A Survey On Honeypots, Honeynets And Their Applications On Smart Grid*

Christos Dalamagkas[1], Panagiotis Sarigiannidis[1], Dimosthenis Ioannidis[2], Eider Iturbe[3], Odysseas Nikolis[2]
Francisco Ramos[4], Erkuden Rios[3], Antonios Sarigiannidis[5] and Dimitrios Tzovaras[2]

***Abstract*—Power grid is a major part of modern Critical Infrastructure (CIN). The rapid evolution of Information and Communication Technologies (ICT) enables traditional power grids to encompass advanced technologies that allow them to monitor their state, increase their reliability, save costs and provide ICT services to end customers, thus converting them into smart grids. However, smart grid is exposed to several security threats, as hackers might try to exploit vulnerabilities of the industrial infrastructure and cause disruption to national electricity system with severe consequences to citizens and commerce. This paper investigates and compares honey-x technologies that could be applied to smart grid in order to distract intruders, obtain attack strategies, protect the real infrastructure and form forensic evidence to be used in court.**

I. INTRODUCTION

The management and monitoring of electrical grids has been significantly improved during the last decade due to the rapid evolution of Information and Communication Technologies (ICT). ICT has also influenced the power industry, allowing transmission system operators (TSOs) to apply technologies that automate the administration of their electrical grid, while improving the utilization of existing resources and increase reliability, robustness and responsiveness to safety-critical situations. All those applied technological advancements have converted the traditional electrical grid into a "smart grid" [1].

A conventional power grid architecture is illustrated in Fig. 1. It consists of four domains, namely the Generation domain, where power generation takes place, the Transmission domain that carries electricity over large geographical areas, the Distribution domain that brings electricity to the customers and the Customer Premises, where power consumption occurs. Along those domains, several substations perform various functions like voltage transformation, data acquisition and supervision, by using Supervisory Control and Data Acquisition (SCADA) methods. In addition, several metering devices exist at customer's premises that monitor power consumption [1].

ICT modernize conventional grids by enabling Industrial Control System (ICS) devices, like Remote Terminal Units (RTUs), Programmable Logic Controllers (PLCs), Intelligent Electronic Devices (IEDs) and an Advanced Metering Infrastructure (AMI) of smart meters. AMI is used to communicate through a converged ICT infrastructure and automate the processes of monitoring, by taking actions towards billing the customers. Even more ICS devices use the Transport Control Protocol and Internet Protocol (TCP/IP) stack to exchange data, however additional security concerns may rise by this convergence. ICS infrastructure is often an attraction point for hackers that aim to intrude to a smart grid, obtain sensitive data or cause service disruption, leaded by various interests, from financial to political, even terrorism.

Disorientation of hackers and collection of useful information about the performed attacks are some efficient countermeasures that can protect a smart grid. Honey-x technologies, where x stands for honey-pots or honey-nets, are a common technique used in computer and industrial networks to identify attacks, collect intelligence about attack strategies and mislead cybercriminals from attacking the real infrastructure.

In this paper, the role of honeypots and honeynet frameworks are studied subject to securing and protecting the smart grid. Several honey-x technologies are presented and thoroughly discussed in terms of supported protocols, interaction and scalability as well as their efficiency in the smart grid applications.

The remainder of this paper is organized as follows. Section II describes the motivation and the contributions of this work. Section III provides the technical background about honeypots, by including related technical definitions, their classification as well as a short presentation of the most common industrial protocols that they emulate. In Section IV and V we describe the honeypots and honeynets, found in the literature, respectively. In Section VI we compare them and discuss about the best option that could be applied for smart grid use cases. Also, we describe our initial efforts to deploy the Conpot honeypot to a smart home use case of the EU-funded SPEAR project.

II. MOTIVATION

Networking and digitization of Critical Infrastructure (CIN) have leveraged cybersecurity as an emergency issue that draws major attraction in the research community. Power grids are an integral part of a state's CIN. The disruption of their operation,

*This work is supported by the Horizon 2020 programme of the European Union under grant agreement No 787011.

[1]C. Dalamagkas and P. Sarigiannidis are with the Dept. of Informatics and Telecommunications, University of Western Macedonia, Kozani, Greece {cdalamagkas, psarigiannidis} at uowm.gr

[2]D. Ioannidis, O. Nikolis and D. Tzovaras are with the Center for Research and Technology Hellas / Information Technologies Institute, 6th km Charilaou-Thermi Road, Thessaloniki, Greece {djoannid, odynik, Dimitrios.Tzovaras} at iti.gr

[3]E. Iturbe and E. Rios are with the Fundacion Tecnalia Research & Innovation, E-20009, Derio, Spain {Eider.Iturbe, Erkuden.Rios} at tecnalia.com

[4]F. Ramos is with the Schneider Electric, Charles Darwin s/n, Edificio Bogaris, 41092 Sevilla, Spain, francisco.ramos at schneider-electric.com

[5]A. Sarigiannidis is with the Sidroco Holdings Ltd, 3113, Limassol, Cyprus, asarigia at sidroco.com

978-1-5386-9376-6/19/$31.00 ©2019 European Union

C. Dalamagkas et al., "A Survey On Honeypots, Honeynets And Their Applications On Smart Grid," in 2019 IEEE Conference on Network Softwarization (NetSoft), 2019, doi: 10.1109/netsoft.2019.8806693.

