Taxonomy of Fingerprinting Techniques for Evaluation of Smart Grid Honeypot Realism

Vanessa Tay, Xinran Li
National University of Singapore
Singapore
{vanessa_tay, xinran.l}@u.nus.edu

Daisuke Mashima, Bennet Ng
Illinois Advanced Research Center at Singapore
Singapore
{daisuke.m,bennet.ng}@iarcs-create.edu.sg

Phuong Cao
National Center for Supercomputing Applications
University of Illinois Urbana-Champaign
pcao3@illinois.edu

Zbigniew Kalbarczyk, Ravishankar K Iyer
University of Illinois Urbana-Champaign
Champaign, Illinois
{kalbarcz,rkiyer}@illinois.edu

***Abstract*—Honeypots are a form of deception technology that provides an additional line of defense, and is regarded as a cybersecurity application of digital twins in the smart grid domain. By misleading attackers into a decoy and thereafter, performing threat intelligence collection and analysis, honeypots allow operators time to conceptualize mitigation strategies. In that regard, the most important property of a honeypot is realism from the attackers' perspective, but the task of imitating the real system remains non-trivial, especially for smart grids which encompass immensely intricate infrastructures. The absence of an established way to guide the design of, or assess the effectiveness of smart grid honeypots, further compounds the problem. To supplement such research gaps, this paper first surveys existing literature on honeypot detection strategies, and thereafter, delineates a taxonomy of fingerprinting techniques geared towards smart grids. Such a taxonomy can be used to judge the realism of smart grid honeypots, and this paper demonstrates relevant evaluation applications after discussing our own implementation of a comprehensive smart grid honeypot. In essence, the aforementioned efforts are made to elucidate varied dimensions of smart grid honeypots' realism and thereby provide an effective guide for the design of smart grid honeypots that are robust against fingerprinting.**

***Index Terms*—Cybersecurity, Smart Grid, Honeypot**

I. Introduction

Honeypots have proven themselves as potent cybersecurity tools for the early detection of security threats in a broad spectrum of systems and devices. By misleading and trapping malicious actors in a dummy environment, honeypots buy time for the preparation of defense measures, which include the analysis of attackers' profiles and tailoring of security mechanisms like firewalls and intrusion detection systems.

The dawn of the new millennium saw a push for worldwide interconnectivity in which ICS is increasingly exposed to the Internet, with approximately 70,000 of them detected in 2021 [1]. A notable precedent would be smart grids, which have evolved from electrical power grids and are now one of the main targets of ICS attackers [2]. Adopting advanced computerized mechanisms, smart grids are large-scale distributed systems more efficient at electricity distribution than their traditional counterparts. Such increased digitization, however, gives rise to a broad and complicated attack surface [3] that is difficult to defend. On that note, honeypots would be valuable in allowing malicious actors, themselves, to reveal unseen and potentially novel attack vectors on which operators could base their defense tactics and better secure the inherently convoluted smart grids. In the smart grid domain, honeypots are essentially straightforward cybersecurity applications of digital twins.

As the value of honeypots stands in the quantity and quality (novelty) of received attacks [4], they must be designed with a high level of realism and are essentially indistinguishable from actual systems. However, due to the emergence of new honeypot fingerprinting methodologies and the lack of extensive studies in the smart grid honeypot domain, developing a realistic smart grid honeypot is, by all means, not a trivial task. In addition, as far as our knowledge goes, there is an absence of a systematic framework that guides the construction of smart grid honeypots. At present, a typical and conventional strategy used to evaluate the goodness of honeypots is ad-hoc and empirical (i.e., deploy the honeypot, observe how well it attracts access attempts, make improvements iteratively). Such an approach is, however, not the best option as a honeypot experiment is ideally an one-shot attempt to be completed before they are identified as a decoy (i.e., the honeypot should be used only once for its full effectiveness). From the perspective of developers, therefore, the qualitative evaluation of honeypot designs prior to real-world deployment is often desired.

In this paper, we first define a taxonomy of smart grid honeypot fingerprinting techniques that could be used by cyber attackers to distinguish honeypots from real systems, through a survey of relevant literature. Then, robustness against the enumerated fingerprinting tactics in out taxonomy are utilized as the criteria for qualitative and comparative assessment of existing smart grid honeypot implementations, including our own smart grid honeypot prototype. In the process, we venture to demonstrate tangible applications of our taxonomy and elucidate certain design considerations that existing honeypot implementations need to account for. All in all, with the enumeration of fingerprinting and anti-fingerprinting strategies, along with a review of existing implementations, this paper guides the construction of highly deceptive (i.e., realistic) smart grid honeypots.

II. Related Work

Existing classification schemes for the fingerprinting and anti-fingerprinting strategies of decoy environments do not address the unique specificities of smart grid honeypot systems. Representative efforts are summarized below [5]–[8].

