Towards a Grid-wide, High-fidelity Electrical Substation Honeynet

Daisuke Mashima, Binbin Chen, and Prageeth Gunathilaka
Advanced Digital Sciences Center
Singapore
{daisuke.m, binbin.chen, prageeth.g}@adsc.com.sg

Edwin Lesmana Tjiong
National University of Singapore
Singapore
e0013307@u.nus.edu

***Abstract*—After a number of real-world cyber incidents targeting power grid systems in the recent years, early detection of such cyber attacks and analysis of their attack vectors are in urgent need. One of the technologies that serve such purposes is honeypot. Although the concept of honeypot is not new, its use in a smart grid context is still nascent. Moreover, honeynet, a network of honeypots, is not yet systematically explored in a smart grid context. In this paper, we design and implement a virtual smart grid honeynet system that can emulate an entire smart grid field communication infrastructure including multiple networked substations by taking advantage of virtualization technologies. Our honeynet system is tied to a simulated power grid to provide consistent and realistic view to deceive attackers for a prolonged period. Especially, our system can emulate high-fidelity power grid behavior when a sophisticated attacker launches *grid-wide probing* at his preparation phase, by changing settings in one substation and examining the expected changes in another. Finally, we present our proof-of-concept implementation and evaluate its realism, performance, and scalability.**

I. Introduction

In the recent years, we observed real-world cyber attacks aiming at subverting power grid systems, and unfortunately some of them were successful. The notable example is the Ukraine case that caused a massive blackout in 2015 [1]. In this specific incident, attackers utilized a combination of cyber-attack strategies, ranging from spear phishing for sending malicious files to targeted employees and computers, malware for enabling remote manipulation of the control system, and malicious firmware update on routers for slowing down recovery [2]. As a result, the control center system was abused to send a large number of "open" commands to circuit breakers, leaving nearly 30 substations off-line for hours.

One cyber-security measure that helps detect and analyze unknown attack vectors is honeypot. Honeypot is in general a "decoy" system designed to lure cyber-attackers for the sake of early detection of attack attempts, slowing down and mitigation of attacks, and/or gathering of real-world attack traces for learning attack vectors and designing cyber security systems. In particular, we emphasize use of a honeypot for retaining attackers in it to buy time for defending real systems and tracing back attack sources, before massive attack against a real system is launched. There is an advanced concept called *honeynet*, which usually consists of a network of multiple honeypots. Although honeypot systems for enterprise IT systems are well-explored [3], ones for cyber-physical systems (CPS) pose unique challenges and therefore are not established yet. Besides, most of the literatures in CPS honeypot, or more specifically smart grid honeypot, emulate a standalone device or a small-scale system, namely a single IED (intelligent electronic device) [4] or a single, independent plant [5], and multiple instances of honeypots are not connected or coordinated with each other. However, in reality, a smart grid field system consists of multiple modernized substations implementing remote control and automation, all of which are tied to a control center, power grid, and often with each other. Unfortunately, existing smart grid/CPS honeypot projects do not provide such a grid-wide, high-fidelity environment for deceiving and retaining sophisticated attackers for a long duration and therefore are unable to offer sufficient time buffer for defense of real systems or to help the analysis of longitudinal, horizontal, coordinated attack activities.

In this paper, through the discussion of attacker models and practical honeypot deployment strategies to counter them, we derive requirements in honeypot systems in smart grid context. We then present design of a comprehensive and scalable smart grid honeynet system that consists of multiple instances of honeypot substations connected to a simulated power grid for providing attackers with realistic network visibility as well as fake, but consistent, power grid view and control experience. We further elaborate the system architecture and present a proof-of-concept implementation using open-source tools, namely VirtualBox [6] for host and device vitualization, Mininet [7] for network emulation, and SoftGrid [8] for integrated cyber-physical simulation. Using virtualization technologies, even a commodity quad-core PC can safely emulate up to 16 substations (nearly as many substations as the number of over-230kV substations in Singapore), each of which can be flexibly configured with different network topology and configuration. The scale can be increased by using higher-end, many-core processors or a cluster of PCs, such as [9], as will be demonstrated later. To the best of our knowledge, ours is the first in developing a smart grid honeynet that emulates a network of modernized substations, backed by a grid-wide, high-fidelity power system simulation.

The rest of this paper is organized as follows. In Section II, we overview related work focusing on cyber-physical system (CPS) and smart grid honeypot systems. We then discuss attacker models and the use of honeynet against them, which

helps derive the honeynet requirements, in Section III. System design and prototype implementation are presented in Section IV, which are then evaluated in Section V. Finally, we conclude the paper with future directions in Section VI.

