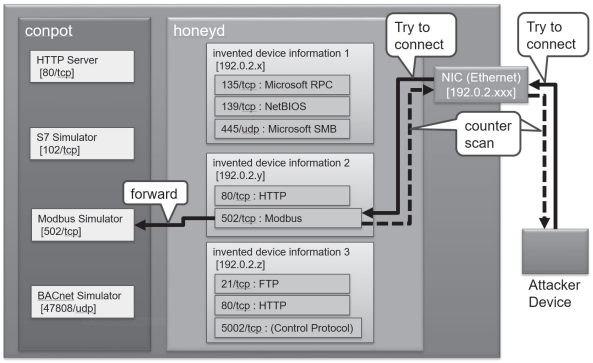

Fig. 1 Software design of traceback honeypot system.

JPCERT/CC's system implemented function to check the source of scan packets and identify the attackers as well as to send the device-specific information for attack detection in ICS networks. The basis of the honeypot is "honeyd" [15], which has two components: one to receive packets and classify them into specified actions and another to actually perform an action. As a response to a received scan packet, the latter component conducts scans to the source as well as sending device-specific information (Fig. 1). This feature enables identifying the source of the scan packet. This is considered effective in identifying devices that have suspicious behaviour or that are used as a steppingstone for other attacks. ICS network structure rarely changes during operation except when replacing failed devices or adding machines such as human machine interfaces (HMIs). When there are network scan packets originated in devices connected to ICS network, it implies the possibility of failure in security configuration which allows unintended operation by the admin or cyber security incidents as to the device being leveraged as a steppingstone. In this sense, identifying the source of scan packets is the crucial part in incident analysis. It would also be difficult for attackers to recognise the fact that they are also being scanned. From this perspective, the THS is considered as an effective method for information collection as a counteraction against attackers.

Nevertheless, there still is a possibility that attackers realise the existence of honeypot since the above function does not cover responses to packets which are sent for purposes other than scanning. In order to enhance the response function of the system, the suggested system integrated "conpot" [16] into the honeypot, which imitates responses to main protocols used in

ICS (Table 1) as well as providing device-specific information. This is software dedicated for ICS networks among other honeypots and has been used quite widely in this area. For example, it was implemented to imitate a system for a gas tank distributed abroad and observed attacker activities [17]. It was also used in an experiment to observe whether such imitative devices can be registered to search engines for Internet-connected devices such as SHODAN [18]. Compared to the previous implementation which sends device-specific information only, this technique enables collecting more information since it replies to various commands of ICS protocols. The THS was tested using the ICS network described in Fig. 2 to verify if it is capable of providing expected replies to requests without affecting other devices. The results confirmed that the newly developed honeypot only responds to communication sent to itself and is able to operate corresponding to packet source (Fig. 3). This suggests that it would not affect other devices on the network during normal operation. For this counter-scan function, NMAP, a common scanning tool, is implemented in the THS. As a defence measure for ICS, some researchers suggest techniques to deceive attackers connecting to the honeypot and lead to a decoy network (separated from the network for ICS operation) to securely isolate the attack traffic [19]. By improving the response function of the honeypot, attackers may be deceived and continue attacks for a longer period of time, which allows collecting more information to identify communication from attackers. Such observation can be used to enhance the accuracy of the above isolation technique.

Table 1 ICS-related protocols implemented in conpot.

Port Number	Protocol
502/tcp	Modbus
102/tcp	S7 Comm
80/tcp	HTTP
161/udp	SNMP
47808/udp	BACNet
623/udp	IPMI
10001/tcp	automated-tank-gauge