Zamiri et al. [5] group ICS honeypot fingerprinting methods into four overly simple and broad categories - *Default Configuration*, *Missing Protocol Features*, *Unusual Behavior*, and *Fingerprinting the Underlying Platform*. For instance, the proposed *Unusual Behavior* category could comprise methods related to system processes, service functions, or communication workflows. Distinctions could be made for a more accessible and clear-cut classification. Gajrani et al. [6] introduce a framework that classifies techniques used to identify emulated environments into twelve fine-grained categories. Conceptualized with a focus on mobile environments, the categories, such as *Phone ID* and *Device Build*, are not directly applicable to general IT honeypots, not to mention smart grid-specific ones that are built on large-scale, distributed network environments. Chen et al. [7] propose a taxonomy that categorizes anti-virtualization and anti-debugging strategies into four high-level categories - *Hardware, Environment, Application* and *Behavioral*, each comprising a maximum of two subcategories. While relevant to general honeypot environments, the categories are too broad and wide-sweeping to be actionable. For instance, the suggested *Behavioral* category is standalone, revolving around latency checks, and could be further sub-categorized into network-wide and localized domains for increased granularity. Building upon the taxonomies defined in [6] and [7], Uitto et al. [8] present one that is two-tiered with four high-level categories - *Temporal, Operational, Hardware* and *Environment*. Altogether, comprising a larger number of subcategories, Uitto et al. suggest that their taxonomy is less abstract than the one in [7] and better tailored to general honeypot environments than the one in [6].

III. Taxonomy of Smart Grid Honeypot Fingerprinting Tactics

In this section, we define a taxonomy of smart grid honeypot fingerprinting techniques based on the taxonomy proposed by Uitto et al. [8], as discussed in Section II. Among existing efforts in the literature, Uitto et al.'s taxonomy is the most comprehensive and applicable for our adaptation as it is sufficiently detailed and relevant to the current security landscape, taking into consideration malware evolution. For instance, the *Operational: Propagation* subcategory checks if propagation of botnet infections are allowed, thereby accounting for the rise of botnet attacks. Nevertheless, changes to Uitto et al.'s taxonomy are still necessitated for increased smart grid relevance.

A notable change includes the exclusion of the following four subcategories. Firstly, the *Device* & *Driver* subcategories of Hardware, along with the *Memory* subcategory of Environment fingerprint honeypots by detecting the presence of Virtual Machines (VMs). VM identification is a universal challenge faced by honeypots that make use of virtualization technologies, and while we acknowledge that VM detection strategies need to be addressed, this paper is concentrated on smart grid specificities. We will not delve into VM fingerprinting which is in itself a broad, large-scale study and has been extensively researched upon [9]. Next, the *Propagation* subcategory identifies honeypots based on their incapability to propagate malware. This is more of a liability issue, as operators should not allow their honeypots to spread viruses (e.g., as part of a botnet). However, from our perspective, for the sake of security, malware propagation to external machines must be prevented regardless of the system's identity (as a honeypot or otherwise). Moreover, as smart grids enforce stringent security requirements over traffic flows [10], it is reasonable for the movement of malware to be blocked by edge firewalls or data diodes, possibly aided by threat detection and mitigation strategies [11], [12]. We removed the *Propagation* subcategory because of the mentioned reasons.

Additional modifications are also introduced, with the conception of new categories and subcategories, and the merging of certain existing ones. The taxonomy of smart grid honeypot fingerprinting tactics defined by us is summarized in Table I and is detailed in the rest of this section.

A. Structure

Structure, a proposed category addition, fingerprints smart grid honeypots by identifying inappropriate compositions, and comprises two subcategories: *Layout* and *Components*. Smart grids are large complex systems, and lackluster attempts in the structural emulation process should be weeded out.

The *Layout* subcategory checks for overly simple network layouts. Going into specifics, a smart grid infrastructure typically consists of a control center and substations at the transmission and distribution levels, which are connected via WAN. Narrowing down, devices within a substation are often arranged in a ring topology for higher redundancy and tend to be segregated into multiple levels, namely the station level and the bay level [14]. Altogether, high-level arrangements are often done in adherence to the Purdue Model [17], which has served as a reference to guide ICS network segmentation since the early 1990s. In any case, the absence of such layout characteristics could be identified through the *'traceroute'* command in the probing of possible paths, thereafter presented in a graph setting for topology reconstruction [13].

The *Components* subcategory checks for the absence of common smart grid components. Smart grids typically comprise a SCADA system, which necessitates the inclusion of an HMI for system-wide monitoring, in turn, connected to a historian database for the logging of power grid measurements. Devices like IEDs are also necessary for the execution of automated control/protection functions, and PLCs could be included for higher-level coordinated monitoring and control (e.g., [18]). Adding on, other elements like local SCADA/substation gateways and RTUs are often found in actual smart grids for better communication management. Undeniably, there are variations in the choice of components present in actual smart grids, but, regardless, unjustified omissions of conventional components would invoke suspicion and could be detected through passive traffic and device monitoring tools like Nmap and Wireshark.