II. Related Work

There are a number of efforts devoted to the development of honeypots in the CPS domain [10], [11], [12]. Among them, Conpot [13] 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 fingerprint its presence once attackers get 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 could be detected relatively easily [14]. Thus, it is not suitable for retaining attackers for longitudinal analysis of attack behaviors.

ShaPe [4] is a honeypot system that is specifically designed for smart grid. However, their focus is emulating at a device level, i.e., intelligent electronic devices (IEDs), and they do not provide any support of physical system simulation. In comparison, our honeynet emulates a comprehensive, connected smart grid system and its physical behavior. SoftGrid [8] is a software-based testbed to emulate smart grid remote control setup, consisting of control center and substations with IEDs that support standard protocols (IEC 60870-5-104 and IEC 61850). SoftGrid integrates the SCADA communication at the cyber side with a simulated physical system on PowerWorld [15], a high-fidelity power flow simulator. However, SoftGrid itself is not sufficient to be a honeypot owing to lack of realism to deceive attackers, even though it offers useful building blocks for our honeynet implementation.

In [5], a design for a virtual, high interaction, server-based ICS honeypot is proposed. Based on their MiniCPS framework [16], the authors developed a honeypot connected to a simulated water treatment system. However, besides the fact that their system is not designed for smart power grid, their honeypot focuses on emulating a single site (i.e., a honeypot server connected to one physical plant), and consideration on cyber-physical system consisting of multiple, connected sites is not specifically made. SIPHON [17] is a high-interaction honeypot for detecting attacks against IoT (Internet of Things) devices. It utilizes a small number of real devices and makes them accessible and controllable from multiple network locations for scale-up. However, such a strategy is possible only when each device is assumed to be independent by attackers, which is not the case for smart grid systems.

When designing honeypot systems, we should also worry about honeypot fingerprinting by attackers. We can find a number of research outcomes in this direction. In [14], the authors studied ICS devices exposed to the Internet and also investigated the scanning activities of such devices by using Conpot honeypot systems deployed on Amazon EC2. They developed a heuristic-based scheme to tell whether a discovered device is a honeypot, specifically Conpot, or not. Shodan also offers a feature to tell if the device with a specified IP address is a honeypot [18]. Although the logic behind it is not published, it is likely that they rely on characteristics of popular honeypot implementations [17].

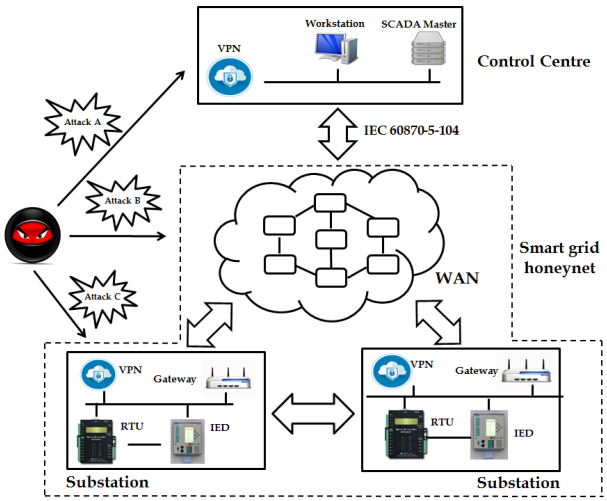


Fig. 1. Smart Grid Communication Architecture and Attacker Models. Dashed lines indicate the scope emulated by a smart grid honeynet system.

III. The Needs for a Grid-wide High-fidelity Electrical Substation Honeynet

In this section, we first discuss attacker models that are practically considered in smart grid communication. Then, we consider honeypot deployment strategies, based on which requirements in a smart grid honeynet system will be defined.

A. Attacker Models Considered in Our Work

Fig. 1 depicts the typical communication infrastructure in smart grid systems and three potential entry points for cyber attackers we consider in this work, which are elaborated below.
[Attack A] Attack from control center: There may be two cases in this category. The first, and worst, case is that an attacker gains access to the SCADA master system at the control center, just like the Ukraine incident [2]. Then, he would be able to issue arbitrary, legitimate-looking, interrogation and control command to substations. Another possibility is the attacker only gains access to the normal workstation within the control center and mounts man-in-the-middle (MITM) attacks [19], [20] for sending replayed or forged commands as well as intercepting messages from substations.
[Attack B] Attack from network: Although typically private, dedicated network is used, some of the substations might be connected to public network (e.g., Internet) either intentionally or by mistake, which could offer entry points for attackers on network. We admit that, even a dedicated infrastructure may not be fully secure (e.g., trunk ports of network equipments could be exploited). However, such an attack requires physical infrastructure access, which is therefore excluded from our scope. On the other hand, wireless or cellular communication is often used. In such a case, MITM attacks could be mounted

by compromising intermediate base stations [21]. Attackers could also establish rogue base stations for misleading communication as well as for impersonating the control center.

