The paper performs a comprehensive evaluation of Conpot, a widely used open-source low-interaction ICS honeypot. The authors assess Conpot's realism, vulnerability, and ability to deceive attackers, as well as its limitations and possible improvements.

SCADA Honeypots

An In-depth Analysis of Conpot

Arthur Jicha, Mark Patton, and Hsinchun Chen
University of Arizona
Department of Management Information Systems
Tucson, AZ 85721, USA
ajicha@email.arizona.edu, mpatton@email.arizona.edu, hchen@eller.arizona.edu

***Abstract*—Supervisory Control and Data Acquisition (SCADA) honeypots are key tools not only for determining threats which pertain to SCADA devices in the wild, but also for early detection of potential malicious tampering within a SCADA device network. An analysis of one such SCADA honeypot, Conpot, is conducted to determine its viability as an effective SCADA emulating device. A long-term analysis is conducted and a simple scoring mechanism leveraged to evaluate the Conpot honeypot.**

***Keywords*—Supervisory Control and Data Acquisition systems, honeypots, Conpot, network security**

I. Introduction

In a world where the value of information is ever increasing, hackers are consistently targeting governments, corporations, and individuals to obtain valuable secrets, proprietary data, and personally identifiable information (PII). Honeypots can be used to better understand the landscape of where these attacks are originating. Honeypots can be leveraged not only to conduct research on threats in the wild, but also to notify an organization if a potential threat is within one's network. Supervisory Control and Data Acquisition (SCADA) systems are a critical target, and with the advent of SCADA honeypots, attempts to access or tamper with SCADA devices can be preemptively identified and analyzed.

A. Background

SCADA Honeypots attempt to mimic an active SCADA system. A typical SCADA system is composed of four parts: "a central computer (host), a number of field-based remote measurement and control units known as Remote Terminal Units (RTUs), a wide area telecommunications system to connect them, and an operator interface to allow the operator to access the system" [1].

Conpot is a low-interactive SCADA honeypot and serves the purpose of being extremely easy to implement. Serbanescu et al., for example, found that Conpot would support the simulation of hypertext transfer protocol (HTTP), Modbus (a serial communication protocol), and Simple Network Management Protocol (SNMP; used for network management), and the integration of programmable logic controllers (PLC) [2]. The Conpot project by The Honeynet Project was released in May 2013. Conpot utilizes a logging system to monitor any changes that are made by intruders. The honeypot logs events of HTTP, SNMP and Modbus services with millisecond accuracy and offers basic tracking information such as source address, request type, and resource requested in the case of HTTP [3].

B. Research Gap

In a literature review of SCADA honeypots, a gap was identified regarding the analysis of the effectiveness of the various honeypots. Studies were found that detailed the interactions occurring with a given honeypot, i.e., Digital Bond Honeynet and Conpot; however, studies of the actual effectiveness of any given honeypot have not been conducted. The closest approach to this field of study was carried out by Fronimos, et. al., whose study focused on evaluating the usability and performance of Low Interaction Honypots, but did not examine the specifics of SCADA honeypot efficacy [4]. A more detailed look at the efficacy of SCADA honeypots that takes into account their unique requirements has not been conducted prior to this research. This paper performs a detailed evaluation of the Conpot SCADA Honeypot.

II. Experiment Approach

To conduct a full analysis of the SCADA honeypot Conpot, a virtualized image was created and used in multiple Amazon Web Services' (AWS) zones. The SCADA honeypots ran from March 25th to April 11th and the logs were subsequently analyzed. An additional log set was pulled April 27th for further analysis. The following section outlines the steps for setup and process for creating instances of Conpot.

Installation of Conpot is quite simple; however, certain dependencies are necessary for it to fully function. Due to the age of some of the required packages, repositories must be manually added. Ubuntu 12.04, an open source software platform used for various mobile and other devices, was used as the base operating system for a micro-instance within AWS, after configuring basic settings and conducting updates.

A. Experiment Setup

After successfully obtaining the Conpot start screen, the AWS micro-instance was shut down so that an image could be created. Utilizing the "Create Image" function within AWS, the image was then added to the Images – AMI folder for deployment. This image was then propagated to additional AWS deployment zones. After deploying the image twice in each zone (see Table I), the SCADA honeypots were booted and accessed via SSH to finalize their deployment.

This material is based upon work supported by the U.S. National Science Foundation under Grant No. DUE-1303362 and SES-1314631.

An advantage to leveraging AWS is its key management and port security options. Each instance of the Conpot was set up to allow all ports to be accessible and to provide an accurate review of port information when running any given honeypot template. Furthermore, the key pair options facilitated maintaining secure access to each instance. After obtaining the private key necessary to create a connection, each instance was generated using the same public key which allowed access using one private certificate combined with the instance password.

After accessing each honeypot, the following command was utilized to start the Conpot with the designated template:

- sudo conpot --template [template name]

If a template name is not selected, the default option of "default" is used. For the purposes of the honeypot analysis, an in-depth review of both the Guardian AST gas pump monitoring system and default Siemens S7-200 ICS was performed together with a brief analysis of the IPMI - 371 and Kamstrup – 382 smart meter SCADA devices.

B. AWS Deployment

The following table summarizes the deployed Conpot honeypots by their location, IP address, and template details. The honeypots were deployed globally across AWS for future analysis into regional variations on attack frequency and type:

TABLE I. AWS CONPOT DEPLOYMENT ZONE INFORMATION

AWS Location	Name	IP	Details
us-east-1a	Conpot1	52.23.225.126	Default template
us-east-1a	Conpot2	54.86.249.160	Emulation of gas tank level
us-west-2b	Conpot3	52.36.62.44	Default template
us-west-2b	Conpot4	52.32.45.32	Emulation of gas tank level
eu-west-1b	Conpot5	52.30.167.154	Default template
eu-west-1b	Conpot6	52.19.95.69	Emulation of gas tank level
ap-northeast-1c	Conpot7	52.192.20.179	Default template
ap-northeast-1c	Conpot8	52.196.47.205	Emulation of gas tank level
ap-southeast-1b	Conpot9	54.254.141.38	Default template
ap-southeast-1b	Conpot10	54.254.140.52	Emulation of gas tank level
sa-east-1a	Conpot11	54.207.96.59	Default template
sa-east-1a	Conpot12	54.232.248.38	Emulation of gas tank level

III. DATA AND RESULTS

A. Nmap Scan Data

The security scanner Nmap was utilized to check the open ports after starting Conpot. Nmap was chosen as it is a mature, robust connection-oriented canning tool that is widely used and has broad support for many protocols. For initial comparison, a "vanilla" installation of Ubuntu was also deployed and scanned to show what ports are open by default. The following Nmap scanning commands were used:

- nmap -A -v [IP Address]
- nmap -A -v -Pn [IP Address]
- nmap -A -v -Pn -p- [IP Address]

Nmap was used in a staged approach to show what different scanning techniques showed as the open port results (Tables II and III). The flag -A results in Nmap turning on "version detection and other Advanced and Aggressive features" (nmap.org). This scanning technique is intrusive and readily detected due to its aggressive scanning and operation systems (OS) detection, but it provides a good representation of what to expect for identification. Using the -Pn resulted in Nmap suppressing pings when conducting scans to determine if a host is up. For the purposes of the analysis, the virtual machines were already known to be operational and in some cases their configurations rejected pings. The -p- flag was also used to conduct a scan over the entire port range (ports 1-65535). Lastly, the flag -v (version detection) was used also, although it was later deemed not necessary, as the –A flag already included version detection.

TABLE II. NMAP SCANNING (UTILIZING FLAGS –V AND –A)

Honeypot Type	Result	Ports Opened by Conpot
Siemens S7-200	22, 80	80,102, 161, 502, 623, 47808
Guardian AST	N/A	10001
IPMI	N/A	623
Kampstrup Smart Meter	N/A	1025, 50100

Scanning with the -v and -A flags resulted in no results from the Guardian AST, IPMI, and Kampstrup smart meter, due to pings being rejected by these SCADA configurations. The revelation of port 22 through a ping scan should allow an attacker to question whether the Siemens S7-200 emulator is a honeypot or an actual SCADA device.

TABLE III. NMAP SCANNING (UTILIZING –V, -A, AND -PN FLAGS)

Honeypot Type	Result	Ports Opened by Conpot
Siemens S7-200	22, 25, 80, 514, 6009, 8443	80,102, 161, 502, 623, 47808
Guardian AST	22, 25, 514, 6004, 10001	10001
IPMI	22	623
Kampstrup Smart Meter	22, 25, 514, 1025, 1068	1025, 50100

After utilizing the -Pn flag to stop the ping option during scans, many more ports were identified across the various usable templates within Conpot. However most of these additional ports were not SCADA ports; for example, port 514 was for system logging, while many of the opened SCADA ports remained undetected. This indicates that Conpot installations running on Ubuntu appear to be very susceptible to having Ubuntu default services enabled and running across a multitude of ports that would not be available on a standard SCADA installation.

As a final scan to compare against, *all* ports were scanned to determine what a full Nmap scan would show as open port results (Table IV). On average the scans took around three to four hours to fully process due to the intensity of the scans. The wide range of additional open ports, including ports in the dynamic/private range of 49152-65536 (note: The Kampstrup Smart Meter statically assigns a port in this range) again calls into question the ability of a default Conpot installation that

does not actively close all other port-opening Ubuntu services to masquerade as an actual SCADA device, if comprehensive port scanning is utilized, or even if repositories such as Shodan are.

