# On Design and Enhancement of Smart Grid Honeypot System for Practical Collection of Threat Intelligence

Daisuke Mashima, Derek Kok, Wei Lin
*Illinois at Singapore Pte Ltd*

Muhammad Hazwan, Alvin Cheng
*Custodio Technologies Pte Ltd*

## Abstract

The smart grid system is exposed to cyberattacks, as demonstrated by the number of real-world incidents in the last few years. The attack strategies keep evolving, and security mechanisms must identify novel attack vectors ideally before they actually hit the system. In this direction, honeypot systems for smart grid infrastructure are considered effective. While use of honeypot systems for general IT security has a history already, implementations for smart grid systems, and industrial control systems in general, are not mature yet. In this paper, we summarize our efforts for designing, implementing, and evaluating our smart grid honeypot system. We started with a prototype implementation of the virtual smart grid infrastructure using open-source tools, evaluate the realism of it from an attacker's perspective through collaboration with cybersecurity experts. We then refined the honeypot system to offer better realism as well as logging features for capture attackers' behaviours.

## 1 Introduction

While a honeypot is an effective cybersecurity tool for misleading attackers to delay attacks and collecting threat intelligence based on captured real-world attack attempts, design and implementation for smart grid systems is still in nascent stage. In general a honeypot is relatively easy to fingerprint by using typical cybersecurity tools such as Nmap [1]. In addition, honeypot systems for cyber-physical systems, including smart grid systems, suffer from identification based on physical system behaviors. For instance, if an attacker has good knowledge about power systems, he may notice inconsistencies between the estimated power grid status and measurements conveyed in SCADA (supervisory control and data acquisition) messages. Only recently, a smart grid honeypot incorporating real-time power flow simulation to address challenges in cyber-physical inconsistencies has been proposed [2].

However, to our knowledge, how realistic the honeypot implementation looks to cyber attackers have not been intensively studied. Thus, in this work, we construct an in-house implementation of a hypothetical, but realistic smart grid communication infrastructure, which consists of a control center and a field substation, which are compliant to widely-used international standards such as IEC 60870-5-104 and IEC 61850 [3], and evaluate the setup from attackers' point of view, in order to learn what would hint attackers to allow them to identify whether the system is real or not. To conduct this study, we partner with cybersecurity experts to conduct penetration testing according to a realistic attack scenario. Such findings will be meaningful to decide the directions for the honeypot enhancement. We then design the improved version of the smart grid honeypot system based on the findings from the penetration testing experiment as well as logging features to capture the experimented attack scenarios.

The rest of the paper is organized as follows. In Section 2, we discuss related work, including existing smart grid honeypot systems. Then, we move on to the discussion of the design and implementation of the preliminary smart grid honeypot system, which is seen as the baseline implementation that many researchers would likely start with, in Section 3. Section 4 elaborates our strategy for evaluating the honeypot from an attacker's perspective, followed by the summary of findings from the study. We discuss our strategy for improving honeypot system in Section 5 and 6, and Section 7 presents the preliminary evaluation of the improved smart grid honeypot. Finally, we conclude the paper with future directions for the smart grid honeypot implementation in Section 8.

## 2 Related Work

There exist open-source honeypot implementation for industrial control systems [4–8]. Among them, Conpot [7] is an open-source, low-interaction honeypot designed for industrial control systems (ICS) and is actively maintained. Conpot supports several Internet and ICS-specific protocols such as Modbus. However, it does not offer anything at the physical side. Another limitation is that it is relatively easy to get fingerprinted once attackers get shell access to the honeypot device,

for example by seeking its python process in the process list on the machine. Even based only on network characteristics, Conpot can be detected relatively easily [9].

In the academia, a number of efforts are devoted for implementing honeypot system for smart grid systems. The many of prior efforts, including CryPLH [10] and SHaPe [11], only imitate the cyber side, and thus, not enough to deceive power-system-aware attackers. Recently a honeypot system that emulates internal details (e.g., communication protocols supported, network topology etc.) as well as provides cyber-physical integrated view of the system by means of integrated power-flow simulation was proposed [2]. However, its scope is limited to a substation system and only incorporates minimal types of devices, namely IEDs and substation gateway.

Shodan [12], a search engine for Internet-connected devices, has the capability to identify whether each indexed device is deemed as a honeypot or not [13]. The detailed logic is not published; it is likely that they rely on characteristics of popular honeypots, including those mentioned earlier [14]. Therefore, it is effective only for known implementations. Moreover, our honeypot exposes to the Internet only a dummy, but real, VPN service as the entry point for attackers, and thus is not flagged as honeypot by Shodan.

To our knowledge, there is no extensive study in evaluating realism of honeypot implementations from an attacker's perspective in the cyber-physical systems domain. In the rest of this paper, we tackle this problem by presenting the design of virtual smart grid communication infrastructure and use it for finding what characteristics would potentially hint attackers to detect honeypot systems. While we particularly focus on smart grid, similar approaches can be tailored to honeyhpot systems for other types of cyber-physical systems.

# 3 Testbedding Smart Grid Communication Infrastructure

## 3.1 Typical Setup and Configuration

At a high level, as illustrated in [2], the smart grid consists of a control center and multiple (perhaps thousands of) field substations at the transmission level and distribution level connected via wide-area network (WAN), which may be a dedicated, private network, or virtual private network (VPN) on public network (e.g., the Internet), and communication medium may be wired (e.g., fibre optic cables, power-line communication) or wireless (cellular or mesh).

The control center hosts a number of components, and the representative components include a SCADA master HMI (human machine interface), historian (or some sorts of database), and a VPN server so that the control center system can be accessed remotely or from PCs in the enterprise IT network of the grid operator or device/system vendors.