[Attack C] Attack via virtual private network (VPN) interface: Communication between control center and substations could take place via VPN, e.g., in emergency cases. VPN may be implemented on substation gateways for the sake of remote maintenance by operators or device vendors. Once attackers obtain access credentials by means of phishing etc., they could intrude into substations or use tunneling to inject control commands into substations. In either case, it is possible for them to cause significant physical damage. Once one substation is compromised, attackers may further attempt to access and compromise other substations via inter-substation channels.

To comprehensively support these attack surfaces, instead of a standalone honeypot, a smart grid honeynet, which should cover the scope shown in Fig. 1, is desired. In the next section we discuss how such a smart grid honeynet, as the first of its kind, can be deployed and operated to counter them.

B. Potential Use Cases of Smart Grid Honeynet

To catch an attacker that intrudes into the control center, we can deploy a dummy host (a SCADA master and/or normal workstation) that offers the same functionality as the original system but is made intentionally vulnerable so that an attacker would first try to compromise it. For instance, a dummy host could pretend to be a backup system. Design and implementation of such dummy hosts largely depend on the real system of interest, and therefore they are outside of the scope of this paper. The smart grid honeynet, which should present a comprehensive view of the entire power grid, can be connected to the dummy SCADA master system to provide sandboxed environment where an attacker can not only send arbitrary interrogation and control commands for probing and testing his capability at the preparation phase [2] but also mount attacks. Regarding attackers on a normal workstation attempting man-in-the-middle attacks, again we can direct traffic outgoing from the intentionally-vulnerable workstation entirely to the smart grid honeynet, e.g., by tweaking routing tables. In order to completely isolate the honeynet from the real communication infrastructure, such a vulnerable workstation can be used along with the aforementioned dummy SCADA master, which can be configured to generate realistic SCADA communication traffic. In this way, an attacker on the workstation would see realistic interrogation traffic from the dummy SCADA master and also observe reasonable response to control commands sent by the dummy SCADA master, without jeopardizing the real system.

Concerning attackers from network, a honeypot substation, which is part of the honeynet, can expose interface that supports protocols used in the smart grid context (e.g., IEC 60870-5-104) to accept commands from attackers. Such an interface could be made accessible to the Internet or other types of wide-area network. To attract attackers manipulating cellular communication, some of the honeypot substation gateways may implement a cellular module (e.g., GSM). Such a module would also contribute to detecting rogue base stations set up by an attacker. Implementation of cellular interface is not included in our current prototype and is part of our future enhancement.

In order to catch attackers targeting VPN interface implemented on substation gateways, we can expose honeypot substation's VPN interface (usually implemented on a gateway) to the Internet so that the honeypot will be found by attackers, for instance via Shodan [22]. We may need to additionally embed some known vulnerability so that an attacker could exploit it. In this scenario, the honeypot then must present realistic view of substation internals (e.g., LAN topology). Also, devices that are directly exposed and accessible to attackers, such as substation gateways, should not be easily fingerprinted as fake ones. Last but not the least, the honeypot should also implement inter-substation communication, which again motivates the need of honey*net*.

C. Requirements for Smart Grid Honeynet

Based on the discussion so far, we summarize the requirements for the smart grid honeynet we develop.

Comprehensive, Consistent Power Grid View: If an attacker is in the control center, he could technically have complete visibility throughout the grid, for example by passively eavesdropping communication traffic between the SCADA master and substations, as well as capability to control any remotely-controllable devices. Thus, it is imperative to emulate the entire communication infrastructure including interconnected substations. In addition, all of the substations in the honeynet should provide consistent view of the power grid to deceive an attacker upon his probing. Namely, when an attacker would perform some control on the grid (e.g., open a circuit breaker), the reasonable outcome, including change in power flow, voltage, and frequency, as well as updated status of the controlled circuit breaker, should be visible to the attacker when it is interrogated later.