B. Temporal

Guided by communication standards detailed in [19], [20], this category fingerprints smart grid honeypots through sig-

TABLE I: Taxonomy of Smart Grid Honeypot Fingerprinting Techniques.

Category	Subcategory	Smart Grid Honeypot Fingerprinting Techniques
Structure	**Layout**	○ Topology discovery via tools like *'traceroute'* [13], [14]
	Components	○ Identification of machines (Historian, VPN server, etc.) via Nmap/Wireshark
Temporal	Network	○ Measurement of link latencies (within and across substations) via *'ping'* ○ Measurement of SCADA communication (IEC 60870-5-104, IEC 61850 etc.) network latencies via Wireshark/ping-pong method [15]
	Local	○ Examination of system call timings via clock cycle analysis [16] ○ Measurement of SCADA communication (IEC 60870-5-104, IEC 61850 etc.) periodicities via Wireshark ○ Measurement of internal processing delays of smart grid devices via ping-pong method [15]
Operational	Communication	○ Examination of traffic flows and protocol types via Wireshark ○ Probing of firewall permissions via *'traceroute'*
	Operation	○ Examination of operation responses against smart grid specifications/power grid status ○ Testing for the implementation of relevant network services ○ Testing for known vulnerabilities of ICS devices as detailed in the CVE/CVSS database
	Idiosyncrasies	○ Testing of basic operational functionalities of smart grid devices
Hardware	**Identity**	○ Examination of smart grid devices' MAC addresses against ICS vendors' specifications
Environment	Data	○ Examination of system processes (e.g., via *'ps aux'*) ○ Examination of system software/files (e.g., via *'apt list –installed'* or *'ls'*) ○ Examination of smart grid devices' OS fingerprints via Shodan/Nmap
	Mode of Exposure	○ Examination of exposed public IP addresses via Shodan ○ Examination of exposed ports (and their services) via Shodan/Nmap ○ Discovery of remote access pathways via *'traceroute'* [13] ○ Testing of access authentication mechanisms with common/default credentials
Cyber-Physical	**Integration**	○ Testing of cyber-physical consistency (e.g., by manually opening a circuit breaker)

Note: Categories/subcategories with a greyed background indicate the proposed additions.

nificant deviations of established timing specifications, and comprises two subcategories: *Network* and *Local*. It would be inefficient to fully enumerate the multitude of timing standards, so the more important and easily detected ones are discussed.

The *Network* subcategory checks for significant delays in the network environment, which could be caused by the deployment of multiple large-scale honeypots on a single host. In this regard, the one-way delivery time of control packets (e.g., ICMP ECHO packets) in smart grid honeypots should not exceed 16 ms for transmissions confined to a substation LAN and 1 s for transmissions across substations [19]. Additionally, network latency of communication protocols need to be realistic. For instance, the RTT of SCADA interrogation/poll messages should not exceed 200 ms [19], and the network transfer time of time-critical messages for protection functions between IEDs should not exceed 600 µs (IEC 61850 standard) [20]. While Wireshark would suffice for most timing measurements, the ping-pong method [15] could be used by attackers for more accurate determination of IEC 61850 GOOSE latencies.

The *Local* subcategory checks for significant deviations in localized timing specifications. For smart grid systems, localized timing checks would also refer to the periodicity of SCADA transmissions and the internal processing latencies of smart grid devices. With regards to the former, SCADA interrogation/poll messages should be sent by the HMI every 1 to 10 s [19], and the GOOSE protocol should employ a reasonable transmission interval (e.g. 1 s) that shortens when the power grid status changes, to ensure timely information delivery. Next, with regards to the latter, in a typical time-sensitive protection workflow, each IEC 61850 stack processing instance in IEDs should not exceed 1.2 ms (IEC 61850 standard). Measurement methods are as given in the *Network* subcategory.

C. Operational

Attackers may attempt to fingerprint smart grid honeypots by identifying the presence of unrealistic services. Tactics of this category comprises three subcategories: *Communication*, *Operation*, and *Idiosyncrasies*.

The *Communication* subcategory checks for unusual communication and traffic flows. For example, some smart grid honeypots rely on proprietary protocols for coordination among the system components. Traffic associated with such protocols, which do not exist in actual smart grids, can be detected through Wireshark and must remain concealed from external actors. In the same vein, the absence of common smart grid communication traffic (IEC 104, IEC 61850, etc.) would trigger suspicion, and so will the omission of reasonable firewall/traffic flow control which could be probed through *'traceroute'*.