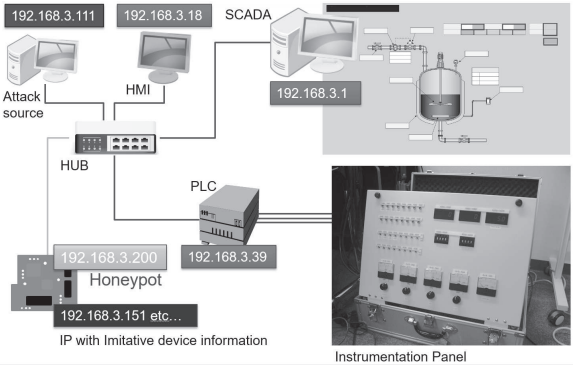

Fig. 2 Current test environment (single segment 192.168.3.0/24).

```
Starting Nmap 6.47 ( http://nmap.org ) at 2017-03-06 15:44 JST
Initiating ARP Ping Scan at 15:44
Scanning 192.168.3.111 [1 port]
Completed ARP Ping Scan at 15:44, 0.24s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 15:44
Scanning 192.168.3.111 [1000 ports]
Discovered open port 445/tcp on 192.168.3.111
Discovered open port 135/tcp on 192.168.3.111
Discovered open port 139/tcp on 192.168.3.111
Discovered open port 49157/tcp on 192.168.3.111
Discovered open port 49152/tcp on 192.168.3.111
Discovered open port 49156/tcp on 192.168.3.111
Discovered open port 49155/tcp on 192.168.3.111
Discovered open port 49154/tcp on 192.168.3.111
Discovered open port 49153/tcp on 192.168.3.111
Increasing send delay for 192.168.3.111 from 0 to 5 due to 245 out of 815 dropped probes since last increase.
Increasing send delay for 192.168.3.111 from 5 to 10 due to 11 out of 35 dropped probes since last increase.
Increasing send delay for 192.168.3.111 from 10 to 20 due to 11 out of 32 dropped probes since last increase.
Increasing send delay for 192.168.3.111 from 20 to 40 due to 11 out of 30 dropped probes since last increase.
Increasing send delay for 192.168.3.111 from 40 to 80 due to 11 out of 31 dropped probes since last increase.
Completed SYN Stealth Scan at 15:45, 28.23s elapsed (1000 total ports)
Nmap scan report for 192.168.3.111
Host is up (0.00056s latency).
Scanned at 2017-03-06 15:44:34 JST for 28s
Not shown: 991 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:0C:29:7C:F0:6C (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 29.03 seconds
           Raw packets sent: 1369 (60.220KB) | Rcvd: 1261 (50.608KB)
```

Fig. 3 Results of counter-scan by the THS (THS automatically conducts counter-scan to the scan source (192.168.3.111) by NMAP).

3. Evaluation of the THS on ICS

JPCERT/CC evaluated the THS's behaviour in two different attack cases within ICS networks 1) originating in Havex RAT malware infection and 2) caused by worms infecting PLCs. As opposed to JPCERT/CC's previous research on threats against Internet-reachable ICS [20], this research focuses on an ICS network which is compromised by attackers and exposed to the risks posed by attack activities.

3.1 Case 1: Evaluation with Havex RAT

Havex RAT [11] is a type of malware which was reportedly used in sophisticated cyber attacks against ICS around 2013 [21]. The overview of malware infection flow is as follows; 1) Attackers compromise a website of a company which provides ICS software and embedded the malware in the software distributed on the website. 2) When its constituents download and install the software, the device gets infected with Havex RAT while legitimate software operates (Fig. 4). In-

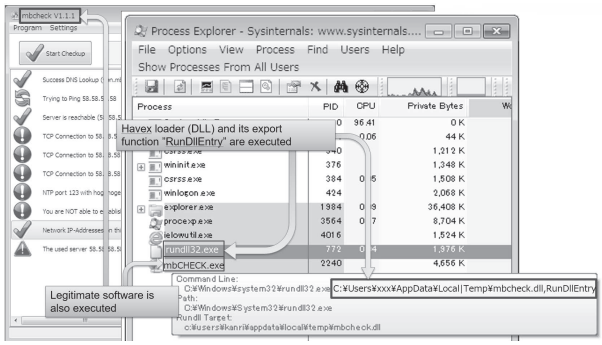

Fig. 4 Havex RAT (Havex RAT is executed while legitimate software operates).

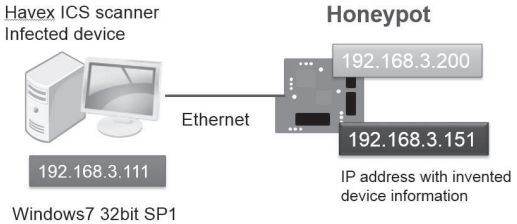

Fig. 5 Test environment.

Table 2 Scan target ports of Havex RAT.

Port Number	Protocol
44818/tcp	EtherNet/IP, Rockwell Rslinx
502/tcp	Modbus, Modicon PLC
102/tcp	Siemens PLC
11234/tcp	Measuresoft ScadaPro
12401/tcp	7-Technologies IGSS SCADA