Realistic Network Configuration: In order to attract and deceive attackers, network attributes, such as IP and MAC addresses, of virtual devices in the honeynet should be consistent with the real system. In addition, other network characteristics, including packet loss ratio, bandwidth, latency, protocol used, and network topology in substations, should also be well imitated. For instance, in many substation systems, IEC 60870-5-104 is used for communication with the control center and IEC 61850 is used on a substation local area network (LAN) organized in ring topology [23], [24]. It is also desired to support other associated protocols such as Spanning Tree Protocol to manage the topology.

Scalability for Grid-wide Emulation: In the real power grid, there can be a large number of substations. For instance, a power grid in Hong Kong has over 200 substations [25]. Thus, a system of this size should be supported with a reasonable infrastructure investment and operational cost. System virtualization technologies are often used for enhancing scalability.

Fingerprinting Resistance: There are a number of tools available for attackers, such as Nmap [26]. These can be used to mount scanning against nodes in the honeynet. Information derived by such tools should not reveal the presence of

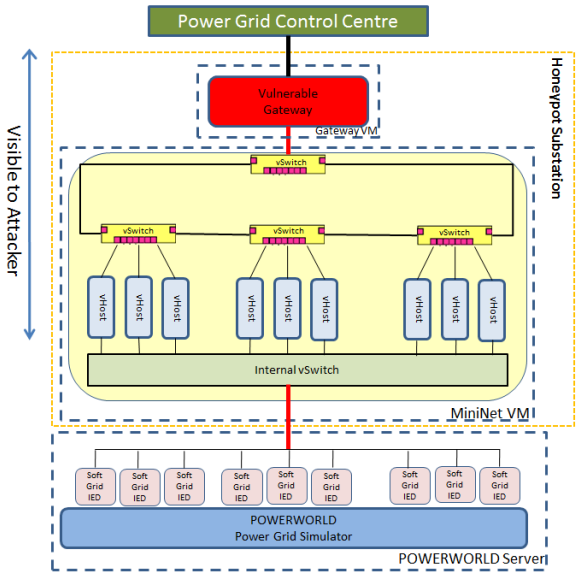


Fig. 2. Overview of a honeypot substation architecture. Blue dotted lines represent boundaries of a machine (or virtual machine).

honeypots. In addition, each node that is to be compromised by attackers should not exhibit characteristics of honeypots (e.g., a peculiar process used by a honeypot implementation), even when he obtains shell access. Use of virtualization technologies should also not be recognizable. In particular, when multiple virtual substations are hosted on a single physical machine for the sake of scalability, isolation among them should be taken into account (e.g., high CPU load on one instance should not impact performance of another).

IV. Design and Implementation

In this section, we present our proof-of-concept implementation to accomplish the requirements discussed in the previous section. The high-level design of a single honeypot substation is shown in Fig. 2. Our honeynet consists of multiple instances of honeypot substations whose gateways are interconnected.

A. Substation Gateway

A gateway device in the substation serves as the entry point to the substation network from the wide-area network. Being the single point of connection, this device is usually the most vulnerable to attacks. Besides handling connection with the control center and/or peer substations, a substation gateway typically implements protocol translation (e.g., between IEC 60870-5-104 and IEC 61850) [24], and, in the real system, it can be a dedicated protocol translator, e.g., [27], or a RTU (Remote Terminal Unit). Moreover, some of the gateway is equipped with VPN interface (e.g., SSH) so that engineers can perform maintenance remotely. Another type of gateway could also support tunneling of IEC protocols [28], which can be used when the primary wide-area network would be unavailable for some reason. The gateway may need to be deliberately configured to be vulnerable to attract attackers to the honeypot. For instance, we could configure VPN and/or SSH services with weak authentication credentials.

Owing to its exposure to external network as well as the vulnerable remote access interface, we assume that an attacker could have full control (and shell access) on the gateway. Thus, implementation of it requires careful consideration to avoid honeypot fingerprinting. To look like a real device, it should only run a representative set of processes that are essential for gateway features and should not run processes that can be easily associated with honeypots. For instance, if an attacker on the gateway would find processes corresponding to IEDs, which are supposed to be other nodes, he would immediately tell something is wrong. To accomplish this goal, we decide to use a dedicated virtual machine (VM) for each gateway. This also allows the gateway to have more flexibility to be customized to imitate a real gateway device in terms of OS, memory, storage size etc. For our proof-of-concept implementation, we use VirtualBox [6] for host virtualization. We could instead employ a real gateway device or gateway implementation on low-cost embedded platform like Raspberry Pi (e.g., one shown in [8]) for even better realism. The gateway is connected with substation LAN, which will be explained later, via a network interface of the VM.