On the other hand, a field substation system consists of a different set of devices that are more specific to the power grid control. In this paper, we focus on a modernized substation that is compliant to international standard protocols, such as IEC 60870-5-104 and IEC 61850. As illustrated in a reference model published by International Electrotechnical Commission (IEC) [3], there are a large number of intelligent electronic devices (IEDs), which are communication endpoints on the cyber infrastructure (i.e., receiving and handling to commands and queries via IEC standard protocols) while at the same time are responsible for interacting with physical power system components. Besides, there is a substation gateway device (also called proxy in the IEC's model [3]), which is responsible for protocol translation, e.g., between IEC 60870-5-104 used on WAN and IEC 61850 used in substation local area network (LAN). A large-scale substation may also have a local SCADA HMI for monitoring and controlling devices within the substation. Furthermore, recently implementation of VPN interface for substation network is becoming common for the sake or remote maintenance by engineers and/or as backup SCADA communication channel. Typically a modernized substation system is organized in 3 levels, namely station level where the gateway, SCADA HMI, and VPN servers are located, bay level where IEDs are deployed, and process level where physical power system components are deployed. While, for communication in the station level and communication between the station level and bay level (also called station bus), IEC 61850 MMS protocol is commonly used in the modernized substation system, the communication at the bay level and below (also called process bus) is usually done by using IEC 61850 GOOSE [15, 16].

## 3.2 Preliminary Implementation of Virtual Smart Grid Infrastructure

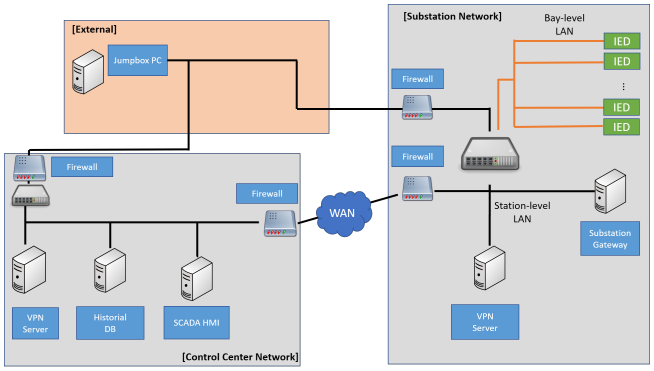


Figure 1: Overview of Virtual Smart Grid System

Based on these observations, we designed and implemented the virtual smart grid infrastructure, which is seen as the prototype of the smart grid honeypot, by using open-source tools, virtualization technologies, and a back-end power flow simulator, following the approach employed in [2]. In particular, we utilized the same open-source smart grid testbed, called

Table 1: Smart Grid Testbed Components

| Segment | Name | OS | Services | Virtual? | Description |
|---|---|---|---|---|---|
| External | Jumpbox PC | Linux | SSH | Yes | Entry point for remote researchers/pentesters |
| Control Center | Firewall (1) | Linux | - | Yes | Firewall facing external network. Block inbound traffic except for the port-forwarding for VPN |
| | Firewall (2) | Linux | - | Yes | Firewall facing WAN, which allows only TCP session initiated by devices in the control center |
| | VPN Server | Linux | VPN | No | OpenVPN server for remote access from External segment. Enforces password-based authentication. Besides, we plant ShellShock vulnerability [17] on this VPN service. |
| | SCADA HMI | Windows | RDP, Web | No | Runs OSHMI and periodically sends general interrogation request towards substation using IEC 60870-5-104 protocol. |
| | Historian | Windows | DB | No | Runs Timescale DB on PostgreSQL database to store data collected by SCADA HMI. |
| Substation | Firewall (1) | Linux | - | Yes | Firewall facing external network. Block inbound traffic except for the port-forwarding (VPN and SSH) |
| | Firewall (2) | Linux | - | Yes | Firewall facing WAN, which blocks inbound traffic other than IEC 60870-5-104. |
| | VPN Server | Linux | VPN | Yes | OpenVPN server for remote access from External segment. Enforces password-based authentication with weak password. |
| | Substation gateway | Linux | IEC 60870-5-104 | Yes | Protocol translation gateway for SCADA communication, which handles IEC 60870-5-104 and IEC 61850 MMS. |
| | IED(s) | Linux | IEC 61850 MMS | Yes | Virtual communication end points that interact with back-end power system simulator according to the received MMS message (control and interrogation commands) |

SoftGrid [18], as a building block. While it is not intended for use as honeypot per se, it offers cyber-connected power system simulation. The logical and physical views are shown in Figure 1. As seen in the figure, the setup consists of 2 network locations, namely the control center and a substation. Each includes components discussed earlier.

When configuring all system components, we followed typical configurations used in real systems. Both the control center and substation have a firewall at the WAN side, which implements appropriate network traffic filtering. The firewall on the substation side filters incoming traffic other than port 2404, which is used for IEC 60870-5-104 protocol, the VPN service for remote access, and the SSH service used for device configurations etc. On the other hand, the WAN-facing firewall at the control center side filters all incoming traffic (except for the TCP traffic on a connection established from the control center side). In addition to the WAN, both the control center and substation implement VPN interface connected to the external network.

Within the control center system, we set up OSHMI [19], an open-source SCADA HMI that supports IEC 60870-5-104 protocol that is configured to issue periodic general interrogation (querying measurements/status from all IEDs) to the substation. Because use of Windows PCs is still common in the industrial control systems, as seen in the recent Ukraine incident reports [20, 21], we used a Windows PC to host it. SCADA HMI in the control center is typically used with a database system, also called a historian, to store the collected power grid measurements, as seen in the state-of-the-art, IEC 61850-based smart grid testbed [15,22]. OSHMI is often used with Timescale DB [23], we also set it up on another PC. OpenVPN [24] is utilized for implementing the VPN.