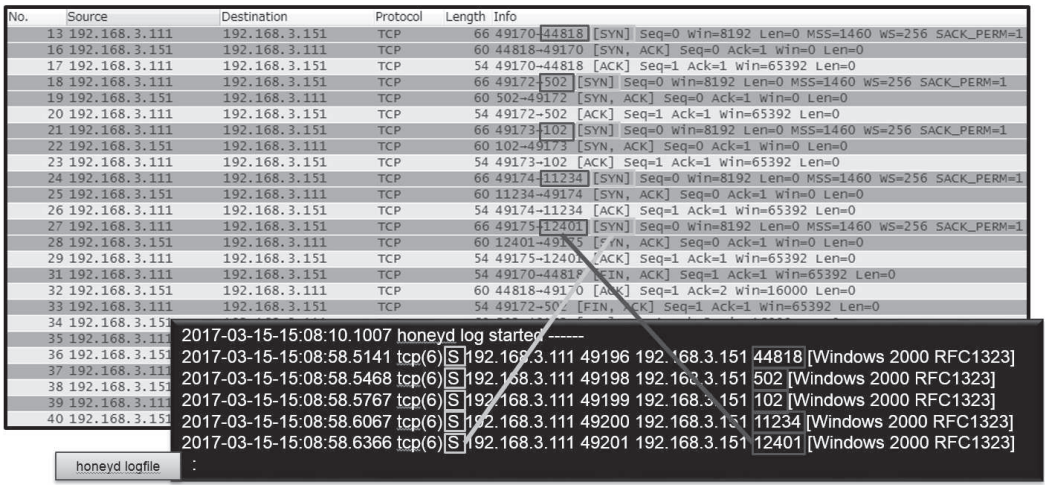

Fig. 6 Observation result for Havex RAT. Infected device (192.168.3.111) sends packets to THS (192.168.3.151) scanning ICS well-known ports. These packets are captured at THS and its consequences are recorded in honeyd logfile.

fection can also spread to PCs without Internet connection if the malware-embedded software is distributed through a USB memory or maintenance device. On top of the function as a remote access tool (RAT) controlling infected devices through command and control servers, Havex RAT has a plugin function to conduct scans against certain ports in the infected networks to search for object linking and embedding for process control (OPC) servers. This paper focuses on the scanning function of Havex RAT. Table 2 shows the scan targets of Havex RAT. These ports are commonly used in ICS. There are also characteristics in the selection of target IP addresses.

JPCERT/CC evaluated the THS's behaviour assuming an ICS network infected with Havex RAT. The malware only conducts scans against devices and checks for connection, and it does not have any function to execute commands on devices that are identified. Based on the protocols in Table 1, some devices which respond to such protocols were selected and its device-specific information was created so that it is integrated in the response to scans. A PC infected with Havex RAT was connected to the network implementing a honeypot so that it is exposed to scans (Fig. 5). Figure 6 describes the results of the communication captured between the infected PC and the honeypot when the scan was conducted.

The trace of scans is recorded both in the honeypot and the infected PC. Havex RAT records the scan results in a file as in Fig. 7. The malware leaves a record that there is some target information to steal within the ICS network, and imitative ICS information is sent from the honeypot.

The honeypot records the scan from the infected PC in the log, and in return conducts scans back to the device to collect its information, including a list of OS that the infected PC may be using and its open port, which helps in identifying the attack vector in the ICS network. However, not all types of malware targeting ICS have reconnaissance functions in the network with an aim of lateral movement. Although the THS is not effective in detecting attacks that does not send scan packets or affect other devices (e.g. key logger attack or data destruction), it is capable of collecting information in case of attacks involving scans against the network and worm infection spreading to devices within a network.

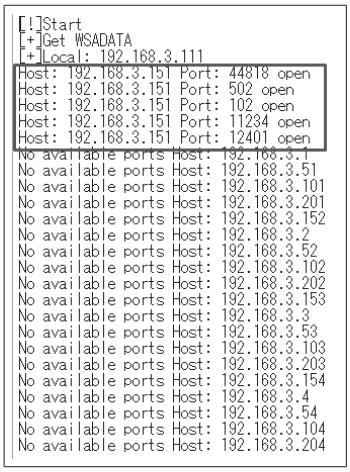

Fig. 7 Havex RAT's records on scan (the honeypot's address is recorded).

3.2 Case 2: Evaluation with Attacks Targeting PLCs

Concepts of malware targeting PLCs which spreads infection within ICS networks have been already discussed [7],[12]. Attacks where PLC programs get encrypted and its admin is demanded for money in exchange for the data (so called "ransomware attack") [13] also take into account that malware infection spreads through PLCs. Malware infection in ICS devices has been recognised as a threat, and there have been some experimental studies being conducted. If the THS is implemented, there is a possibility that such malware activities are detected, as verified in the previous subsection. This subsection verifies the THS capability in collecting information about infection activities referring to two research concepts involving PLCs: PLC blaster [12] and Modbus Stager [7]. For observing such ICS-specific attacks, sending device specific information as a response is not sophisticated enough to deceive attackers into attacking the device. In order to observe various attack activities as long as possible, the THS needs to be compatible with multiple commands that are potentially used for attacks. In this verification, conpot was configured to emulate responses according to the specifications of Modbus protocol and PLC blaster to confirm that THS's is practical in observing such at-