Finally, a gateway should look like a real IEC 60870-5-104 device from the wide-area network. We use OpenMUC [29], a popular open-source tool, for implementing IEC 60870-5-104 interface, and only open minimal ports. (We studied ZNX202 [27], a commercial protocol translator designed for small-scale substations, by using Nmap [26], which found that it opens SSH ports besides port 2404 used by IEC 60870-5-104. Although Nmap identified some other non-standard ports opened, we excluded them for the sake of generality since our intention is not to emulate this specific device.)

B. Substation Network Emulation

Within substation LAN behind the substation gateway, we need to emulate a network of IEDs that are organized in a realistic topology. In large-size substations, there may be over 200 IEDs, so network of this size should be emulated. We decided to use Mininet [7], a widely-used network emulation tool. As seen in Fig. 2, Mininet is hosted in a different VM from the one for the gateway. The Mininet network is configured with several virtual layer-2 switches ("vSwitch" in the figure) connected in a ring topology with several Mininet virtual hosts ("vHost" in the figure) that serves as "virtual" IEDs. The Spanning Tree Protocol (STP) employed by the switches prevents routing loops and also determines the direction of the packet flow in the network. The bandwidth of links are set to be 100 Mbps to mimic the typical configuration of a substation network. Mininet supports network connectivity between the virtual switches and the network interface provided by the VM, which allows the virtual nodes on Mininet to communicate with external devices such as the substation gateway.

Mininet virtual hosts present themselves as IED devices in a substation. These virtual IEDs must be able to receive and respond to IEC 61850 MMS messages. We outsource essential IED functionality to SoftGrid [8] run on a separate machine, which will be discussed next. By leveraging IEDs provided

by SoftGrid, we can significantly simplify the logic to be implemented on Mininet virtual hosts. Specifically, we use *SOCAT*, a light-weight port-forwarder application, to forward all the packets received at port 102 on each Mininet virtual host (for IEC 61850 MMS) to SoftGrid via a secondary network interface that belongs to another subnet. We assume that an attacker cannot gain access to IED devices via network. We think this assumption holds in practice since commercial RTU devices we evaluated only offer limited configuration interface that requires a cable connected to a dedicated port on the board. Thus, an attacker can only interact with IEDs through the standard SCADA protocols (namely IEC 61850 MMS in our prototype) or can just passively monitor network traffic sent to or from them. Note that, under this assumption, the port-forwarding mechanism run on virtual IEDs and network traffic forwarded to SoftGrid are not visible to the attacker on the substation gateway.

Regarding the network among the control center and substations, we implement star topology. It is feasible to utilize other topology, e.g., ring topology, by using another Mininet.

C. Cyber-connected Simulation of Power Grid

To implement a simulated power grid as well as its connectivity to the network, we use SoftGrid, an open-source smart grid testbed [8]. SoftGrid's IED module is used to emulate IEDs in substations. Such IEDs are associated with physical power system components in a power grid, e,g., circuit breakers and transformers, that are simulated on the PowerWorld simulator [15] and are responsible for retrieving status of each device and other measurements as well as for exercising control according to the received IEC 61850 MMS messages. In our design, all virtual IEDs in all honeypot substations are connected to the same power flow simulation. In other words, while there can be multiple machines running honeypot substations, all of them are connected to a single machine running SoftGrid and PowerWorld. In this way, the entire honeynet shares a consistent view of the power grid. SoftGrid [8] can handle a large scale power grid, such as 2,000-bus system designed based on power grid in Texas, which includes over 1,500 substations [30] as well as 7,000 simulated IEDs. Even with this scale, the steady-state power flow simulation can be updated every 20ms on PowerWorld [8]. Concerning round-trip communication latency, the average is below 500ms when over 500 interrogation commands per second are processed [8], which satisfies the guideline by IEEE Power and Energy Society (PES) [31]. Thus SoftGrid will not be a performance bottleneck.

Changes in power flow and voltage reach new steady state within short time, namely less than 1 second based on our preliminary experiment using PowerWorld [15], when a minor control (e.g., opening a line with small power flow) is done on the grid, which is usually the case when an attacker is probing. Thus, steady-state simulation can practically approximate such changes in terms of realism. On the other hand, frequency change could take longer, e.g., in the order of 10 seconds or even minute, to stabilize. To incorporate such transient state, we modified SoftGrid so that transient-state simulation is run in parallel whenever control commands are received, to provide probing attackers with realistic frequency fluctuation. Therefore, our honeynet backed by SoftGrid can support utility-scale power grid with high fidelity and realism.

