# Using Cloud Honeypot Platforms for Gathering Industrial-Control-System Attack Intelligence

Cloud Honeypot Platforms

Alexander D. Washofsky

U.S. Naval Postgraduate School, adwashof@nps.edu

Neil C. Rowe

U.S. Naval Postgraduate School, ncrowe@nps.edu

Thuy D. Nguyen

U.S. Naval Postgraduate School, tdnguyen@nps.edu

Industrial control systems (ICSs) are significant targets of cyberattacks because they are often critical infrastructure with software that is difficult to update. Honeypots are essential tools for collecting cyberattack intelligence, and they are particularly useful with ICSs because of the specialized nature of these sites. Because targets often subjected to randomly varied exploits, sets of honeypots ("farms") are necessary to use them effectively. Cloud services appear to be a good place to run honeypots, but ICS systems until recently have rarely used cloud services to avoid security problems, so having ICS honeypots in the cloud might seem suspicious. Our experiments investigated whether operating electric-grid honeypots in the cloud significantly affected their traffic rates or types. It also investigated whether a more generic honeypot platform T-Pot that supports a variety of honeypot types would also be detectable by attackers, since a generic design would simplify their deployment. We ran honeypots for a year using a cloud service unaffiliated with our school, and collected all traffic directed at them by real visitors (not staged exploits). Results showed only minor differences in Web and ICS traffic for cloud honeypots, T-Pot, and different honeypot locations, confirming that such variations in deployment are an effective way to collect more data for security research on ICSs.

CCS CONCEPTS • Security and privacy - Intrusion/anomaly detection and malware mitigation • Networks - Network types - Cyber-physical networks

**Additional Keywords and Phrases:** industrial control systems, cloud computing, honeypots, platforms, testing, cyberattacks

**ACM Reference Format:**

Alex D. Washofsky, Neil C. Rowe, and Thuy D. Nguyen. 2021. Using Cloud Honeypot Platforms for Gathering Industrial-Control-System Attack Intelligence. In: ICSS 2021: ACSAC Workshop on Industrial Control System Security, December 7, 2021, Online. ACM, New York, NY, USA, 10 pages.

.

## 1 INTRODUCTION

Industrial control systems (ICSs) have become increasingly complex and interdependent. They often have a planned lifetime of 20 to 30 years and are difficult to update, so older systems with well-known vulnerabilities continue to be used [9]. Thus researchers are examining novel ways to improve their security. One method is to deploy honeypots (decoy devices and servers) to collect interaction data [17].

Cloud services are ideal for implementing honeypots [5] since they can support many simultaneous deployments efficiently. Many cyberattack campaigns choose methods randomly, and offering many honeypots permits defenders to see more variety of attacks. Organizations increasingly use cloud services to reduce their resource requirements [1]. However, ICSs in the cloud may be unconvincing to visitors because ICSs have been slow to adopt cloud services for management, and cloud services are easy to detect through public information. However with so many easy targets, attackers rarely inspect the characteristics of a site carefully, as seen in our previous honeypot research, so we should fool most visitors with simple honeypots.

This work studied power-grid (electric-grid) systems [6]. They are major users of ICSs in managing power generation, transmission, and distribution, and have increasingly been vulnerable to cyberattack. An example occurred in December 2015 when cyberattacks cut electrical power service to 230,000 people in the Ukraine.

## 2 PREVIOUS WORK

One honeypot project [13] installed a standard version of the T-Pot honeypot platform (github.security.telekom.com) on cloud-computing platforms in different geographic locations. This platform included SSH, Web, and download-management preconfigured honeypots. Results showed that the market share of the cloud-service providers did not affect the rate of visitors. [3] set up multiple T-Pot instances in different cloud services and saw no significant differences between regular servers and cloud-based systems for SSH interactions recorded by Cowrie (github.com/cowrie/cowrie) or the malware executables captured by Dionaea (github.com/DinoTools/Dionaea).