Regarding the substation system, we utilized the substation gateway, which is responsible for protocol translation between IEC 60870-5-104 and IEC 61850, as well as IEC 61850-compliant IEDs that are part of SoftGrid [18] with modification to support the general interrogation, which allows the control center to send a single query message for the substation gateway so that it collects measurements and status from all IEDs in the substation and send them back together. As discussed earlier, such communication on the station bus of the substation utilizes the IEC 61850 MMS protocol. Besides, just like the substation honeypot system developed in [2], Mininet [25] is utilized to implement virtual IEDs and connectivity among them. The virtual IEDs are connected to back-end SoftGrid IEDs, which, behind the scenes, interact with the power flow simulator according to IEC 61850 messages, in a one-to-one manner. Because the

process-level communication is typically implemented as a separate network, which is often not Ethernet based, we did not specifically include it. However, it is approximated by SoftGrid IEDs [18].

Finally, the orange region represents an entry point for penetration testers (or hypothetical attackers). In this segment, we implement 2 separate firewalls connected to the aforementioned 2 network sites. Note that this segment is considered as DMZ connected to external, public networks, such as the Internet where attackers would reside. Alternatively, it can be positioned as part of enterprise IT network of the power grid operator, which uses VPN connection to access the control system, just like the Ukraine case [21]. While the access to this segment is currently limited to our partners, when we expose the honeypot, we can directly connect it to the Internet.

## 4 Evaluating the Smart Grid Honeypot

While we consider this design to be what many researchers would likely start with, it is not clear what clues potential attackers would find to identify the system as a honeypot. In order to evaluate the realism of the virtual smart grid infrastructure, we performed scanning by means of penetration testing by cybersecurity experts. We first explain our attack scenarios that are realistic in the modernized power grid system, and then summarize the findings.

### 4.1 Penetration Scenarios

We defined attack scenarios based on the common attack vector listed by ICS-CERT [26]. One of the most popular attacks is one through the corporate IT network, from which the attacker will then pivot to the OT (operational technology) network (i.e., in our case smart grid communication infrastructure). Another common attack vector is when OT devices are exposed directly to the Internet. This is commonly done to allow OT device vendors to remotely access the devices to conduct troubleshooting. The rest of this section elaborates the attack scenarios that are followed when we performed the evaluation of honeypot system from an attacker's perspective.

We assume the stage one of the ICS cyber kill chain [27] to be successful and that the attacker has gained a foothold in the corporate network of the power grid operator or one of the contractors that supply the devices deployed in the plant. The attacker then proceeds to search for VPN access from the corporate network into the grid operator's IT network and then from there through to the actual OT network. As is common, a contractor might have direct VPN access to the OT network to access the end devices for the purpose of troubleshooting and maintenance. Another possibility that may be sought by attackers would be scanning for other types of remote access interfaces, such as SSH. Also, sometimes Remote Desktop (RDP) services might be exposed intentionally (for convenience) or by mistake. These services may be of attackers' interest.

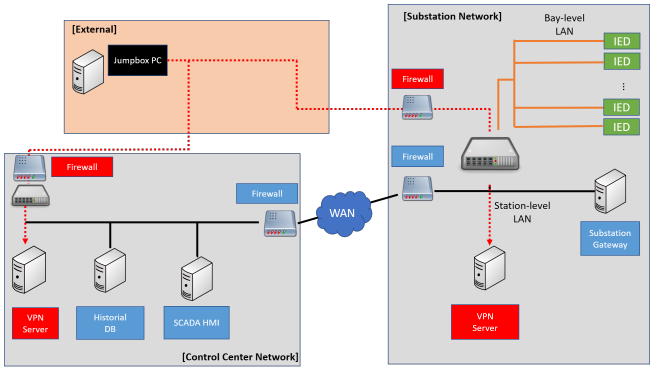


Figure 2: Penetration and Enumeration

After successfully penetrating into the control center and/or substation, the next step of the attack is to enumerate devices and services to mount attacks. The primary targets will be workstations and HMI in the control center as well as substation (see Figure 2). In the figure, the black box indicates the source of attack while red indicates path of the attack.

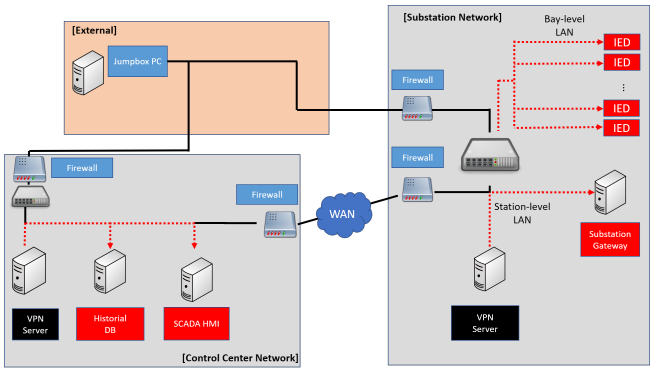


Figure 3: Sending Malicious Commands to ICS Devices

Since the ultimate goal of attackers is to make physical impact to disturb power grid operations, they attempt to send out control/query commands to ICS devices (e.g., a protocol translator or IED) in the substation. Such an attempt may be done by scanning ICS devices in the network to directly inject malicious commands (see Figure 3).

Overall, the attack scenario we consider in this paper involves the following attack vectors:

- Scanning for VPN/SSH/RDP services
- Compromising the identified remote access service
- Creating a persistent account on the compromised host
- Enumerating and launching attack

## 4.2 Findings from Penetration Testing

Based on the initial smart grid honeypot system (see Figure 1), we conducted attack experiments to evaluate the implementation from an attacker's perspective. We grant a hypothetical attacker (i.e., cybersecurity experts we collaborate) access to the *jumpbox* in the orange segment in Figure 1 via VPN, and then executed the testing according to the aforementioned attack scenario. Tools used by the pen-tester included the following: Nmap [1], Metasploit [28], both of which are widely used. Based on the evaluation by the cybersecurity expert, the following anomalous/suspicious traits that could hint attackers were reported.

(1) Presence of virtual machines hinted by open ports

(2) OS/device fingerprinting results that are different from typical smart grid devices (IEDs, substation gateways)

(3) Lack of user accounts on Windows machines, which does not look like active, lively used systems

Further experiments involving more human subjects are necessary for extensive evaluation, which is planned in our future work. Having that said, a penetration testing imitates the typical or experienced attacker's strategy and thus finding here is still meaningful as the baseline.

Regarding Problem (1), it is identified through the network ports (902 and 912) opened by default by VMWare software, which is used to run virtual machines in the control center. Regarding Problem (2), Nmap's OS fingerprint of the virtual IED developed in our earlier work appears as shown in "Initial IED" raw of Table 4. Compared to the real IED's fingerprint, which is collected from IEDs deployed in a IEC 61850-based smart grid testbed [22] and also shown in Table 4, there are a number of differences, which are highlighted with yellow. Besides, owing to the fact that each virtual IED is implemented as a virtual node on Mininet [25], which is basically a copy of the host OS, Nmap identifies that each virtual node (i.e., virtual IED) runs *"Linux 3.2-4.9"*, while the fingerprinting result for the real IED in the smart grid testbed [22] is *"No exact OS matches"*. Regarding (3), since an attacker often attempts to exploit existing user account to compromise Windows machines, lack of any user account may seem suspicious.

# 5 Enhancing the Smart Grid Honeypot

Based on the findings from the testing in the previous section, in this section we elaborate our strategy to improve our honeypot implementation. Problem (1) was addressed closing the corresponding ports to hide the presence of virtualization technologies. Note that closing ports after VMs are started didn't affect the functionality of VMs. Regarding Problem (3), we change Windows machine (VM) configurations and created an user account with administrator privilege and with weak password, which is often the case in real-world control systems. In the rest of section we mainly discuss Problem (2).

In the smart grid system, there are two types of ICS devices. The first category includes devices that only work as servers that open TCP and/or UDP ports to accept request messages (e.g., interrogation or control commands) from clients. IEDs are usually working only as IEC 61850 MMS servers, and thus belong to this category. On the other hand, programmable logic controllers (PLCs) or protocol protocol translators work not only as servers but also clients. For instance, a protocol translators work as a server for one protocol (e.g., IEC 60870-5-104) as well as a client for another protocol. PLCs usually collect measurements from IEDs and then based on such inputs issue control commands according to the predefined logic. In the rest of this section, we call the former type of devices *passive* device while call the latter *active* devices and discuss our countermeasures separately.

## 5.1 Fingerprinting for Passive Devices

Device or OS fingerprinting is typically done by analyzing OS-specific characteristics found in network protocol stack implementation. For example, Nmap [1] sends crafted packets to check response from the device of interest, which we call *active* fingerprinting. Ideally, we could make modification in kernel level to tweak the protocol stack implementation, it is not feasible in practice. To counter such fingerprinting tools and techniques more practically, we introduce a tool called Honeyd [5], an open-source honeypot software. One notable feature of Honeyd is that it can deceive the active fingerprinting tools like Nmap. Specifically, if we want to make a honeypot resemble to a certain device, we can collect the OS fingerprint of the device and configure it on Honeyd. By doing so, Honeyd can imitate the protocol stack characteristics to fool active fingerprinting tools. For this project, we utilized fingerprints collected from real smart grid devices in the IEC 61850-based smart grid testbed [22] for the sake of realism. Regarding MAC address, we configured MAC addresses on Honeyd that belong to the same vendors as the real IEDs (namely "ipcas GmbH" and "Siemens AG" in [22]).

Besides the protocol stack characteristics, it is also crucial to run the same set of services as the real smart grid devices. For instance, IEDs in [22] runs web services besides the IEC 61850 MMS service. We employed Nginx for web services, because it is widely used for embedded devices, and configured protocol headers according to the real device. We also loaded the fake HTML contents that resemble the real device's. Regarding IEC 61850 MMS, we utilized SoftGrid [18].

The resulting architecture of our virtual IED (i.e., a passive device) is found in Figure 4. Note that the web service and IEC 61850 service are running behind Honeyd and all incoming packets are "proxyed" by it to deceive fingerprinting. In the figure we see another box labeled "IEC 61850 GOOSE Traffic Generator". IEC 61850 compliant IEDs utilize IEC 61850

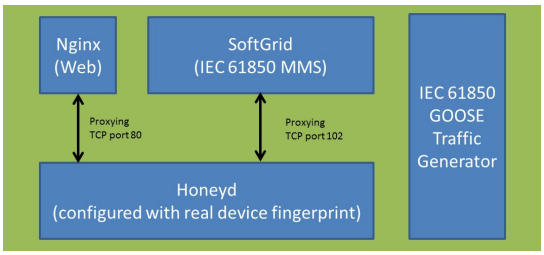


Figure 4: Improved Virtual IED Architecture

GOOSE protocol at the link layer for status exchange among IEDs [15]. Thus, we incorporated a dummy traffic generator for better realism.