V. Evaluation

This section provides qualitative and/or quantitative evaluation based on the requirements discussed in Section III-C.

A. Comprehensive, Consistent Power Grid View

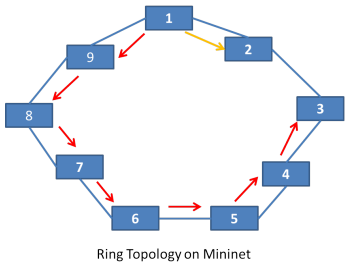
In our design, all the honeypot substations (and IEDs in them) are connected to the same power flow simulation. Moreover, SoftGrid supports real-time update of status of physical components on the simulated grid and re-calculation of steady-state power flow [8], which means that control commands received by IEDs are executed in a real-time manner and shortly after that the change is made visible.

Besides, as discussed in Section IV-C, we utilize transient-state simulation for generating frequency information. Based on our measurement, transient-state simulation in PowerWorld offers over 40x speedup. In other words, when a control command is executed, simulated frequency change for the next 1 minute can be made available within 1.5 second. Although this implies that, if an attacker sends interrogation before the simulation is done, the data may not be ready yet. However, when remote control is performed on a real-world power grid, there is a variety of inherent delay involved (e.g., time for operation, communication, and actuation, etc.). Thus, we believe a simulation latency of this level is sufficiently short to retain realism against *grid-wide probing* attacks, in which the attacker changes settings in some substations and examines the expected changes in other substations. Using the control center module in SoftGrid [8], We have carefully measured all the delay components involved for our honeynet to respond to such probing attacks and confirm that our honeynet design and implementation can stand such grid-wide probing attacks. If faster response is desired (e.g., for more advanced attackers), we could either shorten the simulation duration of the transient behavior (e.g., only simulate the next 30 seconds), or use simplified, equivalent model for reducing complexity. By doing so, the required computation time could be further lowered to the order of 100ms.

B. Realistic Network Configuration

While spoofing MAC addresses to make them look realistic (i.e., belonging to certain device vendors) is simple, getting realistic IP addresses from a power grid operator or utility company is difficult in practice. However, according to [17], use of Amazon IP address did not increase the likelihood that the system is detected as honeypot by Shodan [22]. [14] also states that Conpot deployed on Amazon EC2 still captured significant amount of ICS traffic.

Regarding other network characteristics, such as packet loss ratio and bandwidth, we can flexibly configure them on Mininet. In the current implementation, as explained in



Switch ID	Latency[ms]
2	0.041737
3	0.073851
4	0.069002
5	0.065155
6	0.059757
7	0.056181
8	0.049224
9	0.044994

Fig. 3. Emulated Ring Topology on Mininet and *ping* Latency Measured

Section IV-B, the bandwidth of communication line is set to 100 Mbps, which corresponds to typical LAN setting in substations, and based on our experiments, the measured bandwidth is reasonably close to it (around 80 Mbps). These parameters can be easily tuned according to actual production environment. Besides, by using OpenMUC [29] and SoftGrid [8], our honeynet supports IEC 60870-5-104 for communication between the control center and honeypot substations and IEC 61850 MMS within the substations. Therefore, an attacker on a honeypot gateway will see these incoming and outgoing traffic. We chose ring topology in substation since it is most frequently used. An attacker, as part of his probing, may attempt to identify topology by measuring network latency among nodes. Based on our experiment using *ping*, we confirmed that the latency well reflects the topology and therefore can convince the attacker (Fig. 3). We also did experiment with different number of virtual IEDs on a single Mininet (i.e., a honeypot substation). Based on our measurements on a desktop PC a quad-core desktop PC with 6GB RAM, each instance can accommodate 200 IEDs, which is the typical size in large-scale substations, although running 400 nodes exhibited a noticeable degradation (Table I).