The published version is available at IEEE Xplore Digital Library: https://ieeexplore.ieee.org/document/8806693

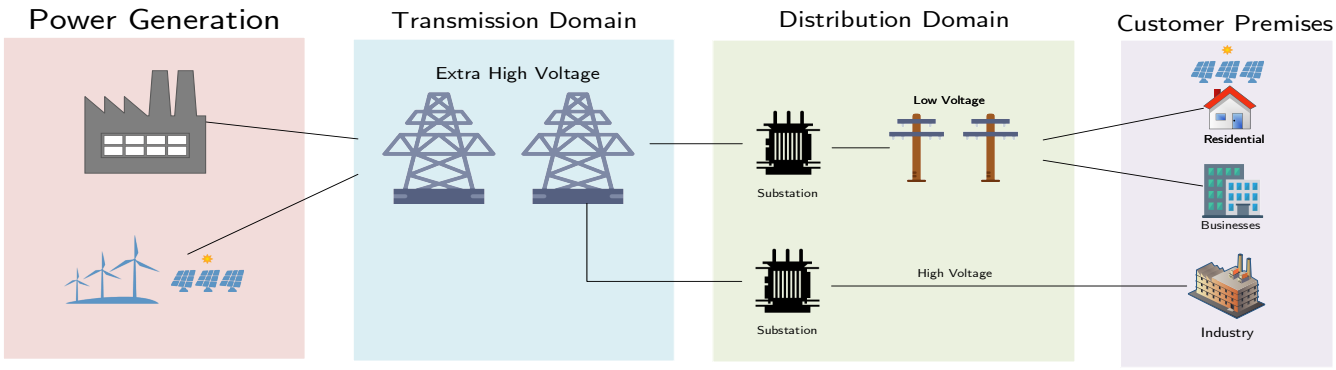

Fig. 1: A typical architecture of a power grid.

even for a short amount of time, may cause major problems to citizens and businesses, even human life loss.

Still recent is the example of Ukraine in 2015, when an impressive and very well orchestrated cyberattack to the national power grid switched off 30 substations and electricity knocked out for 225.000 citizens from 1 to 6 hours [2]. Traditional security measures, like firewalls and access control lists, seem not to be sufficient to prevent such sophisticated attacks. Honey-x technologies, often accompanied by Intrusion Detection Systems (IDS), are considered as an alternative way to protect a network by distracting attackers, whilst hiding the real infrastructure and retaining forensic information that could be used in courts.

To the best of our knowledge, there has not been any attempt yet to investigate and evaluate existing honey-x solutions that could be deployed to smart grid related use cases. This paper aims to provide an updated overview of existing honey-x technologies for smart grids that could be useful for both industry and academia.

III. BACKGROUND

A. Definition of honeypots and honeynets

Honeypots serve various purposes and have diverse capabilities, although they are commonly defined as "*an information system resource whose value lies in unauthorized or illicit use of that resource*" [3]. Honeypots are found to be useful when they deceit and trap attackers by simulating real devices, services or vulnerabilities. Trapping may be desirable in order to avoid getting attacked to production devices, to consume valuable resources of the intruder or to collect intelligence about their activity. A network of interconnected honeypots is called honeynet.

Honeypots are considered to be a strong defensive tool of modern IDS systems. More specifically, honeypots are able to complement the functionality of an IDS, if they are configured to send appropriate reports to log collectors, by capturing ingress traffic traces. Specialized parts of an IDS could further process these reports to identify attacks, or even to form legal evidences of cybercrime. As a digital tool that can also collect evidence of criminal activity, honeypots can also be exploited in network forensics, a sub-branch of digital forensics science, that encompass the recovery and investigation of data in order to investigate computer crimes. It should be noted that any interaction with a honeypot is considered a priori illicit [4].

B. Classification of honeypots

Honeypots are distinguished into various categories. First of all, honeypots can be classified by the level of interaction that they offer. Low-interaction honeypots simulate one or more services that offer limited and simple functions barely to attract attackers and to record interactions. The implementation of low-interaction honeypots is generally simple but they could easily get perceived, when an attacker tries to interact in a way that has not been foreseen in the implementation. In contrast, high-interaction honeypots provide an increased level of interaction because they offer advanced emulation, even real services. The advantage of high-interaction honeypots lies in the fact that attackers consume more time interacting with them, as they offer more functions, thus more activity data could be captured. A disadvantage is that they require more effort to develop and consume more resources at deployment and maintenance, so they are costly [3].

Honeypots can further be categorized as production and research honeypots. On the one hand, production honeypots forge real services, devices or entire operating systems. They are deployed in production environments to mitigate risks in an organization, by trapping attackers and preventing them to further invade the network. On the other hand, research honeypots are focused on gathering information about attackers, while are mainly used as a tool to study attack strategies and gain knowledge about existing cybersecurity threats [5].

Honeypots can also be differentiated in terms of being virtual or real. In the case of virtual honeypots, simple scripts can be implemented to represent services and protocols. Virtualization solutions like Docker and Cloud Provider Services (Amazon Web Services - AWS and Google Cloud Platform) as well as Software Defined Networking (SDN) technologies, like Mininet, can be utilized to emulate entire operating systems or networks that offer low or high interaction. Honeypot infrastructure can also include real devices that are used as trap for hackers and

can be more persuading than virtual ones. However, they are costly in terms of deployment and maintenance. Networking techniques like port mirroring can be applied to monitor real honeypots [6].

C. Emulated protocols and services

The operation of smart grid and ICS infrastructure is based on a variety of common ICT and industrial protocols that a honeypot should emulate in order to appear as a persuading emulation of a device or system.