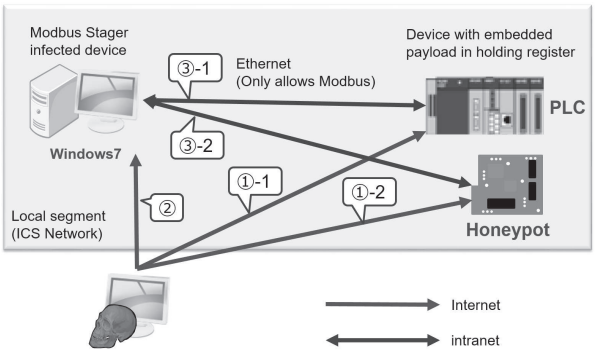

Fig. 8 Modbus Stager Infection Scenario. (①-1) Attacker writes payload by Modbus (*1) protocol in the PLC's Holding Register. (①-2) Previous step (①-1) is also done on services that a honeypot imitates. (②) Modbus Stager is sent to Windows 7 on the device and infection occurs. (③-1) When Modbus Stager is executed, the embedded payload in the PLC's Holding Register is obtained by Modbus protocol (*2) and executed. (③-2) Previous step (③-1) is also done on services that a honeypot imitates

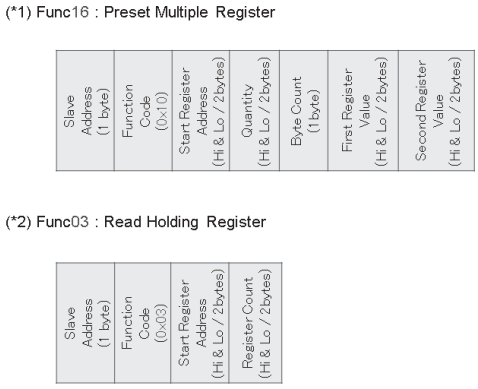

Fig. 9 Modbus commands used in Modbus Stager.

tacks.

One factor to observe is a worm that spread infection among PLCs. Based on a programming specification of a particular PLC, PLC blaster research presents codes to spread worm infection, which can be added to a legitimate program, and rewrites programs stored in the PLC [12]. Since the reporter has not disclosed the Proof-of-Concept (PoC) code in detail, the commands used for this exploit is unknown. However, in case of popular PLCs, programming specifications are sometimes available on its manuals, and some researchers disclose scripts to execute commands without engineering tools. Various commands can be analysed through capturing communication between an engineering tool and a PLC. Based on the analysis results, it is possible to implement functions to rewrite programs without engineering tools or conduct scans against other PLCs within the network. If an infected PLC operates as a worm, it sends commands to other PLCs within the ICS network. If the THS is implemented, it is expected that activities of such PLCs operating as a worm can be revealed by observing commands sent within the network. Especially when the infection is widespread to multiple PLCs, similar commands may be sent simultaneously from several PLCs, and this may increase the possibility to detect anomalies in the ICS network.

On the other hand, the reporter of Modbus Stager focused on the holding register on Modbus protocol [7]. Depending on the model of PLCs, there are read/writable areas in the holding register. Usually, the area is used to store configuration and other information, however, a failure has been discovered there which allows malware to be embedded by attackers instead. When devices such as HMIs access the PLC and read the certain area, they will be infected with the embedded malware. Although it is a PoC for a specific product, the reporter provided an explanation around the observation that malware infection can occur solely by the legitimate specification of Modbus protocol. In fact, since the address of the writable area in the holding register is different in each model or vendor and many different types of PLCs are distributed, the whole holding register itself needs to be explorered. In order to further identify the read/writable area and enhance the success rate, attackers are likely to attempt executing Modbus commands repeatedly. Honeypots are expected to capture such attack activities through observing scans against devices and IP addresses as a first step to an attack. Furthermore, attackers need to access PLCs to infect HMIs and other devices with malware when leveraging Modbus Stager. Analysing such access logs recorded in the devices can also be useful in observing trace of attacks. Figure 8 summarises the points to consider in detection of attacks similar to techniques leveraging Modbus Stager. In detecting this activity, the THS needs to be configured to respond to Modbus protocol and commands including reading of the holding register. To be precise, it needs to be compatible with the commands listed in the PoC for accessing PLCs (Fig. 9).