TABLE I
ping LATENCY WITH DIFFERENT # OF VIRTUAL IEDS

# of Virtual IED	Avg. Latency (ms)	# Std. Dev.
50	0.04363	0.01844
200	0.04994	0.02007
400	0.05987	0.01781

C. Fingerprinting Resistance

We first utilized a widely-used network scanning and fingerprinting tool, Nmap [26], against our virtual IEDs run on Mininet to see if the tool can fingerprint them. In practice, an attacker is expected to use such a tool to determine the type and the operating system loaded on the devices of interest. If virtual IEDs are fingerprinted as devices running Windows, for instance, an attacker's suspicion about the system would be triggered. Based on our experiments, the default Nmap was not able to identify OS running on virtual IEDs on Mininet (i.e., seen as devices running uncommon OS). Regarding fingerprinting against a substation gateway, our implementation is detected as a Linux (2.6.32) device listening at port 2404, well imitating real commercial products such as ZNX202 [27]. Our claim is that our implementations are not

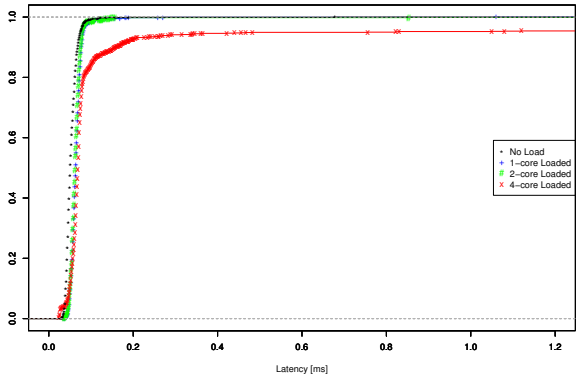


Fig. 4. Empirical Cumulative Distribution Function of *ping* Latency with Different Virtual Machine Settings

immediately suspicious from attackers' perspective, not that we offer perfect camouflage, which is still an open problem.

Another potential factor that could be used for honeypot fingerprinting is the use of virtualization. In particular, if one physical machine hosts multiple VMs running honeypot instances for better scalability, lack of sufficient isolation among instances would give attackers some clue. To evaluate the isolation, we set up two VMs, one of which is allocated one processor core and runs Mininet with 200 IEDs to be evaluated (*VM under Test*) while the other VM with 1,2, or 4 cores with high CPU load (*Load VM*). The result is seen in Fig. 4. The plot labeled "No Load" can be considered as a base line network latency that is considered similar to the real network. For the experiment, We used a PC with 4 cores, which means that allocating 4 cores to Load VM forces the core allocated to VM under Test to be shared with Load VM. As can be seen, if the core is shared, performance of Mininet on VM under Test was affected significantly and average was brought up to 18ms. We also observed noticeable deterioration in bandwidth of emulated network in this setting. These facts may be used for honeypot fingerprinting and therefore should be avoided. On the other hand, when honeypot substation instances did not share the core, we did not observe degradation in performance. In other words, even when one instance is of high load, performance of another (e.g., network latency) is not affected. Another possible deployment option for scalability would be to host multiple Mininet instances within a single VM, which is allocated a dedicated processor core. In this case, dependency among the Mininet instances was negligible (Fig. 5).

D. Scalability and Operational Cost

In the previous section, our finding is that, in order to retain sufficient isolation, we should allocate one physical processor core to each VM. We also found that a single VM can safely host multiple Mininet instances. The number of Mininet instances may be bounded by the hypervisor used. For instance, on VirtualBox, the number of external network interfaces per VM is limited up to 8. Since in our design each Mininet needs to have two interfaces (Fig. 2), a single VM can host up to 4 Mininet instances. Thus, in total, if we use a commodity quad-core PC, we can technically run up to 16

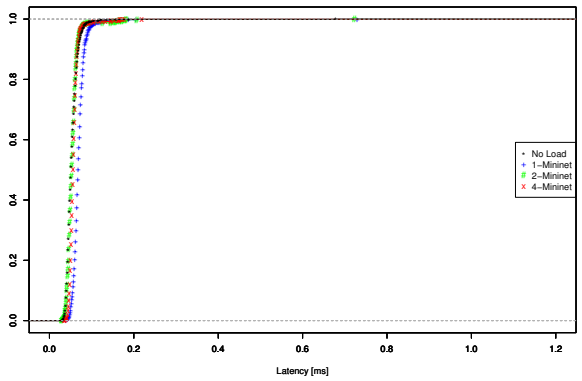


Fig. 5. Empirical Cumulative Distribution Function of *ping* Latency with Different # of Mininet Instances per VM

Mininet instances (i.e., honeypot substation LANs), and it is sufficient to emulate small-scale smart grid systems. Moreover, because lower-voltage substations are often controlled by (or via) high-voltage stations, even this number of high-voltage substations would suffice to emulate Singapore-scale grid.