An integral part of a SCADA system is the Human Machine Interface (HMI). HMI encloses a set of technologies that interpret data acquired from SCADA methods and represents them in a human-readable form. Some ICS devices incorporate the capability to communicate with web clients and HMI software. Thus, honeypots should be able to emulate basic network services and protocols that are used for HMI. The Hypertext Transfer Protocol (HTTP) is a very common protocol that is implemented in honeypots alongside with a lightweight web server that hosts a minimal web interface. This interface provides useful information about the device identity and/or operation as well as some basic control functions.

Since many ICS devices are directly connected to Ethernet Local Area Networks (LANs) and are using the TCP/IP stack, common protocols for network management and monitoring are also enabled on those devices. Telnet, Secure Shell (SSH), File Transfer Protocol (FTP) and Trivial FTP (TFTP) are often used for administering and file transferring. Administrators usually execute commands remotely or upload new firmware through these protocols. In addition, the Simple Network Management Protocol (SNMP) is also implemented for network monitoring and management.

Apart from common networking protocols, ICS devices also use dedicated industrial protocols and standards for encapsulation and message formatting. Modbus TCP is the TCP/IP variant of the Modbus protocol and carries the communication between PLCs, RTUs, power meters, control devices and HMIs that control and gather information from ICS devices. Modbus is widely adopted by vendors because it has simple implementation and is open-sourced. However it is known to have serious security issues [7]. Modbus TCP services listen on port 502.

Standards from the International Electrotechnical Commission (IEC) are also widely used by ICS devices. IEC-60870 is a large family of standards that define SCADA system operation and communication framework. IEC-60870-5-104 (IEC-104) defines the communication of industrial devices over standard TCP/IP networks and offers utilization of various data-link layer technologies like Ethernet, Asynchronous Transfer Mode (ATM) and Frame Relay. IEC-104 has been adopted by many vendors due to the flexibility and interoperability that TCP/IP offers, while it is a common communications protocol for ICS honeypots. Although, IEC-104 is known not to provision any security function [8], so it is vulnerable to packet interception and injection. IEC-104 uses the port 2404. In the same family of protocols, the IEC-60870-6/TASE.2 (ICCP) is a protocol that defines real time message exchanging between master stations over local and wide area networks using the International Standardization Organization (ISO) protocol stack. ICCP claims the ISO Transport Services Access Point (ISO-TSAP) port 102.

Focusing on substation automation, IEC has also standardized IEC-61850 in order to enhance and converge the automated control of electrical substations. IEC-61850 aims to provide a common and vendor neutral protocol for substation control and interoperability between RTUs from different vendors. IEC-61850 is a common choice for honeypots dedicated for smart grids and substations. Developers who aim to simulate IEC-61850 should consider implementing Generic Object Oriented Substation Events (GOOSE), a fast and reliable protocol that uses standardized data structures to transfer event data and operates directly over Ethernet. Also, the Manufacturing Message Specification (MMS) is of high interest as a protocol for exchanging real-time data and supervisory control information that operates over TCP and uses the ISO-TSAP port 102 [9].

In the context of substation automation, the Distributed Network Protocol 3 (DNP3) is the main alternative of IEC-61850, mostly used in United States, which mainly facilitates communications between supervisory stations and RTUs [10]. As an open standard, DNP3 is one of the most commonly used SCADA protocols in the electric power industry due to its efficiency and interoperability. These characteristics are based on the full TCP/IP stack support, the adoption of a 3-layered network model and compliance with IEC specifications. However, despite the popularity of this protocol, it has limited adoption on honeypot systems.

IV. HONEYPOTS OVERVIEW

A. Honeyd

Honeyd is one of the first initiatives for the development of an open-source low-interaction honeypot that can simulate a variety of TCP/IP services. Honeyd introduced the term of virtual honeypots, since it can launch many virtual hosts, where each one of them binds to an IP address and to a number of ports. The behaviour of honeyd hosts is defined by a configuration file, which consists of several templates. Each template defines a different kind of host that is simulated, which has a unique personality (operating system) and a combination of ports that are set on listen state. Optionally, a custom script could be specified for each port that is executed when a host interacts with that port. Each template can bind to one or more IP addresses and, optionally, can have a malformed Medium Access Control (MAC) address. Honeyd has the capability of recording activities in a log file, which is an important function for research honeypots [11].

A variant of Honeyd has been developed by the Critical Infrastructure Assurance Group of Cisco Systems to support SCADA protocols, under the SCADA HoneyNet Project [12]. The SCADA honeyd supports Modbus TCP, FTP, Telnet and HTTP. This project is not maintained anymore and is considered incomplete, as services are partially implemented and many

bugs have been reported, although it is a fundamental software for future initiatives in the honeypot domain.

B. HoneydV6

Inspired by Honeyd, authors in [13] proposed HoneydV6, an improvement of Honeyd that supports Internet Protocol Version 6 (IPv6). HoneydV6 is also a low-interaction honeypot and operates at the same logic as Honeyd. It uses the concept of virtual hosts and templates that follow the same syntax.

HoneydV6 supports both IP protocols. In addition, since IPv6 subnets can be huge and difficult to handle, HoneydV6 supports dynamic spawning of virtual hosts based on the activity that is performed by attackers. Transition to IPv6 is recommended as a security measure for ICS systems, although migration to the newer internet protocol is considered a slow procedure [14].

C. Conpot

Conpot is a low-interaction industrial honeypot, introduced by the Honeynet Project, that is focused on simulation of ICS devices [15]. Conpot keeps simple configuration templates, written in the Extensible Markup Language (XML), and each Conpot process is associated with a single template. The software comes with predefined templates that emulate a Siemens S7-200 PLC and a Kamstrup 382 smart meter. In version 0.6, Conpot supports numerous protocols like TFTP/FTP, SNMP, Modbus, IEC-104, EtherNet/IP and BACnet. Conpot showcases an exemplary honeypot solution since its core software can be expanded with new templates, while a single databus is used to provide measurement data to different protocols/templates [16].