To confirm the THS's effectiveness for anomaly detection, some analysis was conducted based on the attack scenario as in Fig. 8. For this purpose, conpot was further improved to emulate PLC's responses when its holding register is accessed using Modbus protocol so that the payload from attackers can be collected. Since the conpot's address of read/writable area in the holding register is configured identically to the actual PLC placed in the network, attackers would observe exactly the same behaviour when sending queries to check whether the holding register is accessible. As shown in Fig. 10, there is no difference between the response sent from the THS and the PLC. Furthermore, the THS also responds to read/write commands in Modbus protocol while collecting malicious payload sent from attackers. This way, it is possible that attackers continue attacking the THS as well without noticing. As a result, the whole process of writing malicious program into the holding register can be observed using the THS. The verification simulated an attack where an attacker compromises the THS (disguised as a PLC) and attempts to write a test program in the writable area of its holding register. The result has shown that a malicious payload was detected in the THS (Fig. 11). Since the commands contained in the PoC are legitimate commands which can be used in normal operation, it is difficult to detect the case as an anomaly. However, the logs obtained by THS can be a clue to identify the malicious activity. After overwriting the holding register, it is likely that an attacker would attempt to overwrite the payload in order to erase the evidence. For this case, it is also possible to examine the entire activity through THS's logs. As the THS deceives attackers and emulates responses as a real PLC, it can observe payload sent from the attackers. This enhanced function to record attack procedures in detail and for the extended area is the significant difference from the preceding study [8].

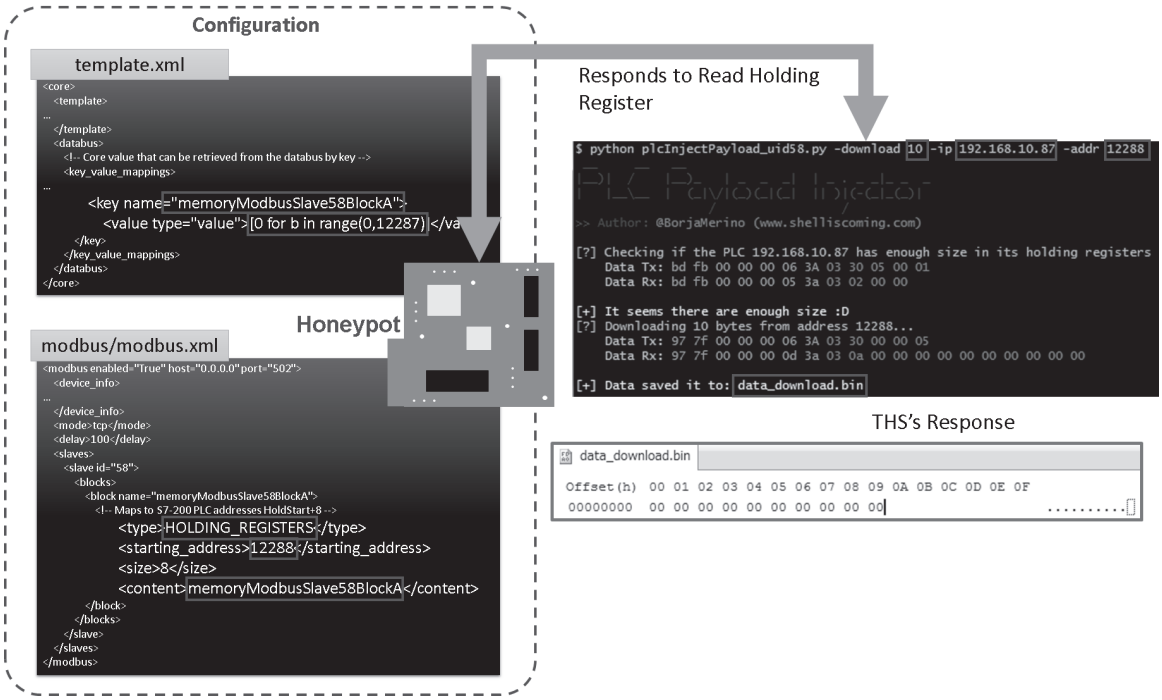

Fig. 10 Implementation and observation of the THS in Modbus Stager.