The *Operation* subcategory checks for inadequate service emulations. Firstly, service responses that are inconsistent with smart grid specifications and power grid status are tell-tale signs. This, for instance, refers to SSH banners that are not ICS-relevant, or static MMS responses (as power grid measurements should change over time). Next, the absence of relevant network services, like the typical SSH on protocol translation gateways for remote configuration purposes, is also indicative of honeypots. In addition, where present in the CVE/CVSS database, vulnerabilities in smart grid devices should also be emulated as attackers may test for them.

The *Idiosyncrasies* subcategory checks for peculiarities in the execution environments of smart grid devices. To further elabo-

rate, this refers to abnormalities in the internal sub-components of devices, such that the realization of basic functions is not ensured. For instance, when an IED detects anomalous power grid measurements, it should instantly trigger circuit breaker control functions to prevent severe damage to the power grid equipment, which is the most fundamental of tasks. The absence of relevant changes to the physical grid would suggest an irregularity in the IED's execution environment (e.g., relays, regulators), which is a sign of a honeypot.

D. Hardware

Attackers may examine hardware-related oddities and comprises a single subcategory: *Identity*, which fingerprints smart grid honeypots by recognizing inappropriate hardware identity values. For instance, the MAC addresses of smart grid devices (e.g., IEDs, PLCs, RTUs) should align with the hardware specifications of actual ICS vendors, which are publicly available information. Note that this category originally focused on identifying the usage of specific VMs (e.g., VMware) through hardware information. However, as mentioned, we are not going deep into VM detection methods and have thus re-framed this category with a smart grid-specific focus for our purpose.

E. Environment

Unrealistic environment characteristics can be exploited for honeypot fingerprinting. Techniques in this category comprises two subcategories: *Data* and the *Mode of Exposure*.

The *Data* subcategory checks for oddities in machine data (files, processes, OS). Firstly, the presence of suspicious data could be identified through simple BASH commands, such as *'ps aux'* that allows for the enumeration of running processes, or *'apt list –installed'* that lists the installed system software. On that note, logging-related data need to be hidden as they indicate a monitored system, which is characteristic of honeypots. Next, through passive and active OS fingerprinting tools like Nmap and Shodan, the OS fingerprints (e.g., TCP/IP stacks) of machines can be easily examined to ascertain their similarity to actual smart grid devices (e.g., IEDs, PLCs, RTUs), and should thus be tailored accordingly. Conversely, the absence of expected data is also fingerprintable. In this regard, machines should appear used with the presence of user accounts and the population of file directories with relevant data (e.g., VPN configuration files on VPN servers).

Mode of Exposure, a proposed subcategory addition, identifies inappropriate entry-point configurations and is relevant to smart grids, which impose stringent access requirements [10]. Notably, Shodan allows for the inspection of access nodes via their public IP addresses, and such nodes should thus be made as realistic as possible. On that point, as smart grids are usually hosted by organizations with power grid associations (e.g., utility companies, universities, large-scale industry sites, data center operators), their exposed IPs should not belong to public cloud platforms (e.g., AWS) or generic IT companies, despite the convenience in allowing so. The presence of large numbers of exposed ports that provide unnecessary/repetitive services is also indicative of attempts at driving honeypot traffic and should be avoided. Lastly, smart grids are not easy targets, and remote access mechanisms need to be secure. To further elaborate, in typical remote setups, only the VPN server is exposed to the Internet, after which a Jump Box will be used to facilitate internal access [21]. Similar pathways should be established in the honeypot while ensuring that VPN authentication is reasonably secure, such that default credentials are denied.

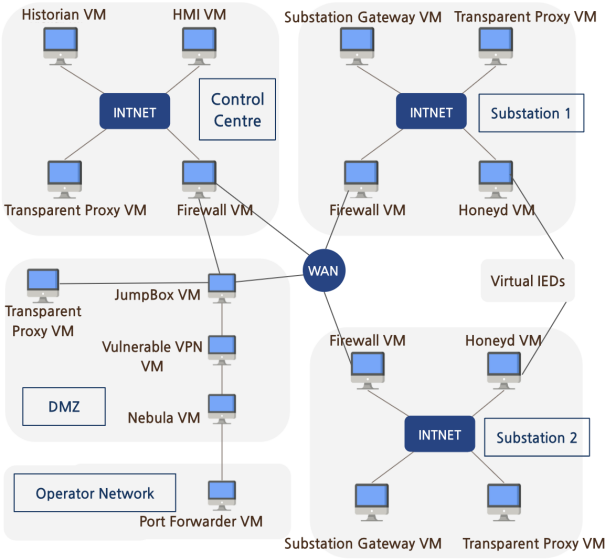

Fig. 1: Overall Honeypot Layout

F. Cyber-Physical

Cyber-Physical, a proposed category addition, comprises a single subcategory, *Integration*, that fingerprints incoherent cyber-physical integrations in smart grid honeypots. Here, consistency is crucial, especially with regard to the measurements and statuses in ICS messages exchanged between the cyber and physical ends. For instance, when a circuit breaker is opened, either by an attacker or as a result of IED protection functions, the corresponding circuit breaker status reported to other devices, along with the voltage/current measurements on the affected line or bus should be changed accordingly.