An extensive description of the deployment and evaluation process of an earlier version of Conpot is described in [17]. Authors deployed an Ubuntu image with Conpot across several virtual machines around the world through AWS. In each geographical location, two different templates were enabled that simulated a Guardian AST gas pump and the default Siemens S7-200 PLC. The deployment was followed by a stage-approached port scanning with Nmap, using various flags, as well as the SHODAN engine. Evaluation results show a successful depiction of ICS devices but also reveal an important issue: additional ports were also opened, so the identity of those honeypots was exposed to sophisticated attackers.

D. CryPLH

A more complex ICS honeypot is presented in [18]. Authors introduced the Crysys PLC (CryPLH), a high-interactive honeypot that emulates a Siemens Simatic 300(1) PLC. Strong aspects of this implementation is the ease of configuration and the high accuracy of the emulated protocols and services. CryPLH implements both HTTP and HTTP Secure (HTTPS), by combining Miniweb and Nginx servers, as well as SNMP and S7Comm. All those services are integrated into a virtual machine that runs a minimal version of Ubuntu. The CryPLH Virtual Machine (VM) is protected through iptables, which filters unwanted traffic. Evaluation results show great similarity between the honeypot and the real device, although authors highlight the need to improve the TSAP implementation in order to overcome the Linux kernel limitation that prevents fixed TCP windows size, in conjunction with minor improvements of the SNMP client implementation.

An improved version of CryPLH is published in [19], where authors fixed many weaknesses of CryPLH. In addition, they expanded the simulation level of various protocols, whilst the Siemens ET/200S PLC device is now chosen for emulation. More specifically, authors improved the implementation of STEP7, based on the snap7 open-source project. They also decided to merge HTTP and STEP7 protocols into an integrated web and management service, as they were both used to query information from the PLC. Authors chose to keep the MiniWeb server but they replaced nginx with the Symbion SSL (Secure Sockets Layer) Proxy, as the former offers advanced capabilities that could bring suspicions to hackers. Also, authors arranged the limitations of the Linux kernel by applying run-time modifications to the TCP/IP stack, by incorporating constants and algorithms that are applied in the PLC. Evaluation results indicate a persuading simulation of the Siemens ET/200S PLC by the honeypot. Also, authors exposed the honeypot to the Internet and recorded that the majority of attacks was destined to non-industrial protocols.

E. SHaPe

SHaPe [20] is a low-interaction honeypot that is focused on substation automation systems. SHaPe is provided as a module of Dionaea, an open-source honeypot that simulates various non-SCADA services and aims to trap malware that tries to exploit vulnerabilities. SHaPe is able to simulate any IED that is compatible to the IEC-61850 standard. The desired configuration is applied by writing or even providing real configuration files of the IEC-61850 compliant device that the operator desires to emulate. SHaPe reads configuration files written in the Substation Configuration Language (SCL), which is based on XML, and allows the description of the network topology of the substation, including the data objects and the capabilities and properties of the device.

SHaPe is described as a lightweight process, able to bind to multiple IP addresses simultaneously. Thus, a SHaPe honeynet on a single virtual or physical machine can easily be deployed. The honeypot cannot handle messages from the GOOSE protocol, however it incorporates a complete implementation of the MMS protocol, since all IEC-61850 services that are mapped to MMS are supported by the honeypot. SHaPe also presents a fully-featured logging mechanism that records TCP interactions, like establishing and resetting connections, as well as various events specified in IEC-61850.

F. The CockpitCI project

In [6] authors provided a generic framework and guidelines for the development of ICS honeypots in the context of the CockpitCI project. The proposed framework implements a Modbus honeypot that consists of two main blocks, namely the honeypot front-end interface and the event monitor.

The front-end interface contains modules that simulate services, while interacting with the network. More specifically, the Modbus Application Programming Interface (API) module implements the Modbus protocol and replies to Modbus commands like a real PLC. Yet, the FTPD/SNMPD modules provide the corresponding services. The Port Scan module detects any activity that is related to the remaining ports. All modules report any activity to the Event Monitor component. The communication of the front-end interface with the rest of the local network is protected with a firewall that prevents outbound connections. This is a security measure that prevents the honeypot to be hijacked and been turned into an attack vector.

The Event Monitor component contains modules that collect raw logs from the front-end modules, while sending notifications to the event correlator in the standardized Intrusion Detection Message Exchange Format (IDMEF). In a nutshell, this component filters reports based on the operator's preferences, while aggregating reports that share similar characteristics. Then it transmits security event messages in IDMEF format to an Event Correlator, which is not part of the honeypot framework.

In addition, the authors proposed three different approaches regarding the honeypot framework deployment. The first approach is a cost-effective solution that uses the Raspberry Pi Single Board Computer (SBC), while the implementation was made in Python and C using the modbus-tk, pymodbus, net-snmpd, vsftpd and libpcap libraries. As a second approach, authors considered the deployment of the honeypot in a virtualized environment, since the software is fully portable. The third approach differs substantially from the aforementioned, as it uses a real non-production PLC as a high-interaction honeypot that is being exposed to attackers and monitored by a security adapter. The monitoring is implemented by an intermediate switch that mirrors all traffic of the PLC directly to the security adapter that implements the Event Monitor module.

V. HONEYNETS OVERVIEW

A. DiPot

DiPot [21] is a distributed network of honeypots that adapts a three-tier architecture that consists of honeypot nodes, a module for data processing and a visualization interface.