If we consider further scale-up, for instance, National Cybersecurity R&D Lab (NCL) [9] offers 12-core node with 64GB RAM for US$0.09 per hour. We deployed 6 VM hosting three Mininet instances (with 1 core and 1GB RAM) along with 18 VMs (with 1 core with 50% execution cap and 1GB RAM) running gateway for each Mininet instance (i.e., 18 honeypot substations) on a physical node in NCL. During the experiment continuously sending decent amount of IEC 60870-5-104 commands to these honeypot substations, the overall CPU usage of the node did not exceed 30%, with short interrogation response time (on average 140ms), which implies we can practically run over 36 substations per node. Based on this number, assuming use of NCL, conservatively estimated cost to continuously operate a honeynet with 200 and 1,500 substations, the latter which is equivalent to Texas-2000 bus case [30], are around $390 and $2,700 per month respectively.

VI. Conclusions and Future Work

In this paper, we presented the design and implementation of a smart grid honeynet. The honeynet can offer comprehensive, high-fidelity view of power grid and communication infrastructure to attackers and therefore can potentially retain them for a long time to study their strategies as well as to buy time before they initiate real attacks. Our honeynet can be constructed largely based on open-source tools, so it can be quickly tried and flexibly configured. We plan to open-source our implementation as extension of SoftGrid [8]. Although our effort is in the early stage, we hope our design will be used as foundation to develop further enhanced, customized systems.

As a future work, we will expose the smart grid honeynet to collect real-world attack vectors. Besides, we are planning to work on extensions including the followings: integration of physical latency models for countering latency-based fingerprinting, enhancement of virtual IEDs for generating realistic SCADA traffic (e.g., IEC 61850 GOOSE and SMV) within substations, and so forth.

References

[1] K. Zetter, "Inside the cunning, unprecedented hack of ukraines power grid," http://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/.

[2] D. U. Case, "Analysis of the cyber attack on the ukrainian power grid," 2016.

[3] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, "A survey on honeypot software and data analysis," *arXiv preprint arXiv:1608.06249*, 2016.

[4] K. Koltys and R. Gajewski, "Shape: A honeypot for electric power substation," *Journal of Telecommunications and Information Technology*, no. 4, p. 37, 2015.

[5] D. Antonioli, A. Agrawal, and N. O. Tippenhauer, "Towards high-interaction virtual ics honeypots-in-a-box," in *CPS-SPC Workshop*. ACM, 2016, pp. 13–22.

[6] "Virtualbox," https://www.virtualbox.org/.

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

[8] P. Gunathilaka, D. Mashima, and B. Chen, "Softgrid: A software-based smart grid testbed for evaluating substation cybersecurity solutions," in *CPS-SPC Workshop*. ACM, 2016, pp. 113–124.

[9] "National cyber security r&d lab," https://ncl.sg/.

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

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

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

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

[14] 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 (PST16)*, 2016.

[15] "PowerWorld," http://www.powerworld.com/.

[16] D. Antonioli and N. O. Tippenhauer, "Minicps: A toolkit for security research on cps networks," in *CPS-SPC Workshop*. ACM, 2015, pp. 91–100.

[17] 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.

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

[19] P. Maynard, K. McLaughlin, and B. Haberler, "Towards understanding man-in-the-middle attacks on iec 60870-5-104 scada networks," in *Proceedings of the 2nd International Symposium on ICS & SCADA Cyber Security Research 2014*. BCS, 2014, pp. 30–42.

[20] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, E. G. Im, Z. Yao, B. Pranggono, and H. Wang, "Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in smart grid scada systems," in *SUPERGEN 2012*. IET, 2012, pp. 1–8.

[21] D. Perez and J. Pico, "A practical attack against gprs/edge/umts/hspa mobile data communications," *Black Hat DC*, 2011.

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

[23] 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.

[24] D. Mashima, P. Gunathilaka, and B. Chen, "An active command mediation approach for securing remote control interface of substations," in *IEEE smartGridComm 2016*. IEEE, 2016.

[25] I. Y. Lee and N. K. Chan, "Advanced power quality monitoring system in hong kong," in *The 16th Annual PQSynergy International Conference and Exhibition*, 2016.

[26] "Nmap.org," https://nmap.org/.

[27] "ZNX 202," http://biz.co188.com/content_product_63460696.html.

[28] "Tofino firewall lsm," https://www.tofinosecurity.com/products/Tofino-Firewall-LSM.

[29] "OpenMUC," https://www.openmuc.org.

[30] "Texas 2000-june 2016," http://icseg.iti.illinois.edu/synthetic-power-cases/texas2000-june2016/.

[31] IEEE Power and Energy Society, "IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation."