Characterizing Background Noise in ICS Traffic Through a Set of Low Interaction Honeypots

Pietro Ferretti
Politecnico di Milano
Milan, Italy
pietro1.ferretti@mail.polimi.it

Marcello Pogliani
Politecnico di Milano
Milan, Italy
marcello.pogliani@polimi.it

Stefano Zanero
Politecnico di Milano
Milan, Italy
stefano.zanero@polimi.it

ABSTRACT

Industrial Control Systems (ICS) are nowadays interconnected with various networks and, ultimately, with the Internet. Due to this exposure, malicious actors are interested into compromising ICS — not only for advanced and targeted attacks, but also in the context of more frequent network scanning and mass exploiting of directly Internet-exposed devices. To understand the level of interest towards Internet-connected ICS, we deploy a scalable network of low-interaction ICS honeypots based on the popular `conpot` framework, integrated with an analysis pipeline, and we analyze the in-the-wild traffic directed through a set of ICS-specific protocols. We present the results of running our honeypots for several months, showing that, although most of the traffic is originated by known, legitimate network scanners, and follows patterns similar to those of well-known ICS network mapping scripts, we found several requests from unknown actors that do not follow this pattern and may hint at malicious traffic.

CCS CONCEPTS

• **Security and privacy** → *Intrusion/anomaly detection and malware mitigation*;

KEYWORDS

industrial control systems, honeypots

ACM Reference Format:
Pietro Ferretti, Marcello Pogliani, and Stefano Zanero. 2019. Characterizing Background Noise in ICS Traffic Through a Set of Low Interaction Honeypots. In *ACM Workshop on Cyber-Physical Systems Security & Privacy (CPS-SPC '19), November 11, 2019, London, UK*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3338499.3357361

1 INTRODUCTION

Industrial Control Systems are used to control and monitor a variety of cyber-physical systems, ranging from buildings, to environmental data, to the production of factories and plants, to nation-critical systems such as the power grid and critical infrastructure. Such systems were originally designed to work in closed networks, often under strong real-time requirements, making reliability and safety the most important concerns—but offering few to no security mechanisms. Despite this, ICS are nowadays connected with other networks and, ultimately, with the Internet, to provide features such as real-time remote monitoring, remote plant maintenance and control, as well as the collection of cyber-physical data for analysis purposes in third-party and cloud-based systems. This shift opened up security concerns with the exposure of critical (and less critical) cyber-physical systems on the Internet: Shodan[1], a popular Internet scanning engine, lists more than 10,000 endpoints running a variety of the BACnet protocol and more than 13,000 endpoints exposing Modbus/TCP. This exposure has sparked the interest of malicious actors and researchers alike in performing scanning, reconnaissance and—in some cases—attacks targeted at ICSes.

In this paper, we look at the "background noise" ICS traffic, i.e., at the ICS traffic that is not specifically targeted to a particular organization or as part of a multi-stage attack, but aimed at directly Internet-exposed ICS. Specifically, we deploy a network of honeypots on a variety of network vantage points and simulating a variety of ICS devices, and we analyze the captured well-formed ICS traffic. Unlike previous work (e.g., [12]), we use low-interaction honeypots, rather than data from a network telescope, to analyze complete ICS protocol requests rather than just connection attempts, and we present the results of a comprehensive analysis. Furthermore, as our aim is to analyze "background noise" traffic rather than targeted attacks, we deem low interaction honeypots effective, rather than resorting to the simulation of complex networks and to the simulation of the underlying physical process as performed by [1, 2]. The use of low interaction honeypots, rather than replica of real devices, allows for an easy deployment on a large number of network vantage points and for a deployment of multiple protocols, rather than being constrained by the physical device.

In summary, we present the following contributions:

- We propose a scalable low-interaction honeypot architecture, augmented with an automated extensible analysis pipeline;
- We deploy a set of honeypots in various configurations (different ICS protocols and different network vantage points) resembling real ICS devices;
- We analyze the results of running our honeypots for several months, providing a comprehensive analysis of "background noise" traffic for ICS protocols.

In particular, we identify the actors generating the traffic, confirming previous findings [12] that most background ICS traffic is generated by a few regular scanners, of which many are hard-to-identify

CPS-SPC '19, November 11, 2019, London, United Kingdom

ACM ISBN 978-1-4503-6831-5/19/11…$15.00
https://doi.org/10.1145/3338499.3357361

[1] https://www.shodan.io

actors. We expand on previous work by describing interesting behavior of specific actors (e.g. scanning only on selected ranges and classes of IP addresses). We also group actors by looking at specific scan "fingerprints", gaining a few insights about scanning tools.