The Honeypot Nodes (HN) are based on Conpot, which is used as the main interface with the outer world and potential attackers. As being virtual machines, the nodes are logically placed to various locations around the globe. Authors enhanced the original implementation of Conpot in terms of simulation accuracy in order to appear more realistic. For example, authors added support for analog Input Output (IO) to Modbus protocol as well as reconfiguration of more realistic response delays.

HNs forward raw log files to the Data Processing Node (DPN), which is used for filtering the log files, removing duplicate data and applying custom format. The output is saved in a central database of the DPN. Meanwhile, the node uses the k-means algorithm to cluster the formatted logs for each HN separately on the basis of timestamp, source IP, protocol and function/slave Identifier (the last two apply only for Modbus). As a distance metric, authors adopted the Euclidean distance algorithm.

Finally, the DPN forwards it's output to the Management Node, which visualizes information in a user-friendly interface. HNs are plotted in a world map, based on their location, and information of each HN is organized by the aforementioned criteria of the classification process.

Authors evaluated the efficacy of DiPot by performing a wide deployment of HNs in various locations around the world. The experiment was running for 6 months and the results showed successive implementation of the honeynet since it attracted more than 300000 attacks and more than 4000 suspicious IP addresses. In addition, results showed that Modbus was the industrial protocol that received the most interactions, whilst S7comm was the least popular.

B. Serbanescu et al.

The honeynet that is presented in [22], [23] encompasses three separate modules that capture log files, filter traffic and generate events about suspicious activities. The proposed honeynet consists of honeypot nodes (Ubuntu VMs) that are deployed in various platforms, including physical infrastructure and AWS. These nodes are connected to Honeynet Storage Analysis & Management (SAM) nodes, which correlate and analyze traffic. Each honeypot node operates at low-interaction mode and runs basic services that simulate Modbus and IEC-104. For each service, corresponding iptables rules are added that redirect raw traffic to the honeypot software, whilst port scanning attempts to any other ports are captured by Snort. Honeypot nodes interface with the SHODAN engine through an API in order to customize their footprint. Captured data from honeypots nodes are stored in local databases (PostgreSQL), which are synchronized with the SAM databases. The SAM nodes play a crucial role in the honeynet infrastructure, since they carry the correlation and aggregation of the collected events as well as they implement further analysis using MATLAB and Python scripts. Also, SAM nodes control Honeypot nodes through Remote Procedure Call (RPC) services that run on honeypots, which are tunneled via SSH connections. Authors provide scalability to this honeynet topology by utilizing the Amazon Elastic Cloud (EC2) API in order to dynamically deploy and control honeypot nodes in various locations around the world.

Implementation details of the proposed framework include the Twisted framework for communication tasks like asynchronous operations, networking and protocol simulation. It was made in Python environment. The SAM databases run on the PostgreSQL Database Management System (DBMS) and the communication with the honeypot nodes is implemented by the psycopg library. The screen tool has been used to maintain multiple virtual terminals in honeypot nodes. Authors used a) the powerful Scapy tool to manipulate packets, b) nfqueue-bindings and iptables to forward and handle raw traffic, c) PyModbus to simulate the Modbus server as well as d) Snort to capture malicious port scanning attempts. SAM nodes

gather information about the potential attackers through the dnspython library and the GeoIP tool. Remote commands are being executed by the Fabric library, while RPyC and Plumbum enable the execution of remote procedure calls. Finally, the Amazon EC2 and SHODAN APIs are implemented through boto and shodan python libraries.

Authors evaluate the honeynet platform by deploying and bringing online honeypots in AWS and SAM nodes for 28 days. Evaluation results confirm the persuading implementation of honeypots and the fact that Modbus attracts the majority of requests and connection attempts.

The large-scale honeynet has been expanded in [24] in order to study the attraction of additional industrial protocols. More specifically, each honeypot is able to simulate Modbus, IEC-104, DNP3, ICCP, the Extensible Messaging and Presence Protocol (XMPP), TFTP and all three versions of SNMP. It should be noted that IEC-104, DNP3 and ICCP have been implemented in passive mode, so they only receive and analyze traffic. Therefore they cannot be indexed by SHODAN. Authors deployed the aforementioned protocols in 6 different combinations.

Evaluation results of the second deployment show that the combination of protocols that each honeypot simulates does not have any impact to its attractiveness. At the same time, SNMP received the vast majority of the reconnaissance attacks, while Modbus was the most popular industrial protocol. Other industrial protocols received insignificant amount of requests. Authors conclude that a high interaction honeypot would be more useful to gain a bigger picture of the threat environment. Also, authors realize the effectiveness of the SHODAN engine for identifying devices, since honeypots indexed by the engine were more attractive.

C. Mashima et al.

A proof-of-concept implementation of a honeynet that is focused solely on smart grid and electrical substation networks is presented in [25]. Authors realize the need to emulate realistic topologies in order to persuade and distract attackers. Consequently, they defined the following requirements that a successful substation honeynet should satisfy: i) A comprehensive, consistent power grid overview ii) Realistic network configuration iii) Scalability for grid-wide emulation and iv) Fingerprinting resistance.

Each honeypot in the honeynet emulates a substation and consists of two virtual machines: a vulnerable gateway that interfaces with the Internet and a VM that contains a Mininet topology with a number of virtual hosts that represent IEDs of the substation. Virtual hosts provide only the networking front-end that is visible to intruders; each virtual host is connected through an internal virtual switch (vSwitch) to the Powerworld Server, which emulates the actual function of each IED, being able to respond to IEC-104 messages.