[4] deployed multiple independent honeypots in the cloud. [17] ran Dionaea and SSH honeypots over three months on both a local network and a cloud deployment. Their results suggested interactions detected with Dionaea did not differ with where the honeypot was hosted. [16] deployed an ICS honeynet, a collection of honeypots, on a cloud server simulating seven protocol services in different combinations. The researchers concluded that visitors' interest in the honeypots depended strongly on what protocols they handled.

A honeypot project at our school [10] ran the low-interaction honeypot Conpot [11] for four months within a virtual machine on a local Internet-facing server. It serviced the Web's Hypertext Transfer Protocol (HTTP) and several ICS protocols including EtherNet/IP, MODBUS, S7Communications (S7Comm), Simple Network Management Protocol (SNMP), Building Automation and Control Networks (BACnet), and Intelligent Platform Management Interface (IPMI). HTTP accounted for 56% of the traffic to this honeypot, and the MODBUS accounted for 35%. This showed that even a low-interaction honeypot could provide useful data.

An ICS honeypot project [8][15], that we have built upon for the current work, deployed a modified version of Conpot called GridPot that emulates an electrical-distribution system. GridPot replaced Conpot's low-interaction handling of the IEC 60870-5-104 communications protocol ("IEC 104" for short) with a more interactive version that communicated to a power-grid simulator called GridLAB-D [10], providing a more convincing interface for visitors. The interface was important since interfaces are the top target of attacks on ICS systems [12]. This setup was later deployed for standalone honeypots in the cloud [2] and appeared to be a workable configuration for more detailed experiments.

## 3 METHODS AND DESIGN

Our experiments reported here used both low-interaction and high-interaction research honeypots in a cloud-computing environment. The data collected was compared to that of previous honeypot deployments, and different experiments were compared to one another. More details of the experiments are in [19].

We chose to use an "Infrastructure as a service" template from DigitalOcean [7] for a cloud service so we could have the most control over honeypot operation. DigitalOcean can create servers called "droplets" with either shared or dedicated hardware, and with specific numbers of processors and amount of memory. Other machine installation options include a pre-installed operating system, block storage, a datacenter region, and automatic backup. Initial configuration of the machines is done through Digital Ocean's remote console, and once firewall rules and remote-shell parameters are set, the rest of the configuration can be done over SSH.

We installed T-Pot [18] on our DigitalOcean sites. T-Pot is a honeypot deployment platform supporting several kinds of honeypots, with tools for real-time monitoring, intrusion detection, and analysis. T-Pot uses Docker (www.docker.com), a program that can run software in virtual operating systems in instances called containers. These containers can easily be managed and configured on the host machine, and a single machine can run many Docker containers. T-Pot requires each honeypot and tool be containerized and running in separate Docker containers, which provides modularity as well as better security compared to running software directly on the machine.

Snare and Tanner are included in T-Pot. Snare accepts HTTP connections; Tanner logs and analyzes Snare events and identifies attempts to exploit a vulnerability. Snare also includes a tool to copy Web sites and supply them to visitors. The version of Snare that comes with T-Pot includes ten sample websites, and upon installation, one is randomly presented.

T-Pot supports Conpot [11], the service mentioned earlier. Conpot logs all interactions on its ports for its serviced protocols and outputs them to a log file. GridPot (github.com/gridpot) modifies and extends Conpot, simulating an HTTP server that hosts an ICS electric-grid interface showing variables of an ICS system. Instead of the low-interaction Conpot implementation of ICS protocol IEC 104 [14], GridPot includes an electric-grid simulator GridLAB-D. Our previous work [8] modified Conpot and GridPot to send incoming IEC 104 requests to GridLAB-D and return the simulated results of the request. T-Pot does not support GridPot as downloaded, and GridPot was not designed to run in a container.

Most traffic captured by our experiments was Web traffic of HTTP over TCP port 80. We also saw traffic for MODBUS, EtherNet/IP, S7, and the IEC 104 application-level protocol used in electrical and power systems. Since our time was limited, we focused on providing realistic responses to only HTTP and IEC 104.