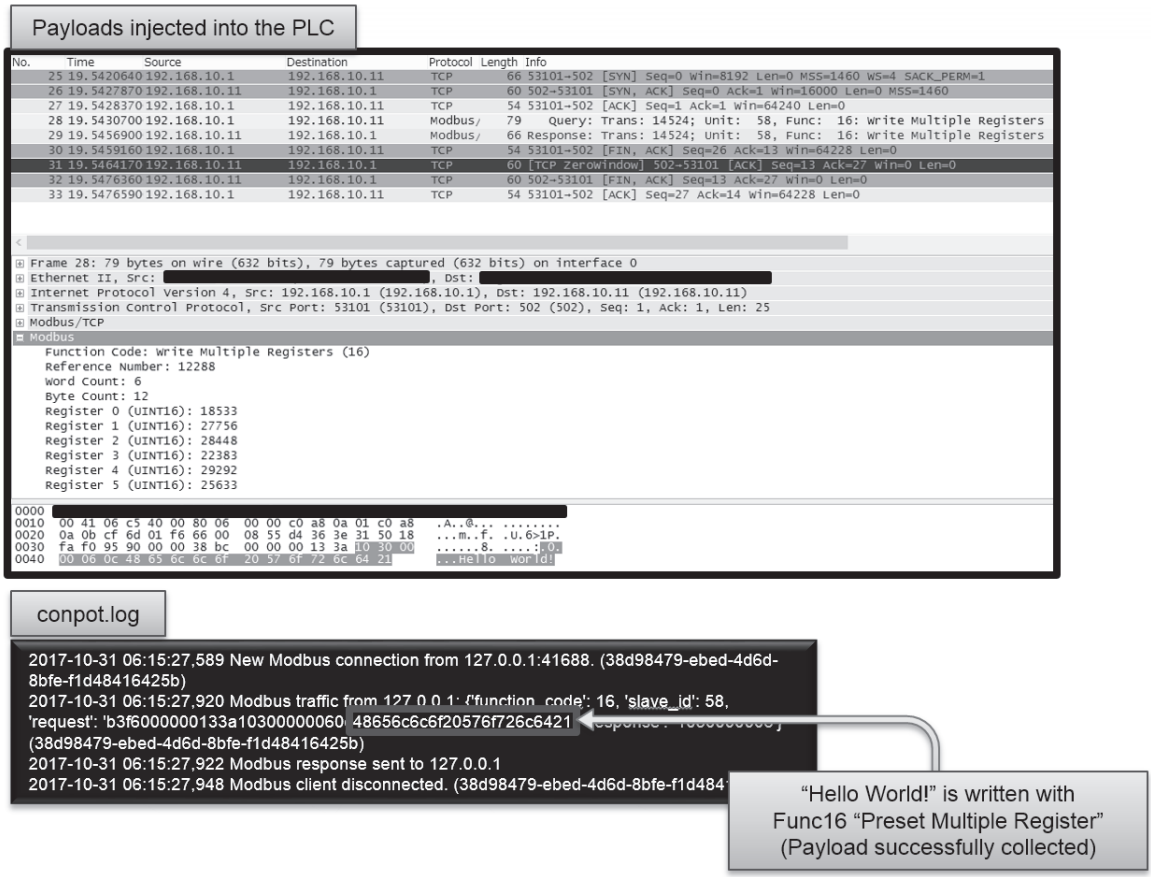

Fig. 11 The THS's observation of the payload written in the PLC.

As the two examples have demonstrated, the THS is more heavily focused on response to commands compared to conventional honeypots. This offers an advantage that it can gain more detailed information of the attack and its method.

3.3 Applying the THS for Deception Network System

By combining the method presented in the preceding study [8] and aforementioned implementation, the THS was tested in the network constructed as in Fig. 2, assuming that there is a malware infected device on the network. When the

infected device conducts scans in an attempt to see the network structure, the THS makes changes to the network structure by placing dummy devices on the network as in the existing method [8]. From the attacker's point of view, the network scale looks larger than it actually is. In addition, the THS conducts counter-scans to the source host as explained in section 3.1 so that detailed information of the source host can be obtained such as device type, OS, open ports etc. If any command is sent to the THS using an ICS-specific protocol, the THS emulates a response according to the customisation as in section 3.2 and pretends to be a real ICS device. Payload that is collected through the communication would be a clue in identifying the attack method.

Therefore, it is suggested that in the network structure as in Fig. 2, the THS is capable of combining the following features: 1) attracting attackers by imitating a larger scale network with ICS devices, 2) performing counter-scans to obtain a profile of the source device, 3) deceiving attackers into compromising the THS for a longer period of time with the enhanced response contents, and 4) collecting payload sent from the attackers and use them as a clue to analyse the attack method. Consequently, with the THS, it is suggested that a "deception network system", can be formed which attracts attackers as an entire network, not just by an individual device. It is also possible to identify infected devices and implement security measures while attackers are preparing to attack actual ICS devices. Previously, it was generally difficult to recognise irregular behaviours on ICS networks, however, the THS offers an opportunity to catch such indication.

4. Conclusion and Discussions

As a mechanism to detect attack activities within ICS networks, JPCERT/CC examined a system which was proposed in some preceding studies [8],[19] and extended some functions to detect some realistic threats. Attack scenarios against ICS as discussed in PLC blaster [12] and Modbus Stager [7] legitimate protocols and commands used in ICS devices. In order to detect such suggested attack techniques, it is required that honeypots have more elaborate functions to emulate an ICS device on top of sending device-specific information. The THS's significant difference from the honeypot concept in the previous research [8] is that it has a function to respond to commands as well as returning device specific information. In the preceding study, in order to attract attackers, multiple honeypots were distributed across the network and change the network structure dynamically upon receiving scans. However, in case of attacks leveraging devices and command specifications that are specific to ICS environment such as Modbus Stager [7], since the honeypot cannot respond accordingly, attackers would identify it as a honeypot. In the verification for this paper, the THS's feature was enhanced by emulating PLC's responses to commands for scanning the holding register and actually writing a program. With this update, it is likely that attackers will continue the malicious activities to the THS without noticing the difference between the actual device and the THS. By improving the honeypot's response to attack packets, it can continue to deceive attackers for a longer period of time. Information that is accumulated through the observation can help identifying the characteristics of communication from the attackers, and this can be useful in enhance network isolation [19].

Some security vendors have already proposed deception network systems for IT systems and ICS. These systems are aimed to deceive attackers and prevent their access to legitimate devices by bypassing attack traffic to a honeypot itself. However, that system would not operate effectively against attacks unless it is designed and configured in accordance with each attack scenario and command in order to provide accurate responses. In IT systems, popular systems and applications tend to be targeted for attacks, and it is relatively easy to assume attack targets on the network. Yet, in ICS, a wide range of systems, applications, devices, network structure and configurations are in place at different organisations, which makes it difficult to narrow down possible attack targets. For this reason, detection criteria and the range of imitation in response need to be customised depending on the attack scenario. This difference is one of the most concerning challenges for a deception network system in ICS. On the other hand, as for devices that are likely to be attacked, OPC servers and PLCs can be suggested. In Havex RAT's instance, one of the malware's features is to scan OPC servers. Modbus Stager's case proved that infection can be carried out via a Modbus-enabled PLC in the network, while it was indicated that worm infection spreads to PLCs on the network in PLC blaster. In this paper, as a starting point, a deception network system was developed by focusing on some specific protocols and commands. With a limited range of devices to consider, it is possible to concentrate on some possible attack techniques, which is helpful in preparing security measures and initial incident response.

In creating a generic deception network system, it is important to trace the attack source as well as responding to commands from incident handing perspectives. The other distinctive feature of the THS from previous research [8] is that counter-scanning function was implemented. As seen in the Havex RAT case, early detection of infected devices helps shortening the time for initial incident handling. Besides scanning the source device, in order to thoroughly analyse the attack including its target and technique, detailed information can be obtained by logging into actually affected devices. This will allow access to information on systems and services that operate on the devices and also possibly collect malware itself which may be left there.

Lastly, such collected information can be useful in preventing further attacks when shared with related organisations. Exchange of information for early detection of cyber threats has been commonly practised in IT systems, however, there has not been any example specialised in ICS. Establishing such information sharing scheme needs to be considered to effectively implement security measures and enhance the security level of the industry as a whole. Further discussion and improvement needs to be done to develop an ICS threat sharing framework and generic deception network systems based on the THS.

References

[1] ICS-CERT: Recommended practices, ICS-CERT, https://ics-cert.us-cert.gov/Recommended-Practices.
[2] K. Kent and M. Souppaya: Guide to computer security log management, NIST Special Publication 800-92, NIST, 2006, http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf.
[3] JPCERT/CC: Use of logs and analysis methods for handling sophisticated cyber attacks, JPCERT/CC, 2015, https://www.