IV. High-Interaction Smart Grid Honeypot

In this section, we discuss our high-interaction smart grid honeypot prototype based on our earlier work [22]. The honeypot prototype will be used later for a case study to utilize the defined taxonomy of honeypot fingerprinting tactics for the qualitative evaluation of the realism. We note that the development of a honeypot that counters all the fingerprinting tactics discussed is not the goal of this paper.

A. Overall Architecture

The layout of our honeypot is given in Figure 1, in which the machines are run as VMs on dedicated physical hosts. Structured based on the Purdue Model [17], we layered our honeypot as follows - ***Layer 5:*** Public Internet, ***Layer 4:*** Operator Network, ***Layer 3:*** Control Centre, ***Layer 2:*** Substations 1 & 2, ***Layer 1:*** Virtual IEDs, ***Layer 0:*** Power Grid Simulator (back-end of Virtual IEDs).

B. Access Configurations

Akin to real-world setups, we exposed a VPN interface on our honeypot's entry point, i.e., the Port Forwarder VM. This VPN interface is part of our remote access pathway that also involves a Jump Box, which is a hardened device for managing internal network access. For minimal invasiveness and efficiency, the pathway is further augmented with a port forwarding mechanism and is integrated into a Nebula overlay network.

Firstly, port forwarding is enabled on the Port Forwarder VM which lies in the Operator Network, and is allocated an associated public IP address. Traffic directed to the Port Forwarder VM on port 443 is forwarded to the Vulnerable VPN VM (through the Nebula VM). This allows us to avoid intruding on the Operator Network, which belongs to an organization with power grid relations (that we wish to not disclose), such that there is less security risk on real infrastructures while leveraging the organization's reputation.

Next, we utilize Nebula to allow for a more efficient port forwarding process. Nebula is a highly scalable open-source mesh overlay networking tool that provides direct, encrypted connections between hosts. Deployed on the Port Forwarder VM and the Nebula VM, Nebula enables fast, straightforward communication between the two nodes that are located in different geographical regions. Additionally, as connections are encrypted and secure, the usage of Nebula minimizes the probability of our honeypot entry point being compromised.

Altogether, upon successful OpenVPN authentication at the Vulnerable VPN VM, the attacker will be able to access the JumpBox VM through RDP. From the JumpBox VM, the attacker can connect to the HMI VM / Historian VM within the Control Centre via RDP, or the Substation Gateway VM in either substation via SSH. Note that our OpenVPN service is authenticated with less straightforward, non-default credentials that can be brute-forced with the aid of open-source tools like Hydra. Otherwise, to enter the honeypot, the attacker could also leverage the Shellshock vulnerability (CVE-2014-6271) purposefully injected into the Vulnerable VPN VM, which is a non-trivial task.

C. SCADA System Configurations

SCADA is a core system in smart grids and our implementation of its key components is as follows.

a) HMI & Historian

The HMI and historian are vital SCADA components that allow for high-level management and monitoring. They are found in our Control Centre, namely, the HMI VM and Historian VM. The HMI VM runs OSHMI, a powerful and flexible HMI software, along with QTester104, an open-source client software that enables the periodic generation of SCADA interrogation messages (configured to 2 seconds) via the standard IEC 60870-5-104 protocol. Moving on, the Historian VM runs a database, TimescaleDB, that stores data collected by the HMI. Specifically, TimescaleDB offers the efficiency of PostgreSQL while allowing for higher ingest rates with respect to time-series data.

b) Substation Gateway (Protocol Translator)

The Substation Gateway VMs are crucial to the SCADA communication infrastructure. They serve as protocol translation gateways that translate between the IEC 60870-5-104 and IEC 61850 MMS protocols in the facilitation of communication between the HMI and IEDs. Additionally, akin to real gateways, we configured SSH on the Substation Gateway VMs, albeit in the form of an SSH honeypot, Cowrie. Even though Cowrie can be easily fingerprinted in its default state, a study conducted in [23] details the configuration changes that augment Cowrie into a highly realistic SSH service. Having modified our Cowrie setup accordingly, we can effectively avoid detection while reaping the benefits in terms of the ease of monitoring and better control over the exposed environment.

c) Virtual IEDs

Providing critical protection functions, IEDs are fundamental to the SCADA workflow. Our virtual IED setup is similar to the one in [24], but instead of employing Mininet, we use a Docker container to run each virtual IED for better resource efficiency. As done in [24], some notable features would include the support of the IEC61850 GOOSE protocol for status exchange between IEDs, and the IEC61850 MMS protocol for communication with upper-level devices (i.e., SCADA HMI in the HMI VM). HTTP and SSH services are also offered on our virtual IEDs and basic protection workflows are implemented, such that abnormal power grid measurements will result in the immediate invocation of control functions on the back-end power grid simulator (e.g., tripping of circuit breaker). Lastly, the front end of our virtual IEDs runs on Honeyd, which is an open-source host virtualization software deployed on our Honeyd VMs. Note that we configured Honeyd with OS fingerprints and MAC addresses taken from real IEDs.