As for the implementation part of the honeynet, the gateway serves as the entry point to the substation, so authors configured it with vulnerable SSH and Virtual Private Network (VPN) settings, in order to attract attackers. The VM (or a real device could be used instead) appears as a real IEC-104 gateway, since attackers can connect to it in order to further penetrate into the honeynet. The Java-based OpenMUC tool has been employed for the IEC-104 interface implementation. The substation network consists of one or more instances of the Mininet software, an SDN tool that implements entire network topologies with virtual hosts and vSwitches. Spanning Tree Protocol (STP) is also engaged to prevent switching loops. Virtual hosts use the SOCAT tool to outsource IEC-61850 messages to the Powerworld server, which generates and serves IEC messages. The Powerworld server employs the SoftGrid testbed to accurately emulate IEC-61850 IEDs and connects them to the same power flow simulation. Respond latencies have been carefully configured to respond to realistic values.

Authors evaluated the efficiency of the honeynet under the requirements they set. First of all, the "Comprehensive, Consistent Power Grid View" criterion is satisfied because latency responses are rational. The realistic network configuration requirement has also been satisfied since MAC address spoofing, FastEthernet speed of ports as well as ring topology implementation have been taken place. The "Fingerprinting Resistance" criterion has also been taken care of since scanning attempts with nmap show that virtual host do not reveal their operating system and substation gateways are detected as Linux. A subtle issue that could leave honeypot fingerprints is the fact that if multiple virtualized honeypot instances with high load share the same Central Processing Unit (CPU), then the latency, in terms of network and processing, is critically affected. Authors overcome this issue by hosting multiple Mininet instances on a single VM that owns a single CPU core. At last, the honeynet is proved scalable and inexpensive since it utilizes virtualization and SDN technologies to scale-up instantly, thus able to simulate hundreds of substations at low cost.

D. The Symbolic Honeynet Framework

A honeynet for cyber-physical systems (CPS) is presented in [26]. Authors implement a honeynet framework to simulate a substation that consists of an HMI and emulated cyber-physical devices. Attackers are enticed to perform realistic attacks that disrupt the power flow and the normal operation of IEDs.

The honeynet framework consists of tree different layers. The Honeynet Layer is the front-end of the honeynet system and serves as an abstraction layer that enables multiple honeypots, like Conpot, to interface with the rest of the honeynet. The Interaction Layer interfaces with the Honeynet Layer and is a host-only network that enables hackers to penetrate the cyber-physical environment and fire attacks. The Infrastructure Modeling Layer implements a symbolic data flow of a weak Kahn process network, that simulates various physical variables like power flow, voltage, current and pressure, with the help of GridLAB-D. Each Layer feeds with incidents the Logging Layer, which performs anomaly detection using numerous algorithms.

Implementation details include Conpot for the HMI and GridPot, a template for Conpot that interfaces with the

TABLE I: Comparison of honeynets and Honeypots

	HTTP/HTTPS	Telnet/SSH	TFTP/FTP	SNMP	Modbus	IEC-104	ICCP	GOOSE/MMS	DNP3	S7Comm	EtherNet/IP	BACnet	High-interaction	IPv6
HONEYPOTS														
Honeyd [12]	✓/-	✓	-/✓	-	✓	-	-	-	-	-	-	-	-	-
HoneydV6 [13]	✓/-	-	-/✓	-	-	-	-	-	-	-	-	-	-	✓
Conpot [17]	✓/-	-	✓	✓	✓	✓	-	-	-	✓	✓	✓	-	-
CryPLH [18], [19]	✓	-	-	✓	-	-	-	-	-	✓	-	-	✓	-
SHaPe [20]	-	-	-	-	-	-	-	-/✓	-	-	-	-	-	-
CockpitCI [6]	-	-	-/✓	✓	✓	-	-	-	-	-	-	-	✓	-
HONEYNETS														
DiPot [21]	✓/-	-	-	✓	✓	-	-	-	-	✓	-	✓	-	-
Serbansecu et al. [24]	-	-	✓/-	-	✓	✓	✓	-	✓	-	-	✓	-	-
Mashima et al. [25]	-	-/✓	-	-	-	✓	-	✓	-	-	-	-	✓	-
Symbolic Honeynet [26]	✓/-	-	-	-	-	-	-	✓	-	-	-	-	✓	-

✓: All protocols of the column applies to the honey-x.
- : None of the protocols of the column applies to the honey-x.
✓/- or -/✓: Only one protocol of the column applies to the honey-x.

Interaction Layer and the IEC-61850 standard. The logging Layer receives feed from the hpfeeds tool and is able to apply 9 different anomaly detection algorithms to detect anomaly (e.g. Least square), supported by the ETSY Skyline Project library. GridLAB-D has been adopted to simulate the power flow and a VMware product has been used for virtual networking.

Authors evaluated the framework by deploying a number of honeynets that are connected through the simulated power flow. A malware has been developed to cause a switching attack against GOOSE/MMS protocols of IEC-61850 that has physical impacts to substations (e.g. change of voltage or current). Results show that the attack has been successfully conducted and reflects realistic changes of the power flow.

VI. DISCUSSION

A. Comparison of honey-x solutions

Table I compares the ICS honeypots and honeynets of our research, in terms of supported protocols and depth of interaction. (T)FTP and HTTP are the most commonly supported protocols, Modbus and SNMP are also quite popular, while more sophisticated protocols and standards like EtherNet/IP, BACnet and IEC-61850 are least implemented. We notice the lack of implementation of popular industrial protocols that are observable in smart grids like GOOSE/MMS, DNP3 and ICCP. Also, the support of IPv6 would be considered as a future requisite, since IPv6 is gaining popularity in industrial networks [14], [27].