jpcert.or.jp/research/APT-loganalysis_Report_20161019.pdf.
[4] Y. Tajima, T. Yamagata, H. Yamamoto, and A. Shimura: Anomaly detection based on event prediction for control system, *The 29th Annual Conference of the Japanese Society for Artificial Intelligence*, 2015.
[5] K. Stouffer, J. Falco, and K. Scarfone: Guide to industrial control systems (ICS) security, NIST Special Publication 800-82, NIST, 2013, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82.pdf.
[6] ICS-CERT: ICS-CERT monitor March 2012, ICS-CERT, 2012, https://ics-cert.us-cert.gov/sites/default/files/Monitors/ICS-CERT_Monitor_Mar2012.pdf.
[7] B. Merino: Modbus Stager: Using PLCs as a payload/shellcode distribution system, 2016, http://www.shelliscoming.com/2016/12/modbus-stager-using-plcs-as.html.
[8] H. Naruoka, M. Matsuta, W. Machii, T. Aoyama, M. Koike, I. Koshijima, and Y. Hashimoto: ICS honeypot system (CamouflageNet) based on attacker's human factors, *Procedia Manufacturing*, Vol. 3, Elsevier, 2015.
[9] N. Provos and T. Holz: *Virtual Honeypots: From Botnet Tracking to Intrusion Detection 1st Edition*, Addison-Wesley Profession, 2007.
[10] Y.M. Pa Pa, S. Suzuki, K. Yoshioka, and T. Matsumoto: IoTPOT: Analysing the rise of IoT compromises, *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, 2015.
[11] D. Hentunen and A. Tikkanen: Havex hunts for ICS/SCADA systems, F-Secure, 2014, https://www.f-secure.com/weblog/archives/00002718.html.
[12] R. Spenneberg, M. Brüggemann, and H. Schwartke: PLC-Blaster: A worm living solely in the PLC, RSA Conference Asia, 2016, https://www.blackhat.com/docs/asia-16/materials/asia-16-Spenneberg-PLC-Blaster-A-Worm-Living-Solely-In-The-PLC-wp.pdf.
[13] D. Formby, S. Durbha, and R. Beyah: Out of control: Ransomware for industrial control systems, RSA Conference USA, 2017, http://www.cap.gatech.edu/plcransomware.pdf.
[14] S. Abe, Y. Tanaka, Y. Uchida, and S. Horata: Track attack sources based on traceback honeypot for ICS network, *Proceedings of the SICE Annual Conference 2017*, pp. 712–723, 2017.
[15] N. Provos: A virtual honeypot framework, USENIX Association, 2004, http://www.citi.umich.edu/u/provos/papers/honeyd.pdf.
[16] The Conpot development team: http://conpot.org.
[17] K. Wilhoit and S. Hilt: The GasPot experiment: Unexamined perils in using Gas-tank-monitoring systems, Trend Micro, 2015, https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp_the_gaspot_experiment.pdf.
[18] Shodan: https://www.shodan.io.
[19] W. Machii, I. Kato, M. Koike, M. Matta, T. Aoyama, H. Naruoka, I. Koshijima, and Y. Hashimoto: Dynamic zoning based on situational activities for ICS security, *Proceeings of the 10th Asian Control Conference 2015*, pp. 1242–1246, 2015.
[20] S. Abe, M. Fujimoto, S. Horata, Y. Uchida, and T. Mitsunaga: Security threats of internet-reachable ICS, *Proceedings of the SICE Annual Conference 2016*, pp. 750–755, 2016.
[21] Kaspersky Lab Global Research and Analysis Team: Energetic bear: Crouching Yeti, Kaspersky, 2014, https://securelist.com/files/2014/07/EB-YetiJuly2014-Public.pdf.

Shingo Abe

He received the M.Sc. degree from Graduate School of Science and Technology Department of Mathematics, Tokyo University of Science in 2008. He joined Toshiba Solutions Corporation in 2008. He has been a member of ICS Security Response Group of JPCERT/CC since 2014 as Information Security Analyst, especially in the area of IoT. He is currently a Ph.D. student at the Graduate School of Engineering at Nagoya Institute of Technology.

Yohei Tanaka

He graduated from Faculty of Science, Yokohama City University. He was previously engaged in programming and system development at a vehicle equipment manufacturer. Since 2012, he has been a member of the Analysis Center at JPCERT/CC, in charge of malware analysis and incident response.

Yukako Uchida

She graduated from the School of International Liberal Studies, Waseda University in 2011. She joined Global Coordination Division of JPCERT/CC as Liaison Officer in 2013 and has been engaged in CSIRT communities mainly in the Asia Pacific region, and information sharing in English through JPCERT/CC's blog and others.

Shinichi Horata

He received the Ph.D. degree from the School of Mathematical and Physical Science, the Graduate University for Advanced Studies (SOKENDAI) in 2003, majoring in theoretical physics and mathematical models. He became Assistant Professor at Hayama Information and Network Center of the Graduate University for Advanced Studies in 2006, and Lecture of the Center for Academic Information Services in 2013. He joined Watch and Warning Group, JPCERT/CC as Information Security Analyst in 2015, and has been working as Manager of the Group since 2016.