D. Logging Configurations

Within both substations, the Control Centre and the DMZ, Transparent Proxy VMs are used to log attackers' activities (as observed in network traces). A transparent proxy is a simple and lightweight way of intercepting traffic between devices, and it remains undetected as it is not addressable by attackers and works as part of the network medium.

E. Internal Network Configurations

The Firewall VMs deployed in the Control Centre and both substations utilize OPNsense, an open-source firewall. We use the firewalls to allow for internal traffic flows on a least privilege basis, and the more essential permissions are summarized below.

The Firewall VM in the Control Centre allows RDP traffic (TCP port 3389) between the JumpBox VM and the Historian VM / HMI VM, as well as traffic flows involving the IEC 60870-5-104 protocol (TCP port 2404) between the HMI VM and Substation Gateway VMs. Next, the Firewall VM in either substation allows SSH traffic (TCP port 22) between the JumpBox VM and the Substation Gateway VM, along with the aforementioned IEC 60870-5-104 traffic flows.

F. Cyber-Physical Integrations

A power grid simulator, Pandapower, is used for realistic emulations of physical processes, such as the dynamic calculations of power system measurements. As done in [24], our virtual IEDs interact bidirectionally with the simulator via its database and are thus able to make changes to the simulator

model (e.g., opening or closing of circuit breakers) and obtain updated power grid measurements to send to other devices in real-time. Additionally, as Pandapower is centralized, all IEDs share the simulator outputs and can present a consistent power system view. However, high-fidelity simulation of transient-state behaviors is not provided.

V. Qualitative Evaluation of Smart Grid Honeypot

In this section, using the proposed taxonomy of fingerprinting techniques as the criteria, we present an evaluation case study. of existing smart grid honeypots in the literature as well as ours.

A. Evaluation of Our Smart Grid Honeypot

As discussed in Section IV, our honeypot's layered, hierarchical arrangement and secure internal/external traffic flows mimic those found in actual ICS networks, such that it does not appear to be an easy target. Through a meticulous setup process, we ensured the provision of a comprehensive SCADA system complete with the emulation of OS fingerprints and communication protocols. Along with the efficient use of underlying CPU resources, effective port forwarding functionality, and the adoption of lightweight, unintrusive monitoring processes, localized and system-wide latencies are also realistic and do not deviate from the established specifications. Additionally, the inclusion of Pandapower allows for cohesive cyber-physical interactions and makes for a high-fidelity smart grid honeypot.

However, there are some weaknesses that will be mitigated in future iterations. Firstly, while our virtual IEDs implement common protection functions (over/under voltage and current, differential protection, etc.), the list is not comprehensive. Adding on, the internals of our virtual IEDs are also not modeled after specific IED models, so model-specific vulnerabilities are not replicated. Next, although the lack of accessibility concerning our honeypot's physical processes is not a cause for concern, it could be improved for greater transparency and realism. On that note, we could expose additional services (e.g., port 102 for IEC 61850 MMS) on our access node for enhanced interactivity on the physical end. Finally, given that Pandapower is a steady-state simulator, it is unable to imitate the detailed transient-state dynamics of the physical system. This can be easily addressed by replacing Pandapower with another simulator, Matlab Simulink, at the expense of cost.

B. Comparative Study

Using the taxonomy defined in Section III, we contrast our honeypot implementation discussed in Section IV against some existing designs in the literature [25]–[27]. Their realism and robustness have not been systematically evaluated against honeypot fingerprinting technologies.

Gridpot [25] is a honeypot that emulates electric grids. While it integrates a power system simulator to provide a consistent cyber-physical view, usage of the well-known Conpot results in inherent limitations in terms of identifiable service signatures that are easily exploitable (e.g., the combination of exposed ports detected by Shodan Honeyscore). Adding on, Gridpot does not offer sufficient architectural emulations. Though it allows for more comprehensive component inclusions, it caters primarily to IEDs. Also, unlike the conceptual specifications, actual GridPot deployments are demonstrably low-fidelity with incohesive component integration and flaws in the basic IED functionalities [28], [29].

SHaPe [26] is a low-fidelity implementation, emulating any IEC 61850-compliant IED through ICD/IID configuration files, which (implicitly) prompts for the tailoring of hardware information. Owing to its standalone nature, SHaPe does not provide structural emulations. Moreover, it does not implement all protocols/services provided by typical IEDs (e.g., IEC 61850 GOOSE), while also omitting the emulation of protocol stacks and physical processes. A module of Dionaea, the "inherited" exposed services (e.g., HTTP, FTP) in SHaPe also lack depth.