TABLE IV. NMAP SCANNING (UTILIZING –V, –A, –PN, AND -P- FLAGS)

Honeypot Type	Result	Ports Opened by Conpot
Siemens S7-200	22, 80, 102, 502, 514, 2000, 5060, 8008, 8020, 18556	80,102, 161, 502, 623, 47808
Guardian AST	22, 514, 2000, 3826, 5060, 8008, 8020, 10001, 11190, 19116, 36123, 43787, 48191, 63790	10001
IPMI	22, 2000, 5060, 8008, 8020	623
Kampstrup Smart Meter	22, 514, 1025, 2000, 4368, 5060, 8008, 32469, 50100, 52245, 57565	1025, 50100
Vanilla Ubuntu Install	22, 514, 2000, 5060, 8008, 8020, 38051, 38093, 47785	

B. SHODAN Scan Data

SHODAN data was also analyzed to determine which ports it detected as open within the various Conpot templates. Shodan regularly scans the entire IPV4 internet address space and as such is a reliable indicator of what can be seen by third parties conducting reconnaissance scanning. Unfortunately, the IPMI and Kampstrup templates were never identified by SHODAN due to time constraints.

TABLE V. SHODAN SCAN DATA RESULTS

Honeypot Type	SHODAN Port Scan Results	Conpot Ports
Siemens S7-200	22, 80, 102, 161	80,102, 161, 502, 623, 47808
Guardian AST	10001	10001
IPMI	N/A	623
Kampstrup Smart Meter	N/A	1025, 50100

C. Scan Data Discussion

A very interesting finding in the Nmap scan data is that while the Guardian AST, Kampstrup, and IPMI devices all denied pings, the Siemens SIAMATIC S7-200 did not. When removing the ping option for the result set in Table III, the results were more comprehensive and revealing. In every scan result, port 22 was shown as open, which would be the case due to utilizing SSH to gain access to each honeypot via a terminal in Putty. When comparing what should have been seen as open ports for each respective template within Conpot to the results from Table III, Nmap failed to identify the following ports as open on their respective devices:

- Siemens S7-200: 102, 161, 502, 623, 47808
- IPMI: 623
- Kampstrup Smart Meter: 50100

However, these ports may not have been found due to not being part of the top 1,000 which Nmap commonly scans without being directed to scan each and every port. To that point, Nmap was eventually set to scan each and every port (Table IV). After scanning all ports, some ports that should have been open were still not found. The results are as follows for ports which were not found:

- Siemens S7-200: 161, 623, 47808
- IPMI: 623

This requires further research. In the case of the Siemens device, SHODAN found port 161 and captured a banner from it, while Nmap did not detect it. What was more surprising during the full comprehensive scan was the large number of open ports that were not expected to be open at all within Table V. Due to the large variety of ports that were discovered to be open, the "vanilla" install of the Ubuntu image was deployed without running any Conpot template. Based on a scan of the vanilla Ubuntu, it appears that more ports were being opened than would be originally anticipated when running any given Conpot template. Further analysis will need to be conducted to determine which extra ports being opened might be indicative of a honeypot instead of an effective emulation.

The results from the SHODAN scan were also very insightful in that they more accurately showed the Conpot instances as being SCADA devices. This is primarily because SHODAN focuses its scan results on a much smaller port set, which resulted in the results not showing the large number of open ports that were shown in the all-port scan of Nmap. The most intriguing finding here, as previously mentioned, is that SHODAN found port 161 open on the Siemens device, while Nmap did not. The banner grabbed by SHODAN also showed that the device was a Siemens SIAMATIC S7-200 device. These findings may show that Nmap is indeed not fully effective in determining ports that are actually open. Unfortunately, at the time of this writing, SHODAN had not discovered the IPMI and Kampstrup devices, so a comparison of the SHODAN results of these devices with the Nmap port scans was not available.

Additional future work includes evaluating the SCADA Honeynet Honeypot, analyzing SCADA honeypot attacks, and evaluating log analysis tools. Another future task, cloaking Honeypot signatures that could differentiate them from real SCADA devices then evaluating attack differentials, could help determine if honeypots are being identified.

In conclusion, the devices accurately depicted SCADA ports, but appeared to have additional ports open that could reveal their identity as honeypots to sophisticated attackers.

REFERENCES

[1] S. Wade. "SCADA Honeynets: The attractiveness of honeypots as critical infrastructure security tools for the detection and analysis of advanced threats." Graduate Theses and Dissertations, Iowa State University, USA, 2011.

[2] A. Serbanescu, S. Obermeir, and Der-Yeuan Yu. "ICS Threat Analysis Using a Large-Scale Honeynet," in Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research 2015, 2015, 1-30.

[3] D. Buza, F. Juhasz, and G. Miru. "Design and implementation of critical infrastructure protection system," Budapest University of Technology and Economics, Department of Networked Systems and Services, 2013.

[4] D. Fronimos, E. Magkos, and V. Chrissikopoulos. "Evaluating Low Interaction Honeypots and On their Use against Advanced Persistent Threats," in PCI '14, Proceedings of the 18th Panhellenic Conference on Informatics, Athens, Greece, October 2-4, 2014.