## 5.2 Fingerprinting for Active Devices

To counter fingerprinting for active devices, use of Honeyd alone is not sufficient, because outgoing traffic sent by the virtual device is not mediated by Honeyd and thus cannot benefit from the aforementioned counter-fingerprinting feature. Such outgoing traffic is subject to passive fingerprinting tool like P0f [29]. Passive OS fingerprinting tools basically sniff network and detect device or OS type of a target device.

To evade passive fingerprinting tools, one viable solution is to utilize the same type of OS with similar version to real devices. Many PLCs and protocol translators run on Linux operating system, and are fingerprinted accordingly. Thus, our goal here is to make our virtual devices are fingerprinted likewise. In addition, like the case of passive devices, we run network services found on real devices. Based on our study, PLCs in the IEC 61850-based smart grid testbed offer SSH services, and thus, we utilized Cowrie [8], an open-source SSH honeypot, and configured banner information according to the real devices. We also found some product specific ports opened, and we used simple TCP listeners for such ports.

As seen in Figure 5, we added a PLC implemented this way since it is often found in the real-world substation [22].

# 6 Logging in Honeypot

Reliable and comprehensive logging is one of the essential features for honeypot systems to enable later threat intelligent analysis. We first summarize the logging available on the initial honeypot system (Figure 1) in Table 2.

Table 2: Logging Available on Initial Honeypot

| Location | Logging Feature | Secure? |
|---|---|---|
| VPN Server | Shell command history | No |
| HMI / Historian | Windows default logging | No |
| Virtual IED | Network Monitor on Mininet host | Yes |

For reliable data collection, logging features should not be visible or accessible to attackers. For instance, while logging on (virtual or real) hosts on the honeypot would allow us to collect wide range of information (e.g., shell command history on Linux-based machines) such information can be easily compromised or tampered with by attackers. On the other hand, logging outside of the nodes (e.g., network monitor run on Mininet host to capture traffic around virtual IEDs) is not visible or accessible to attackers, and thus is secure.

In this direction, we explored additional logging features besides the ones listed in Table 2. In particular, based on the above observation, we consider deployment of "transparent proxy" (TP for short) boxes in appropriate places to ensure system-wide visibility. Transparent proxy is configured as a (virtual) machine that has 2 network interfaces, which are bridged. This way, it behaves just like a cable and thus is not addressed or detected by attackers. In the setup, we introduced transparent proxy virtual machines that run Wireshark network analyzer [30]. The deployment of such transparent proxy boxes is shown in Figure 5.

To collect more expressive information we considered application-level logging at virtual IEDs. Since the eventual goal of the attacker is to impact the physical system, such logging would serve as a starting point to back trace attackers' activities in other log data. In order to implement such a logging in a secure way, we take advantage of our "layered" virtual IED architecture, which consists of Honeyd as entry point and back-end SoftGrid IEDs. Honeyd, by default, implements logging feature to record source and destination address for each TCP session. In addition, logging based on IEC 61850 MMS payloads can be implemented on SoftGrid, which is run on another host behind the scene and thus is not visible to attackers. Instead of sending malicious IEC 61850 MMS commands directly to IED, it is also possible that an attacker would send IEC 60870-5-104 commands to the substation gateway logging. In such a case, IED-level logging alone cannot tell the source of the IEC 60870-5-104 commands. Thus, we additionally implement logging on top of the substation gateway. Note that, although our substation gateway exposes SSH port, it is connected to Cowrie and thus logging on the SoftGrid module on the gateway is not accessible to attackers. In sum, application-level logging we implemented is found in Table 3.

Table 3: Logging Available on Improved Honeypot

| Information | Location | Secure? |
|---|---|---|
| Timestamp of event | Honeyd and SoftGrid | Yes |
| Source IP | Honeyd | Yes |
| Destination IP | Honeyd | Yes |
| IEC 61850 MMS Command | SoftGrid (IED) | Yes |
| IEC 104 Command | SoftGrid (Gateway) | Yes |

# 7 Evaluation of the Improved Honeypot

In this section, we show evaluation of the improved honeypot (Figure 5), mainly focusing on device-level imitation and logging. We conducted the testing by the cybersecurity expert following the same scenario as in Section 4.1.

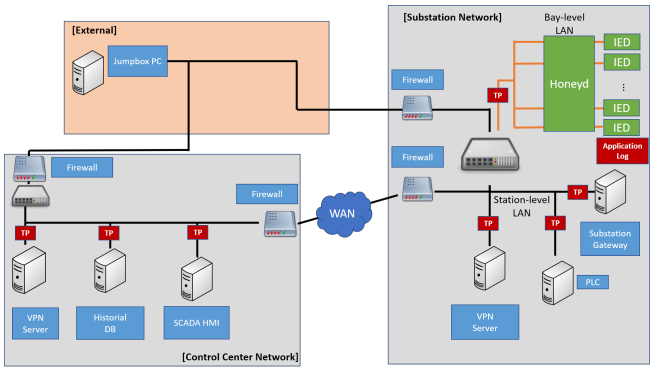


Figure 5: Improved Smart Grid Testbed

## 7.1 Improved IED/PLC Implementation

The major improvement in virtual IED implementation is found in similarity in OS fingerprints. For this evaluation, we utilized Nmap. The captured fingerprints are shown in Table 4. As clearly seen, the improved virtual IED on the top has noticeably similar than the original one at the bottom.