CryPLH [27] specifically emulates Siemens Simatic 300(1) PLCs. It has comprehensive protocol imitations and realistic port exposures, two of which could be used to connect the PLCs in a plausible topology, thus partly catering to structural emulations. However, TCP/IP stack emulation is incomplete, interactive physical processes/services are absent, and the exposed web interface is flawed as it denies all login attempts and does not reproduce a known vulnerability.

The results of the comparative study are summarized in Table II. Having addressed a majority of the fingerprinting criteria, our honeypot's deceptiveness has an edge over the compared implementations, and our design features are thus demonstrably effective anti-honeypot strategies (though improvements, as detailed in Section V-A, are still in progress). Across the different honeypots, it can also be observed that the *Operational: Operation*, *Environment: Mode of Exposure*, and *Cyber-Physical: Integration* subcategories are partially addressed at best. Future research in smart grid honeypots should thus be focused on those three aspects that are more easily overlooked.

VI. Conclusions

While the value of honeypots is recognized and several efforts have been made in the smart grid aspect, there is no established way to evaluate the realism of smart grid honeypot implementations. In this paper, we have presented a taxonomy of smart grid honeypot fingerprinting techniques that can be used to qualitatively evaluate the realism of smart grid honeypots. Usage of the taxonomy has been demonstrated through an evaluation case study on our smart grid honeypot prototype alongside others in the literature, which highlighted the areas for improvement in the studied implementations. Supplementing existing research gaps, we believe that our taxonomy is useful in guiding the development of highly realistic smart grid honeypots. While we focused on honeypots, we envision that the same taxonomy is beneficial also for guiding the design and implementation of smart grid cyber range other virtual testbeds.

We do not claim that our taxonomy is conclusive or comprehensive in this version, and emerging technologies and/or categories may need to be added continually. Having said that, to our knowledge, this paper demonstrates the first attempt at defining a framework for the systematic evaluation of smart grid honeypot realism, which had been an elusive concept that led to assessments being conducted in ad-hoc manners. In future work, we plan to further enhance the taxonomy through empirical study. As attackers interact with our deployed

TABLE II: Comparison of Smart Grid Honeypot Implementations.

	Structure		Temporal		Operational			Hardware	Environment		Cyber-Physical
	Layout	Comp	Network	Local	Comm	Operation	Idio	Identity	Data	MoE	Integration
Gridpot [25]	○	◉	○	○	◉	◉	○	○	○	○	◉
SHaPe [26]	○	○	○	○	◉	◉	●	●	○	○	○
CryPLH [27]	◉	○	○	○	●	◉	●	●	○	◉	○
Honeypot in Section IV	●	●	●	●	●	◉	●	●	●	◉	◉

○: Not Addressed, ◉: Partially Addressed, ●: Addressed

honeypot over an extended period, we would be better equipped to identify previously unnoticed fingerprinting methods through traffic analysis, and thereafter, refine both our honeypot and taxonomy in a positive feedback loop.

VII. Acknowledgement

This research is supported in part by the National Research Foundation, Singapore (through the National Cybersecurity R&D Lab grant office at the National University of Singapore) via a grant (Grant Award NCL-2022-01) awarded under National Cybersecurity R&D Programme (Award No. NRF-NCL-P2-0001), and in part by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program. This material is based upon work partly supported by the National Science Foundation under grant #1935966 (FABRIC), #1547249, #1535070, and #2319190. Phuong is a TrustedCI Fellow, NSF Cybersecurity Center of Excellence. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] T. Basin, Y. Sade, and Y. Harel. Research: Nearly 70,000 sensitive industrial control systems exposed. [Online]. Available: https://www.otorio.com/blog/ics-exposures-research-blog/

[2] Z. E. Mrabet, N. Kaabouch, H. E. Ghazi, and H. E. Ghazi, "Cyber-security in smart grid: Survey and challenges," *Comput. Electr. Eng.*, vol. 67, no. C, p. 469–482, apr 2018. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2018.01.015

[3] H. He and J. Yan, "Cyber-physical attacks and defenses in the smart grid: A survey," *IET Cyber-Physical Systems: Theory & Applications*, vol. 1, 11 2016.

[4] P. Cao, Y. Wu, S. S. Banerjee, J. Azoff, A. Withers, Z. T. Kalbarczyk, and R. K. Iyer, "Caudit: Continuous auditing of ssh servers to mitigate brute-force attacks." in *NSDI*, vol. 19, 2019, pp. 667–682.