Based on our findings, Conpot outbalances all other honeypots, since it is open-source, has active development, is expandable with new templates and can be installed in a common Ubuntu 18.04 LTS without any great effort. Also, as a plus of Conpot is considered the fact that supports numerous industrial protocols like Modbus, IEC-104 and BACnet that can be used in smart grid use cases. An important drawback of Conpot is that it currently lacks native support of additional power-oriented protocols and standards like IEC-61850, DNP3, so it's deployment for substation scenarios comes to be more challenging.

As for the honeynet frameworks, authors in [24] pose an exemplary design of a large-scale honeynet since it takes advantage of AWS and APIs to provide scalability and flexibility regarding deployment and management of the honeynet. Also, the exposure to the SHODAN engine is another advantage since it attracts even more attackers. A possible improvement would be the usage of Conpot as core software. Last but not least, the high-interaction honeynets [26] and [25] present high-quality implementation of substations, although they require great effort to deploy and are restricted to specific use case scenarios.

B. Adoption of Conpot to the Smart Home use case

The SPEAR project proposes a novative platform that exploits honeypot technologies to detect incidents, perform anomaly detection and provide forensic data that prepare the necessary legal evidence in court. SPEAR realizes a number of use cases that will demonstrate the full potential of the SPEAR platform. The Smart Home use case is one of those uses cases and is implemented in a near-zero energy building, equipped with a multi-sensorial network that measures in real time almost every challenging aspect of a modern house/work place (energy, occupancy, health, etc.). The smart home is hosted by the Center of Research and Technology Hellas (CERTH).

A high-level diagram of the SPEAR use case is depicted in figure 2. The testbed consists of real smart devices, gateways and a photo-voltaic (PV) panel that generates and stores power. The PV is also connected to the smart home network for data measurement, acquisition management and control. The SPEAR Security information and event management (SIEM) platform monitors the home network, whilst the Conpot v0.6 honeypot that runs on an Ubuntu 18.04 VM waits passively for connections. The Conpot VM supports Modbus and BACnet, protocols that are also used by devices of the use case to expose measurements.

Installation and configuration of Conpot proved to be simple, although some limitations of Conpot arose that we managed to overcome. For example, Conpot is not permitted to run with root privileges, in order to bind to ports lower than 1024 that some industrial protocols use, like Modbus, therefore appropriate iptables rules to the NAT table of the honeypot VM were applied that redirect traffic from the legitimate ports to the actual non-root ports that Conpot listens to. In addition, we adjusted the source code of Conpot so that it saves logs about incoming traffic in json format that also includes payload information. Finally, we adjusted the Conpot databus in order to provide non-constant measurements to the core software.

VII. CONCLUSIONS

This paper presents a comprehensive survey on honey-x technologies that can be applied to smart grid use cases. The

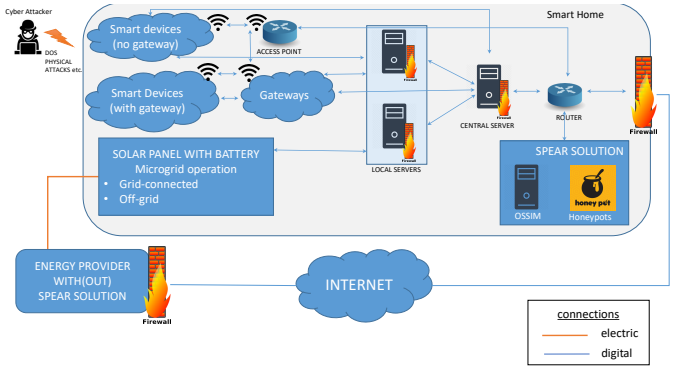

Fig. 2: The high-level diagram of the Smart Home use case.

comparison highlights Conpot as the honeypot able to support numerous smart grid use cases since it supports of many popular industrial protocols. Even though Conpot lacks the support of specific power grid protocols like GOOSE/MMS and DNP3, it is an open-source project and the research community can contribute to the development of additional templates. Also, regarding substation scenarios, it is worth mentioning that [25], [26] are deemed as remarkable implementations of realistic simulations that provide realistic, high-interaction honeypots.

Conpot is chosen by the SPEAR project consortium for various use case demonstrations since it is a flexible, open-source, expandable and customizable tool. Also, it is supported by an active community, while it encloses numerous industrial protocols. As a next step, we intend to deploy Conpot in many pilot sites of the SPEAR project for capturing and analyzing malicious traffic.

ACKNOWLEDGMENT

This paper is supported by the SPEAR project, a Horizon 2020 program, funded by the European Union under the grant agreement No. 787011.

REFERENCES

[1] P. Kalkal and V. K. Garg, "Transition from conventional to modern grids: Modern grid include microgrid and smartgrid," in *4th IEEE International Conference on Signal Processing, Computing and Control, ISPCC 2017*, vol. 2017-Janua. IEEE, 9 2017, pp. 223–228. [Online]. Available: http://ieeexplore.ieee.org/document/8269679/

[2] J. E. Sullivan and D. Kamensky, "How cyber-attacks in Ukraine show the vulnerability of the U.S. power grid," *Electricity Journal*, vol. 30, no. 3, pp. 30–35, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.tej.2017.02.006

[3] L. Spitzner, "Honeypots: catching the insider threat," in *19th Annual Computer Security Applications Conference, 2003. Proceedings.*, no. Acsac. IEEE, 2003, pp. 170–179. [Online]. Available: http://ieeexplore.ieee.org/document/1254322/