There are still some differences that are highlighted, but values with yellow background vary over multiple executions and thus, difference of this level is not an issue. We also see difference in "IPL" in the line starting with "U1". Based on our observation, IPL values for other PLCs, PC, and virtual machines are typically 164 while, as far as we are aware, the Siemens IED in the IEC 61850-based smart grid testbed [22] returns 240. The difference is caused by the size of padding ("00") in ICMP Port Unreachable message returned by the scanned device. However, this level of similarity is still effective to deceive attackers without the perfect, device-specific knowledge of real devices and retain them inside for a certain amount of time, which is our purpose.

Regarding our PLC and gateway, we need to worry about not only active but also passive fingerprinting. Thus we show the Nmap and P0f fingerprinting in Table 5 and Table 6 respectively. In sum, for both fingerprinting results, our implementation of the virtual PLC provides sufficient similarity. For Nmap,"ISR" and "SP" are the main differences. "ISR" is TCP ISN counter rate which reports the average rate of increase for the returned TCP initial sequence number. "SP" is TCP ISN sequence predictability index which roughly estimates how difficult it would be to predict the next ISN from the known sequence of six probe responses. These two flags are all average values. For P0f, the difference happened at "mss*" flag, "mss" is the maximum segment size. '*' indicate that MSS varies depending on the parameters of sender's network link, and should not be a part of the signature. In this case, MSS will be used to guess the type of network hookup.

## 7.2 Sufficiency of Logging

In order to evaluate the sufficiency of logging mechanism, we considered the staged attack scenario. The attack scenario

Table 4: Nmap Fingerprints of Virtual IEDs and Real IED

| IED system | Fingerprint |
|---|---|
| Improved IED | SEQ( SP=C5 %GCD=1% ISR=D5 %TI=I%CI=I%II=I% SS=O %TS=U)<br>ECN(R=N)<br>T1(R=Y%DF=N%T=72%S=O%A=S+%F=AS%RD=0%Q=)<br>T2(R=N)<br>T3(R=N)<br>T4(R=Y%DF=N%T=72%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)<br>T5(R=Y%DF=N%T=72%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)<br>T6(R=Y%DF=N%T=72%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)<br>T7(R=Y%DF=N%T=72%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)<br>U1(R=Y%DF=N%T=72% IPL=164 %UN=0%RIPL=G%RID=G%RIPCK=G%<br>RUCK=6339%RUD=I)<br>IE(R=Y%DFI=N%T=72%CD=Z) |
| Real IED | SEQ(SP=CD%GCD=1%ISR=D6%TI=I%CI=I%II=I%TS=U)<br>ECN(R=N)<br>T1(R=Y%DF=N%T=72%S=O%A=S+%F=AS%RD=0%Q=)<br>T2(R=N)<br>T3(R=N)<br>T4(R=Y%DF=N%T=72%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)<br>T5(R=Y%DF=N%T=72%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)<br>T6(R=Y%DF=N%T=72%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)<br>T7(R=Y%DF=N%T=72%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)<br>U1(R=Y%DF=N%T=72%IPL=240%UN=0%RIPL=G%RID=G%RIPCK=G%<br>RUCK=6339%RUD=I)<br>IE(R=Y%DFI=N%T=72%CD=Z) |
| Initial IED | SEQ( SP=106 %GCD=1% ISR=10E%TI=Z %CI=I%II=I% TS=8 )<br>OPS(O1=M5B4ST11NW9%O2=M5B4ST11NW9%O3=M5B4NNT11NW9%<br>O4=M5B4ST11NW9%O5=M5B4ST11NW9%O6=M5B4ST11)<br>WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)<br>ECN(R=Y%DF=Y%T=40%W=7210%O=M5B4NNSNW9%CC=Y%Q=)<br>T1(R=Y% DF=Y%T=40 %S=O%A=S+%F=AS%RD=0%Q=)<br>T2(R=N)<br>T3(R=N)<br>T4(R=Y% DF=Y%T=40 %W=0%S=A%A=Z%F=R%O=%RD=0%Q=)<br>T5(R=Y% DF=Y%T=40 %W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)<br>T6(R=Y% DF=Y%T=40 %W=0%S=A%A=Z%F=R%O=%RD=0%Q=)<br>T7(R=Y% DF=Y%T=40 %W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)<br>U1(R=Y% DF=N%T=40%IPL=164 %UN=0%RIPL=G%RID=G%RIPCK=G%<br>RUCK=G%RUD=G )<br>IE(R=Y%DFI=N% T=40%CD=S ) |

follows a full ICS attack consisting of an attack on the administrative network to gain access to the OT network where the next phase of the attack is carried out. The attack scenario discussed in Section 4.1 can be divided into 5 phases as follows:

1. Attacking the exposed Jumpbox machine via SSH
2. Attacking the OpenVPN servers to gain access
3. Identifying machines of interests and attacking them
4. Accessing substation
5. Identifying OT devices of interests and attacking them

As the result of execution of these steps, access to both networks was successfully gained and machines of interests were able to be compromised. As the IEDs can be accessed from the substation network (the station level), an attacker could also attempt to directly access and attack the IEDs through the exposed IEC 61850 MMS service instead of going though through the protocol translation gateway.

Table 7 summarizes our analysis on logging sufficiency. As seen, we can effectively capture the information for threat intelligence analysis, which is our future work.

Table 5: Nmap Fingerprints of Virtual and Real PLCs

| PLC system | Fingerprint |
|---|---|
| Real PLC | PORT STATE SERVICE REASON<br>22/tcp open ssh syn-ack ttl 64<br>80/tcp open http syn-ack ttl 64<br>443/tcp open https syn-ack ttl 64<br>8080/tcp open http-proxy syn-ack ttl 64<br>MAC Address: 00:30:DE:40:D0:DB (Wago Kontakttechnik Gmbh)<br>Device type: general purpose<br>Running: Linux 2.6.X|3.X<br>OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3<br>OS details: Linux 2.6.32 - 3.10<br>TCP/IP fingerprint:<br>SEQ( SP=104 %GCD=1% ISR=109 %TI=Z%CI=Z%II=I%TS=7)<br>OPS(O1=M5B4ST11NW5%O2=M5B4ST11NW5%O3=M5B4NNT11NW5<br>O4=M5B4ST11NW5%O5=M5B4ST11NW5%O6=M5B4ST11)<br>WIN(W1=3890%W2=3890%W3=3890%W4=3890%W5=3890%W6=3890)<br>ECN(R=Y%DF=Y%T=40%W=3908%O=M5B4NNSNW5%CC=Y%Q=)<br>T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)<br>T2(R=N)<br>T3(R=N)<br>T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)<br>T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)<br>T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)<br>T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)<br>U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)<br>IE(R=Y%DFI=N%T=40%CD=S) |
| Virtual PLC | PORT STATE SERVICE REASON<br>22/tcp open ssh syn-ack ttl 64<br>80/tcp open http syn-ack ttl 64<br>443/tcp open https syn-ack ttl 64<br>8080/tcp open http-proxy syn-ack ttl 64<br>MAC Address: 00:30:DE:00:00:01 (Wago Kontakttechnik Gmbh)<br>Device type: general purpose<br>Running: Linux 2.6.X|3.X<br>OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3<br>OS details: Linux 2.6.32 - 3.10<br>TCP/IP fingerprint:<br>SEQ( SP=105 %GCD=1% ISR=105 %TI=I%CI=I%II=I%TS=U)<br>OPS(O1=M5B4ST11NW5%O2=M5B4ST11NW5%O3=M5B4NNT11NW5<br>O4=M5B4ST11NW5%O5=M5B4ST11NW5%O6=M5B4ST11)<br>WIN(W1=3890%W2=3890%W3=3890%W4=3890%W5=3890%W6=3890)<br>ECN(R=Y%DF=Y%T=40%W=3908%O=M5B4NNSNW5%CC=Y%Q=)<br>T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)<br>T2(R=N)<br>T3(R=N)<br>T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)<br>T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)<br>T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)<br>T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)<br>U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)<br>IE(R=Y%DFI=N%T=40%CD=S) |

Table 6: P0f Fingerprints of Virtual and Real PLCs

| PLC system | Fingerprint |
|---|---|
| RealPLC | mod=syn\|cli=172.16.1.211/40392\|srv=<br>172.16.2.41/102\|subj=cli\|os=Linux 3.11 and<br>newer\|dist=0\|params=none\|raw_sig<br>=4:64+0:0:1460:mss*20,7:mss,sok,ts,nop,ws:df,id+:0<br>mod=mtu\|cli=172.16.1.211/40392\|srv=172.16.2.41/102\|<br>subj=cli\|link=Ethernet or modem\|raw_mtu=1500<br>mod=syn+ack\|cli=172.16.1.211/40392<br>\|srv=172.16.2.41/102\|subj=srv\|os=???\|dist=0\|<br>params=none\|raw_sig=4:64+0:0:1460:<br>mss* 3,0 %:mss,sok,ts,nop,ws:df:0<br>mod=mtu\|cli=172.16.1.211/40392\|srv=172.16.2.41/102\|<br>subj=srv\|link=Ethernet or modem\|raw_mtu=1500 |
| Virtual PLC | mod=syn\|cli=172.16.1.211/40392\|srv=<br>172.16.2.41/102\|subj=cli\|os=Linux 3.11 and<br>newer\|dist=0\|params=none\|raw_sig<br>=4:64+0:0:1460:mss*20,7:mss,sok,ts,nop,ws:df,id+:0<br>mod=mtu\|cli=172.16.1.211/40392\|srv=172.16.2.41/102\|<br>subj=cli\|link=Ethernet or modem\|raw_mtu=1500<br>mod=syn+ack\|cli=172.16.1.211/40392<br>\|srv=172.16.2.41/102\|subj=srv\|os=???\|dist=0\|<br>params=none\|raw_sig=4:64+0:0:1460:<br>mss* 20,7 %:mss,sok,ts,nop,ws:df:0<br>mod=mtu\|cli=172.16.1.211/40392\|srv=172.16.2.41/102\|<br>subj=srv\|link=Ethernet or modem\|raw_mtu=1500 |

## 8 Conclusions

In this paper, we designed a honeypot for smart grid communication infrastructure and performed evaluation of the smart grid honeypot system from an attacker's perspective towards further improvement of its realism. We then developed the system improved by using the insights from the evaluation. The developed system can not only be used as a honeypot system to deceive attackers for collecting threat intelligence but also be utilized as a sandboxed, virtual environment for cybersecurity experiments as well as a venue for training and education (e.g., capture-the-flag competition). We also ported the system onto National Cybersecurity R&D Lab (NCL) testbed (`https://ncl.sg/`) for broader accessibility and usability. The virtual machine images and guideline to reproduce the honeypot setup are also published [1].

In future work, we conduct experiments to evaluate how the improvement matters in practice, e.g., by means of a hacking competition inviting broader experts. We further plan to deploy the honeypot to collect real-world attack data. The collected data will be used for threat intelligence analysis as well as the automated translation of such intelligence into functional cybersecurity configurations, such as rules for firewalls and/or intrusion detection systems.

Table 7: Evaluation of Logging under Attack Scenario

| Phase | Effective Logging / Monitoring |
|---|---|
| 1 | A transparent proxy deployed in front of the jumpbox machine can detect attempt of SSH brute forcing. |
| 2 | A transparent proxy deployed in front of the VPN server in the control center can detect VPN connection attempts and reverse shell traffic initiated by the VPN server (as part of Shellshock attack [17]). |
| 3 | (Decrypted) traffic outgoing from the VPN server can also go through the transparent proxy to capture attackers access attempts to other machines in the control center network. Additional transparent proxy machines in front of each computers are also deployed to capture IEC 60870-5-104 traffic initiated by an attacker as well as potential lateral movement of attackers. |
| 4 | A transparent proxy is also deployed in the substation network and it can monitor traffic incoming into and outgoing from the VPN server. |
| 5 | Transparent proxy deployed in front of our substation gateway as well as virtual IEDs can capture the ICS communication traffic to log network-level information. In addition, more specific command types and target information are logged by the substation gateway and virtual IEDs. |

## Acknowledgment

This material is based on research/work supported in part by the Singapore National Research Foundation and the Cybersecurity R&D Consortium Grant Office under Seed Grant Award No. CRDCG2018-S01. This research is partly supported by the National Research Foundation, Singapore, Singapore University of Technology and Design under its National Satellite of Excellence in Design Science and Technology for Secure Critical Infrastructure Grant (NSoE_DeST-SCI2019-005).

[1] https://www.illinois.adsc.com.sg/spotify/index.html

## References


[1] "Nmap: the network mapper," https://nmap.org/.

[2] D. Mashima, B. Chen, P. Gunathilaka, and E. L. Tjiong, "Towards a grid-wide, high-fidelity electrical substation honeynet," in *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2017, pp. 89–95.

[3] IEC TC57, "IEC 61850-90-2 TR: Communication networks and systems for power utility automation – part 90-2: Using iec 61850 for the communication between substations and control centres," *International Electro technical Commission Std*, 2015.

[4] V. Pothamsetty and M. Franz, "Scada honeynet project: Building honeypots for industrial networks," http://scadahoneynet.sourceforge.net/.

[5] "Developments of the honeyd virtual honeypot," http://www.honeyd.org/.

[6] "Digital bond," http://www.digitalbond.com/tools/scada-honeynet.

[7] "CONPOT ICS/SCADA honeypot," https://www.conpot.org.

[8] "Cowrie ssh and telnet honeypot," [Online]. Available: https://www.cowrie.org/, (Date last accessed on Mar. 26, 201p).

[9] A. Mirian, Z. Ma, D. Adrian, M. Tischer, T. Chuenchujit, T. Yardley, R. Berthier, J. Mason, Z. Durumeric, J. A. Halderman *et al.*, "An internet-wide view of ics devices," in *14th IEEE Privacy, Security, and Trust Conference (PST'16)*, 2016.

[10] 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.

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

[12] "Shodan," https://www.shodan.io/.

[13] "Honeypot or not?" https://honeyscore.shodan.io/.

[14] J. Guarnizo, A. Tambe, S. S. Bunia, M. Ochoa, N. Tippenhauer, A. Shabtai, and Y. Elovici, "Siphon: Towards scalable high-interaction physical honeypots," *arXiv preprint arXiv:1701.02446*, 2017.

[15] A. Siddiqi, N. O. Tippenhauer, D. Mashima, and B. Chen, "On practical threat scenario testing in an electric power ics testbed," in *Proceedings of the Cyber-Physical System Security Workshop (CPSS), co-located with ASIACCS*, June 2018.

[16] D. Yang, D. Mashima, W. Lin, and J. Zhou, "DecIED: Scalable k-anonymous deception for iec61850-compliant smart grid systems," in *6th ACM Cyber-Physical System Security Workshop*. ACM, 2020.

[17] "Shellshock vulnerability," https://owasp.org/www-pdf-archive/Shellshock_-_Tudor_Enache.pdf.

[18] P. Gunathilaka, D. Mashima, and B. Chen, "Softgrid: A software-based smart grid testbed for evaluating substation cybersecurity solutions," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*. ACM, 2016, pp. 113–124.

[19] "OSHMI Open Substation HMI," https://oshmiopensubstationhmi.sourceforge.io/.

[20] K. Zetter, "Inside the cunning, unprecedented hack of ukraine's power grid," [Online]. Available: http://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/, (Date last accessed on Jun. 7, 2017).

[21] E. I. Sharing and A. C. (E-ISAC), "Analysis of the cyber attack on the ukrainian power grid," 2016.

[22] "Electric power and intelligent control (epic) testbed," [Online]. Available: https://itrust.sutd.edu.sg/wp-content/uploads/sites/3/2019/02/EPIC_technical_details-231018-v1.2.pdf, 2018, (Date last accessed on Feb. 12, 2019).

[23] "Open-source time-series database powered by postgresql," https://www.timescale.com/.

[24] "OpenVPN," https://openvpn.net/.

[25] "Mininet," http://mininet.org/.

[26] "Overview of cyber vulnerabilities," https://www.us-cert.gov/ics/content/overview-cyber-vulnerabilities.

[27] M. J. Assante and R. M. Lee, "The industrial control system cyber kill chain," *SANS Institute InfoSec Reading Room*, 2015.

[28] "Penetration testing software," https://www.metasploit.com/.

[29] "p0f v3," https://lcamtuf.coredump.cx/p0f3/.

[30] "Wireshark," https://www.wireshark.org/.