[5] M.-R. Zamiri-Gourabi, A. R. Qalaei, and B. A. Azad, "Gas what? i can see your gaspots. studying the fingerprintability of ics honeypots in the wild," in *ICSS: Proceedings of the Fifth Annual Industrial Control System Security (ICSS) Workshop*, ser. ICSS. New York, NY, USA: Association for Computing Machinery, 2019, p. 30–37. [Online]. Available: https://doi.org/10.1145/3372318.3372322

[6] J. Gajrani, J. Sarswat, M. Tripathi, V. Laxmi, M. Gaur, and M. Conti, "A robust dynamic analysis system preventing sandbox detection by android malware," in *SIN '15*, 09 2015, pp. 290–295.

[7] X. Chen, J. Andersen, Z. M. Mao, M. Bailey, and J. Nazario, "Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware," in *2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN)*, 2008, pp. 177–186.

[8] J. Uitto, S. Rauti, S. Laurén, and V. Leppänen, "A survey on anti-honeypot and anti-introspection methods," in *WorldCIST*, 2017.

[9] H. Shi, J. Mirkovic, and A. Alwabel, "Handling anti-virtual machine techniques in malicious software," *ACM Trans. Priv. Secur.*, vol. 21, no. 1, dec 2017. [Online]. Available: https://doi.org/10.1145/3139292

[10] P. Ackerman, *Industrial Cybersecurity: Efficiently secure critical infrastructure systems*. Packt Publishing, 2017. [Online]. Available: https://books.google.com.sg/books?id=FhlKDwAAQBAJ

[11] T. Zheng, M. Liu, D. Puthal, P. Yi, Y. Wu, and X. He, "Smart grid: Cyber attacks, critical defense approaches, and digital twin," 2022.

[12] S. Tufail, I. Parvez, S. Batool, and A. Sarwat, "A survey on cybersecurity challenges, detection, and mitigation techniques for the smart grid," *Energies*, vol. 14, no. 18, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/18/5894

[13] W. Han and K. Xu, "A method for placing traceroute-like topology discovery instrumentation," in *2008 11th IEEE Singapore International Conference on Communication Systems*, 2008, pp. 1160–1164.

[14] 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.

[15] R. Schimmel and S. Gerspach. (2011) Test procedures for goose performance according to iec 61850-5 and iec 61850-10.

[16] C. Condurache, L. Mogosanu, M. Carabas, L. Gheorghe, and N. Tapus, "Performance evaluation of in-kernel system calls," in *2015 4th Eastern European Regional Conference on the Engineering of Computer Based Systems*, 2015, pp. 130–133.

[17] T. J. Williams, "The purdue enterprise reference architecture," in *Proceedings of the JSPE/IFIP TC5/WG5.3 Workshop on the Design of Information Infrastructure Systems for Manufacturing*, ser. DIISM '93. NLD: North-Holland Publishing Co., 1993, p. 43–64.

[18] S. Adepu, N. K. Kandasamy, and A. Mathur, "Epic: An electric power testbed for research and training in cyber physical systems security," in *International Workshop on the Security of Industrial Control Systems and Cyber-Physical Systems*, 11 2018.

[19] IEEE, "Ieee standard communication delivery time performance requirements for electric power substation automation," *IEEE Std 1646-2004*, pp. 1–36, 2005.

[20] S. Meier and T. Werner. (2015) Op 060 – performance considerations in digital substation applications.

[21] T. Krause, R. Ernst, B. Klaer, I. Hacker, and M. Henze, "Cybersecurity in power grids: Challenges and opportunities," *Sensors*, vol. 21, p. 6225, 09 2021.

[22] D. Mashima, D. Kok, W. Lin, M. Hazwan, and A. Cheng, "On design and enhancement of smart grid honeypot system for practical collection of threat intelligence," in *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*. USENIX Association, Aug. 2020. [Online]. Available: https://www.usenix.org/conference/cset20/presentation/mashima

[23] W. Cabral, C. Valli, L. Sikos, and S. Wakeling, "Advanced cowrie configuration to increase honeypot deceptiveness," in *IFIP Advances in Information and Communication Technology*, 06 2021.

[24] D. Mashima, S. M. M. Roomi, B. Ng, Z. Kalberczyk, S. Hussain, and E.-C. Chang, "Towards automated generation of smart grid cyber range for cybersecurity experiments and training," in *The 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (Industry Track)*, 2023.

[25] W. O. Redwood, "Cyber physical system vulnerability research," Ph.D. dissertation, The Florida State University, 2016.

[26] K. Koltys and R. Gajewski, "Shape: A honeypot for electric power substation," *Journal of Telecommunications and Information Technology*, vol. 2015, pp. 37–43, 01 2015.

[27] D. 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*, 02 2014, pp. 181–192.

[28] J. T. Dougherty, "Evasion of honeypot detection mechanisms through improved interactivity of ics-based systems," Master's thesis, Naval Postgraduate School, 2020.

[29] M. M. Kendrick and Z. A. Rucker, "Energy-grid threat analysis using honeypots," Master's thesis, Naval Postgraduate School, 2019.