[4] A. Almulhem, "Network forensics: Notions and challenges," in *2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, no. January. IEEE, 12 2009, pp. 463–466. [Online]. Available: http://ieeexplore.ieee.org/document/5407485/

[5] L. Spitzner, "The Value of Honeypots, Part One: Definitions and Values of Honeypots | Symantec Connect Community," 2001. [Online]. Available: https://www.symantec.com/connect/articles/value-honeypots-part-one-definitions-and-values-honeypots

[6] P. Simões, T. Cruz, J. Proença, and E. Monteiro, "Specialized honeypots for scada systems," in *Intelligent Systems, Control and Automation: Science and Engineering*, 2015, vol. 78, pp. 251–269. [Online]. Available: http://link.springer.com/10.1007/978-3-319-18302-2

[7] P. Huitsing, R. Chandia, M. Papa, and S. Shenoi, "Attack taxonomies for the Modbus protocols," *International Journal of Critical Infrastructure Protection*, vol. 1, no. C, pp. 37–44, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.ijcip.2008.08.003

[8] P. Maynard, K. Mclaughlin, and B. Haberler, "Towards Understanding Man-In-The-Middle Attacks on IEC 60870-5-104 SCADA Networks," *ICS-CSR 2014 Proceedings of the 2nd International Symposium on ICS & SCADA Cyber Security Research 2014*, no. 2014, pp. 30–42, 2014. [Online]. Available: http://dx.doi.org/10.14236/ewic/ics-csr2014.5

[9] Y. Liang and R. H. Campbell, "Understanding and Simulating the IEC 61850 Standard," *Ieee Trans. On Power Delivery*, vol. 22, pp. 1482–1489, 2007. [Online]. Available: http://hdl.handle.net/2142/11457

[10] S. East, J. Butts, M. Papa, and S. Shenoi, "A Taxonomy of Attacks on the DNP3 Protocol," in *IFIP Advances in Information and Communication Technology*, 2009, vol. 384 AICT, no. 0, pp. 67–81. [Online]. Available: http://link.springer.com/10.1007/978-3-642-04798-5_5

[11] N. Provos and T. Holz, *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. Addison-Wesley Professional, 2007.

[12] V. Pothamsetty and M. Franz, "SCADA HoneyNet Project: Building Honeypots for Industrial Networks," 2005. [Online]. Available: http://scadahoneynet.sourceforge.net/

[13] S. Schindler, B. Schnor, S. Kiertscher, T. Scheffler, and E. Zack, "HoneydV6: A Low-interaction IPv6 Honeypot." *Secrypt*, pp. 86–97, 2013. [Online]. Available: http://dblp.uni-trier.de/db/conf/secrypt/secrypt2013.html#SchindlerSKSZ13

[14] D. R. V. A. S. Kumar, A. Taranum, and M. S. Goud, "Technique for Migration to IPV6 for a Secure SCADA Architecture," vol. 4, no. 4, pp. 128–133, 2014.

[15] L. Rist, "Introducing Conpot," 2013. [Online]. Available: https://www.honeynet.org/node/1047

[16] "Conpot on GitHub." [Online]. Available: https://github.com/mushorg/conpot

[17] A. Jicha, M. Patton, and H. Chen, "SCADA honeypots: An in-depth analysis of Conpot," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, no. May 2013. IEEE, 9 2016, pp. 196–198. [Online]. Available: http://ieeexplore.ieee.org/document/7745468/

[18] 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 *Smart Grid Security*. Springer International Publishing Switzerland, 2014, pp. 181–192. [Online]. Available: http://link.springer.com/10.1007/978-3-319-10329-7_12

[19] T. Holczer, M. Félegyházi, and L. Buttyán, "The design and implementation of a PLC honeypot for detecting cyber attacks against industrial control systems," Tech. Rep., 2015.

[20] K. Kołtyś and R. Gajewski, "SHaPe: A Honeypot for Electric Power Substation," Tech. Rep.

[21] J. Cao, W. Li, J. Li, and B. Li, "DiPot: A Distributed Industrial Honeypot System," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10699 LNCS, 2018, pp. 300–309. [Online]. Available: http://link.springer.com/10.1007/978-3-319-73830-7_30

[22] A. V. Serbanescu, S. Obermeier, and D.-y. Yu, "A Flexible Architecture for Industrial Control System Honeypots," pp. 16–26, 2015.

[23] A. V. Serbanescu, S. Obermeier, and D.-Y. Yu, "A Scalable Honeynet Architecture for Industrial Control Systems," in *E-Business and Telecommunications*, 2016, pp. 179–200. [Online]. Available: http://link.springer.com/10.1007/978-3-319-30222-5_9

[24] A. Vlad, S. Obermeier, and D.-Y. Yu, "ICS Threat Analysis Using a Large-Scale Honeynet," in *Proc. 3rd Int. Symp. for ICS & SCADA Cyber Security Research*, 2015, pp. 20–30. [Online]. Available: http://ewic.bcs.org/content/ConWebDoc/55096

[25] 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, 10 2017, pp. 89–95. [Online]. Available: http://ieeexplore.ieee.org/document/8340689/

[26] O. Redwood, J. Lawrence, and M. Burmester, "A Symbolic Honeynet Framework for SCADA System Threat Intelligence," in *Dissertation, FSU*, 2015, pp. 103–118. [Online]. Available: http://link.springer.com/10.1007/978-3-319-26567-4_7

[27] J. Höglund, J. Eriksson, N. Finne, R. Sauter, and S. Karnouskos, "Event-driven IPv6 communication for the smart grid infrastructure," *2011 International Conference on Distributed Computing in Sensor Systems and Workshops, DCOSS'11*, 2011.