## 4 EXPERIMENT AND ANALYSIS SETUP

We did seven experiments with three deployment methods. The experiments used DigitalOcean virtual machines pre-built with Debian Linux 10x64 images. We used virtual machines with two dedicated processors, eight gigabytes of memory, and 25 gigabytes of storage. For comparison purposes, these specifications were the same as for the previously used machines. Each had a single unique public IPv4 address; for simpler analysis, and we did not enable IPv6 connectivity. During initial set-up and installation, we remotely accessed the console of each machine using SSH from our local machine. DigitalOcean's firewall allows each machine to have individual rule sets, and this made it easy to isolate the machines during installation and testing, during which time we limited access only to our local machine's IP address.

T-Pot has six built-in templates for installation. For each experiment, we modified the T-Pot installation script to use a custom template to install the honeypot and tools required. This template used a custom Docker configuration file,

modified from the default T-Pot configuration file, to set up required containers, services, and networks. All experiments were done outside our school's firewall.

Experiment 1 compared a standalone Conpot in [10] with a similar Conpot instance running within T-Pot. Previous work [10] deployed a default configuration of Conpot in a virtual machine; our Experiment 1 used a custom T-Pot template to install a Conpot honeypot running on a DigitalOcean machine in the US. Experiments 2 and 3 compared results of [15] with the Snare and Tanner honeypots in T-Pot. Experiment 2 installed Snare and Tanner on a cloud machine, copying the Web site of previous students Chong and Koh; Experiment 3 installed Snare and Tanner in T-Pot with a custom template. Experiments 4-7 compared our experiments to [2] which included changes to Conpot and GridPot by Dougherty [8]. Experiments 4 and 5 installed GridPot and the additional tools of T-Pot on two machines in Asia and the US, and required some preparation since no support for GridPot was provided in T-Pot. We created a new Docker container to host GridPot. Experiments 6 and 7 were like Experiments 4 but without T-Pot. On each machine, a Linux service ran the TShark packet-capture tool to collect traffic on the TCP ports and push it to a central repository hourly.

The template T-Pot configuration file did not forward SNMP, BACnet, and IPMI traffic to the honeypot using UDP as it should, as Docker defaulted to forwarding using TCP. After we submitted a bug report, this issue was corrected in Issue #781 of the T-Pot project. Experiments 2-7 did not use protocols requiring UDP traffic. Another problem was that Conpot logs misclassified malformed HTTP traffic on port 80 as HTTP 0.9 packets. Furthermore, comparison of packet-capture data with Conpot log entries revealed some HTTP requests recorded by TShark absent from the Conpot log, and timestamps for some MODBUS and FTP packets were inaccurate, as many cached ones appeared in the log only during Conpot's shutdown. We thus changed our "proof of life" procedure to routinely log into the machines to confirm services were running, and to view running Docker containers using Cockpit.

We used unweighted cosine similarity to compare traffic between experiments using a vector of four items representing the counts of the main categories of traffic that we measured: HTTP GET commands, HTTP POST commands, other HTTP methods, and IEC 104 ICS traffic. The formula is $\sum_{j=1}^{4} c_{1j}c_{2j}/[\sqrt{\sum_{j=1}^{4} c_{1j}^2}\sqrt{\sum_{j=1}^{4} c_{2j}^2}]$ where $c_{ij}$ is the count of the jth component of vector i. Cosine similarity measures differences in the direction of vectors and does not reflect overall traffic volume, just the mix of the traffic, so it made sense for this traffic where volume varied widely from day to day. Several methods can provide a measure of significance of the differences in traffic between experiments. Our previous work [19] used the ratios of cosine similarity between distributions, which can be analyzed as a form of F-distribution, but the ratio distributions were often inconsistent. However, one minus the cosine similarity appeared to look like one side of a normal distribution centered on zero, useful for one-tail tests, and cosine similarity is analogous to a standard deviation between distributions. So we used a modified two-sample T-test that computes the number of standard deviations from the mean of this distribution for a pair of experiments, estimated as $2(1 - s_{ij})/(e_{si} + e_{sj})$ where $s_{ij}$ is the cosine similarity between experiments i and j, and $e_{si}$ is the standard error of cosine similarities between successive weeks within experiment i. We used the standard error rather than the standard deviation because of the bursty nature of our data, and we averaged the standard errors of the two compared distributions to give an equal weight to both.

## 5 RESULTS

### 5.1 Scanning our honeypots

The Shodan network-scanning tool (www.shodan.io) did not identify our honeypots as such when we told it to scan our principal honeypot addresses. However, the deprecated Honeyscore tool provided by Shodan did report Experiment 5 as a possible honeypot, while saying that each of Experiments 4, 6, and 7 looked like real systems; no reasons were given.

### 5.2 Overall results of experiments

For Experiment 1 we used data only from Conpot logs since [10] used them exclusively. Experiment 1 had 5,656 HTTP requests and responses using five HTTP methods. 1,079 MODBUS attempts were reported, 609 connections and 470 traffic packets, with occasional spikes in traffic on particular days. 79 EtherNet/IP packets were reported of which only 11 were properly formatted. 651 S7Comm activities were reported, including 286 connections, 157 sessions, and 208 packets. Overall, the distribution of packet types was like Hyun's [10].

Experiments 2 and 3 compared data to of the Snare and Tanner Web honeypots running without T-Pot (experiment 2) and within T-Pot (experiment 3), modifying the previous cloud deployments [2]. These experiments saw 2,644 and 2,615 different alleged IP addresses, of which 886 addresses accessed both experiments. Many addresses were on similar subnets, suggesting the same originator. We also found that a Chinese IP address was more likely to be unique to an experiment than a Russian IP address, as the number of unique IP addresses geolocating to China was 29% and 34%, and to Russia was 2% and 4%. Experiment 2 used 20-25 times less processing power, six times less memory, and four times less disk space than Experiment 3 with T-Pot. Experiment 2 received 28,042 HTTP requests and Experiment 3 received 10,619; however, 13,199 (47%) of Experiment 2's requests were from a single address on February 27th, a large outlier (Table 1).

Table 1: Experiments 2 and 3 HTTP Methods

| | *Experiment 2 without the Feb. 27 outlier* | *Experiment 3* |
|---|---|---|
| *Total HTTP Requests* | 14843 | 10619 |
| *GET* | 11372 (76.62%) | 8976 (84.53%) |
| *POST* | 3295 (22.2%) | 1436 (13.52%) |
| *HEAD* | 122 (0.82%) | 129 (1.21%) |
| *CONNECT* | 41 (0.28%) | 48 (0.45%) |
| *OPTIONS* | 13 (0.09%) | 27 (0.25%) |
| *PROPFIND* | 0 (0%) | 3 (0.03%) |

Experiments 4-7 ran one month each; however, 6 and 7 had an HTTP error and only collected seven days of HTTP traffic. Generally, the T-Pot deployments in Experiments 4 and 5 were accessed by more countries than the honeypots without T-Pot in Experiments 6 and 7. Experiment 7 received four times more traffic than Experiments 4-6, as its IP address appeared to be the former sole address for a commercial Korean website last updated in 2020.

We defined a session as all packets exchanged by a socket pair on a day [15]. We separately counted IP addresses that established only a single HTTP session, established multiple HTTP sessions, established only a single IEC 104 session, established multiple IEC 104 sessions, and established both HTTP and IEC 104 sessions (Table 2). To better compare to the limited HTTP dataset in Experiments 6 and 7, data is given for Experiments 4 and 5 to match the seven days. There were significant differences in traffic between experiments.

Table 2: Experiments 4-7 Session Data

| ***Sessions Established with Unique IP Addresses*** | ***Exp. 4 (Only first 7 days of HTTP)*** | ***Exp. 5 (Only first 7 days of HTTP)*** | ***Exp. 6 (US)*** | ***Exp. 7 (Asia)*** |
|---|---|---|---|---|
| *Number of unique IP addresses* | 320 | 353 | 305 | 1509 |
| *Single HTTP-only session* | 192 (60%) | 211 (60%) | 192 (63%) | 457 (30%) |
| *Multiple HTTP-only sessions* | 108 (34%) | 127 (36%) | 92 (30%) | 1030 (68%) |
| *Single IEC 104-only session* | 10 (31%) | 5 (1.5%) | 7 (2%) | 6 (0.4%) |
| *Multiple IEC 104-only sessions* | 7 (2%) | 6 (1.6%) | 11 (4%) | 14 (0.9%) |
| *HTTP and IEC 104 sessions* | 3 (0.9%) | 4 (1.1%) | 2 (.6%) | 2 (0.1%) |

Table 3 shows the most requested types of HTTP paths in Experiments 4-7 (Experiments 1-3 had very similar data). Many are quite familiar from reports on possible attacks on Web sites, including those related to PHP or SQL, and attempts at privilege escalation with root, manager, and login. The “Crawler” category includes “robots.txt”, “favicon.ico”, and “.env”. The “Files” category includes paths with .txt, .sh, .zip, and .tar extensions. The “Other .env” category includes paths with the string “.env” except for root-level “/.env”. “Other” includes everything, and it was especially common in Experiments 2 and 3. For Experiment 7, “DNS redirect” meant paths including the registered domain as well as paths beginning with “content/”, absent in Experiments 4-6, which were attempts to reach content on the registered domain. The frequent use of most of these paths, as well as the many malformed HTTP packets, indicates that HTTP interaction with our honeypots were mostly cyberattacks rather than routine scanning, since scanning does not need to use these suspicious-looking methods.

Table 3: HTTP Path Types Seen in Experiments 4-7

| | *Exp. 4* | *Exp. 5* | *Exp. 6* | *Exp. 7* |
|---|---|---|---|---|
| *HTTP data* | 31 days | 31 days | 7 days | 7 days |
| */* | 948 (17.9%) | 1028 (19.1%) | 262 (32.5%) | 1880 (23.3%) |
| */ ignition/execution-solution* | 46 (0.9%) | 77 (1.4%) | 14 (1.7%) | 16 (0.2% |
| */manager/html* | 15 (0.3%) | 27 (0.5%) | 3 (0.4%) | 5 (0.1%) |
| */config/getuser?index=0* | 31 (0.6%) | 22 (0.4%) | 6 (0.7%) | 6 (0.1%) |
| */login* | 11 (0.2%) | 21 (0.4%) | 2 (0.2%) | 4 (0.1%) |
| */Jenkins/login* | 8 (0.2%) | 18 (0.3%) | 1 (0.1%) | 4 (0.1%) |
| *Index.html* | 389 (7.3%) | 432 (8.0%) | 102 (12.6%) | 1781 (22.0%) |
| *Category: PHP* | 2833 (53.4%) | 2474 (46%) | 110 (13.6%) | 1100 (13.6%) |
| *Category: SQL* | 5 (0.1%) | 50 (0.9%) | 1 (0.1%) | 20 (0.3%) |
| *Category: Crawler* | 213 (4.0%) | 295 (5.5%) | 67 (8.3%) | 268 (3.3%) |
| *Category: .xml* | 177 (3.3%) | 113 (2.1%) | 31 (3.8%) | 18 (0.22%) |
| *Category: Shell commands* | 58 (1.1%) | 103 (1.9%) | 8 (0.99%) | 36 (0.45%) |
| *Category: JSON* | 71 (1.3%) | 80 (3.2%) | 22 (2.7%) | 23 (2.0%) |
| *Category: Top-level folders* | 162 (3.0%) | 224 (4.2%) | 100 (12.4%) | 55 (0.7%) |
| *Category: Files* | 77 (1.5%) | 124 (2.3%) | 19 (2.4%) | 49 (0.6%) |
| *Category: JavaScript* | 81 (1.5%) | 38 (0.7%) | 0 | 9 (0.1%) |
| *Category: Other .env* | 13 (0.2%) | 82 (1.5%) | 6 (0.7%) | 0 |
| *Other* | 172 (3.2%) | 162 (3.0%) | 54 (6.7%) | 252 (1.27%) |
| *DNS redirect* | 0 | 0 | 0 | 2615 (32.35%) |
| *Total* | 5305 | 5370 | 807 | 8081 |

Across all four experiments, IEC 104 traffic was lower than HTTP traffic. Experiments 4 and 5, running within T-Pot, received more valid IEC 104 messages than Experiments 6 and 7, while Experiments 6 and 7 attracted more visitors as measured by IP addresses. Experiments 4 and 5 were more consistent with results of [2]. Experiments 4-7 traffic with protocol IEC 104 had 78 U-format frames, 17 I-format frames, and 414 invalid ("error") frames in total, so most visitors were either not sophisticated or uninterested in IEC 104. This traffic was almost entirely likely to be cyberattacks rather than scanning, since scanning is generally knowledgeable about the protocols it uses and does not make deliberate errors. Many error frames actually contained traffic with the syntax of HTTP, FTP, or other protocols.

We observed some patterns of coordinated traffic from subnetworks, suggesting coordinated cyberattack campaigns. Table 4 shows an example where twelve addresses of a subnet sent traffic to all of experiments 4-7. Eleven addresses on the subnet sent IEC 104 traffic to some but not all experiments. Based on the GeoIP database, the IP range was in mainland France. The days of activity were Mondays, Wednesdays, and Fridays, and the hours of activity were 0200-1800 Pacific Time, corresponding to 1100-0300 Central European Time. This leaves an eight-hour gap of inactivity. If this traffic required human interaction, their awake hours are 0600-2200, and then the originating time zone may actually be GMT-4 in Brazil and may be spoofing.

Table 4: Example of coordinated subnetwork activity observed

| *End of IP address* | *Experiment 4* | *Experiment 5* | *Experiment 6* | *Experiment 7* |
|---|---|---|---|---|
| *.49.96* | | | | 1 U-Format |
| *.49.97* | 2 U-Format,<br>1 I-Format | | | |
| *.49.98* | | 2 U-Format,<br>1 I-Format | | |
| *.49.99* | | 3 U-Format,<br>1 I-Format | 1 U-Format | |
| *.49.100* | 2 U-Format,<br>1 I-Format | | | |
| *.49.101* | 2 U-Format,<br>1 I-Format | | | |
| *.49.102* | | | 1 U-Format | |
| *.49.103* | | 2 U-Format,<br>1 I-Format | | 1 U-Format |
| *.49.104* | | | | 1 U-Format |
| *.49.109* | 2 U-Format,<br>1 I-Format | | 1 U-Format | |
| *.49.110* | | | 1 U-Format | |

### 5.3 Comparison of traffic in experiments

So far we have presented analysis of individual experiments, but we also compared the mix of packet types within experiments and between experiments using cosine similarities. The standard errors of the similarities ($e_s$) between successive weeks were 0.794 for [11], 0.715, 0.774, and 0.994 for the first three experiments of [2], and 0.830, 0.505, 0.717, and 0.602 for Experiments 4-7 respectively; we used their average of 0.741 for the standard error for the other experiments for which we lacked data. Table 5 shows the inter-experiment cosine similarities and the degree of significance as computed by the formula $2(1 - s_{ij})/(e_{si} + e_{sj})$ described in section 4; the two numbers in each entry are the overall similarity of traffic and the degree of significance. The most anomalous traffic was in experiments "Do" and "Wa7". Although no differences had a degree of significance exceeding 0.46, well below the values of 2 or 3 times the standard deviation often used in statistics, the first and last experiments may be showing occasional different effects on visitors of the honeypots. The differences in the first experiment could be due to the different configuration [8], and the differences in the last experiment are likely due to traffic for the previous use of the IP address as mentioned. Overall, the small differences between experiments confirm that use of cloud services, T-Pot, or varied locations for the honeypots affected their traffic very little, an encouraging result for future development of ICS honeypots.

Table 5: Confusion matrix of cosine similarities and degree of significant difference for seven of our experiments (Wa) [19], three experiments by [2] (Bp), and one experiment by [11] (Do)

| | *Do* | *Bp1* | *Bp2* | *Bp3* | *Wa1* | *Wa2* | *Wa3* | *Wa4* | *Wa5* | *Wa6* | *Wa7* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Do* | 1 / .00 | .809 / .25 | .955 / .06 | .712 / .32 | .759 / .31 | .836 / .21 | .763 / .31 | .699 / .37 | .953 / .07 | .737 / .35 | .965 / .05 |
| *Bp1* | | 1 /.00 | .939 / .08 | .988 / .01 | .994 / .01 | .998 / .00 | .997 / .00 | .985 / .02 | .947 / .09 | .985 / .02 | .767 / .35 |
| *Bp2* | | | 1 /.00 | .876 / .14 | .913 / .16 | .950 / .07 | .910 / .23 | .872 / .16 | .999 / .00 | .899 / .14 | .938 / .09 |
| *Bp3* | | | | 1 /.00 | .994 / .01 | .980 / .02 | .997 / .00 | .998 / .00 | .888 / .15 | .983 / .02 | .667 / .42 |
| *Wa1* | | | | | 1/.00 | .988 / .03 | .997 / .00 | .995 / .01 | .920 / .13 | .986 / .02 | .733 / .40 |
| *Wa2* | | | | | | 1 /.00 | .992 / .01 | .974 / .03 | .960 / .06 | .971 / .04 | .786 / .32 |
| *Wa3* | | | | | | | 1.0 /0 | .994 / .01 | .921 / .13 | .984 / .02 | .718 / .42 |
| *Wa4* | | | | | | | | 1 /.00 | .882 / .18 | .889 / .01 | .667 / .46 |
| *Wa5* | | | | | | | | | 1 /.00 | .902 / .16 | .924 / .14 |
| *Wa6* | | | | | | | | | | 1 /.00 | .722 / .42 |
| *Wa7* | | | | | | | | | | | 1 /.00 |

## 6 CONCLUSIONS

We simulated ICS services for the HTTP and IEC 104 protocols, configured honeypots with Conpot, GridPot, and Gridlab-D to provide a realistic ICS interface, and collected data on real interactions. Overall, our experiments did not show a significant difference in Web and ICS traffic received by honeypots simulating ICS sites as deployed in the cloud, within T-Pot, and in the U.S. versus in Asia. We also concluded that most interactions with our honeypots appear to be cyberattacks and not scanning, for both Web (HTTP) and ICS (IEC 104) traffic. These results are encouraging for future deployments of ICS honeypots as it supports using them with greater flexibility, and that flexibility should enable better deception of possible attackers. A further encouraging result was that our T-Pot deployment was not fingerprinted by a well-known network scanning tool.

T-Pot proved a good honeypot-management platform with many capabilities and support for customization. It also worked well as a base for establishing an industrial-control-system honeypot, our primary interest, although we saw it

worked well for Web traffic and can do much more. Although its data-analytic tools rely on sometimes incomplete log data, packet-capture software with offline analysis can compensate for this problem.

Future work includes running the honeypots for longer intervals and in other more-hardened frameworks. Machine learning will be used to analyze the data and look for evidence of more serious threats.

## ACKNOWLEDGEMENTS

This work was supported by the NPS Foundation, the National Science Foundation under the Scholarship for Service program, and the Department of Energy through Idaho National Laboratories. Views expressed are those of the authors and do not necessarily represent those of the U.S. Government.

## LIST OF REFERENCES

[1] Basu, S. et al. (2019). Cloud computing security challenges & solutions - A survey. *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 347-356. doi:10.1109/CCWC.2018.8301700

[2] Bieker, M., & Pilkington, D. (2020). Deploying an ICS Honeypot in a Cloud Computing Environment and Comparatively Analyzing Results against Physical Network Deployment. *(Master's thesis, Naval Postgraduate School)*. NPS Archive: Calhoun. Retrieved from http://hdl.handle.net/10945/66586

[3] Bove, D. (2018). Using Honeypots to Detect and Analyze Attack Patterns on Cloud Infrastructures. *(Master's thesis, Friedrich-Alexander University)*. Bavaria, Germany. Retrieved from https://davidebove.com/files/thesis-bove-public.pdf

[4] Brown, S., Lam, R., Prasad, S., Ramasubramanian, S., & Slauson, J. (2012, December 19). Honeypots in the Cloud. Madison: Unversity of Wisconsin - Madison. Retrieved from http://pages.cs.wisc.edu/~sbrown/downloads/honeypots-in-the-cloud.pdf

[5] Campbell, R., Padayachee, K., & Masombuka, T. (2015). A survey on honeypot research: Trends and Opportunities. *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, 208-212. doi:10.1109/ICITST.2015.7412090

[6] Chassin, D. et al. (2008). An open-source power systems modeling and simulation environment. *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, 1-5. doi:10.1109/TDC.2008.4517260

[7] DigitalOcean, LLC. (n.d.). *Droplets*. Retrieved July 15, 2021, from https://docs.digitalocean.com/products/droplets

[8] Dougherty, J. (2020). Evasion of Honeypot Detection Mechanisms through Improved Interactivity of ICS-Based Systems. *(Master's thesis, Naval Postgraduate School)*. NPS Archive: Calhoun. Retrieved from http://hdl.handle.net/10945/66065

[9] Hurd, C. M., & McCarty, M. V. (2017). *A Survey of Security Tools for the Industrial Control System Environment.* Idaho Falls: Idaho National Laboratory.

[10] Hyun, D. (2018). Collecting cyberattack data for industrial control systems using honeypots. *(Master's thesis, Naval Postgraduate School)*. NPS Archive: Calhoun. Retrieved from http://hdl.handle.net/10945/58316

[11] Jicha, A. et al. (2016). SCADA honeypots: An in-depth analysis of Conpot. *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 196-198. doi:10.1109/ISL.2016.7745468

[12] Kaspersky (2015). Industrial Control Systems Vulnerabilities Statistics. Retrieved from https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2016/07/07190426/KL_REPORT_ICS_Statistic_vulnerabilities.pdf.

[13] Kelly, C., Pitropakis, N., Mylonas, A., McKeown, S., & Buchanan, W. (2021). A Comparative Analysis of Honeypots on Different Cloud Platforms. *Sensors, 21*(2433). doi:10.3390/s21072433

[14] Matousek, P. (2017). *Description and Analysis of IEC 104 Protocol.* Brno University of Technology.

[15] Rowe, N., Nguyen, T., Dougherty, J., Bieker, M., and Pilkington, D., Identifying anomalous industrial-control-system network flow activity using cloud honeypots. Proc. National Cyber Summit, Huntsville, Alabama, Springer, September 2021.

[16] Serbanescu, A., Obermeier, S., & Yu, D.-Y. (2015, September). ICS Threat Analysis Using a Large-Scale Honeynet. *3rd International Symposium for ICS & SCADA Cyber Security Research 2015*, 20-30. doi:10.14236/ewic/ICS2015.3

[17] Sochor, T., & Zuzcak, M. (2014, February). Study of internet threats and attack methods using honeypots and honeynets. *International Conference on Computer Networks*, 118-127. doi:10.1007/978-3-319-07941-7_12

[18] Telekom Security. (n.d.). *T-Pot*. Retrieved July 15, 2021, from https://github.com/telekim-security/tpotce

[19] Washofsky, A. (2021). Deploying and analyzing containerized honeypots in the cloud with T-Pot. M.S. thesis, U.S. Naval POstgraduate School, September 2021.