2 ARCHITECTURE

To analyze and understand the ICS-targeted traffic, we deploy a set of low-interaction ICS honeypots on multiple network vantage points, and we design an automated pipeline to collect and analyze the captured network traffic data.

2.1 ICS Honeypot

We are interested in simulating ICS devices, such as PLCs, that are by mistake or intentionally exposed directly on the Internet. To this end, we deploy a set of low-interaction honeypots that expose a set of representative ICS protocols. Our honeypots are based on `conpot`[2], one of the most popular open-source ICS honeypots, and we consider the following representative set of protocols[3]: Siemens S7, Modbus/TCP, EtherNet Industrial Protocol (EtherNet/IP) for process automation; IEC-61850-104 for the power grid; BACnet as a building automation protocol.

Out of the box, honeypots generated by `conpot` are extremely easy to recognize and fingerprint, hindering their purpose. We tackle this issue at two levels. On the one hand, we extended `conpot` to improve the accuracy in the implementation of ICS protocols: for example, we reimplemented the BACnet protocol and we improved the implementation of EtherNet/IP, IEC-104 and Siemens S7 to more faithfully resemble real devices, by removing artefacts unique to `conpot` that could be used for fingerprinting the honeypot. On the other hand, we implemented a set of `conpot` "templates" (i.e., the configuration of a specific honeypot instance) by looking at the most common device models and attempting to reproduce their behavior in a black-box like fashion. We searched for exposed devices using Shodan, and we directly collected data sending well-formed ICS protocol requests. As sending actual requests to real devices may cause unintended side effects, we carefully selected a small set of representative requests to collect information from the remote devices. The requests have the only effect of asking for device information, without altering the device state or performing any actual action. We verified with Shodan that our implementation was not trivially distinguishable from the actual device, and, in particular, that it was not flagged by Shodan as a honeypot: indeed, we checked all our honeypot instances on the Shodan Honeyscore service[4], which rates how much a host looks like a honeypot by fingerprinting common honeypots. The service gives a score that goes from 0.0 to 1.0, where 0.5 is the threshold of a likely honeypot; we verified that all of our instances get a score below 0.2.

2.2 Analysis Pipeline

The overall architecture of our analysis pipeline is depicted in Figure 1. The pipeline is divided in three main steps: network traffic parsing, data enrichment, and data analysis.

[2]https://www.conpot.org

[3]Our observations refer only to a subset of relatively open ICS protocols, when exposed to the Internet; we remark that in most cases, ICS communicates through internal networks, and several ICS protocols are closed and proprietary (e.g., DCS protocols).

[4]https://honeyscore.shodan.io/

Parsing. Parsing network traffic data is the most resource expensive part of the pipeline. We regularly download the captured network traffic from the honeypots and we use Pyshark [4], a Python wrapper for the Tshark network protocol analyzer, to parse raw traffic (stored in `pcap` files) and extract only the information useful for our analysis: source IP addresses, the transport-layer protocol, the target port, the application-layer request type and parameters. We store the parsed data in a JSON file for each parsed `pcap` file to make it easy to load later. This makes parsing easier to parallelize, and simplifies any future manual inspection of traffic data.

Enrichment. This step enriches the collected data with information retrieved from external sources, useful to identify the actors. We include the DNS PTR records, Autonomous System (AS) information, as well as the country of origin of the source IP addresses.

Analysis. The analysis uses both parsed data and enrichment information. We implemented the analysis as a set of distinct Python modules, which can run independently and concurrently, and can output intermediate results to be used in subsequent phases. The analysis outputs data in CSV format for easy visualization and reuse in scripts. We developed the following analyses, providing the results that we describe in Section 4:

- *Statistics*: We compute general statistics about the traffic we received: the number of SYN packets, interactions, the number of well-formed requests and well-formed interactions, aggregated by protocol and instance. We also measure the amount of attempted connections made by non-targeted port scans we received on closed ports.
- *Actor identification*: We group IP addresses in distinct actors, using DNS PTR records and Autonomous System information. We then recompute the previous requests statistics, now aggregated by actor, to understand how each actor interacts with the honeypots. We manually check actors to identify public scanners, and further investigate the actors we cannot identify using IP reputation lookup services like VirusTotal[5].
- *Countries*: After having found public and unknown actors, we aggregate unknown actors and the number of requests they made by country, to obtain additional insights on the origin of these unknown actors.
- *Actor behavior*: We investigate how each actor behaves. To do so, we collect many pieces of information: scanned ports, scanned instances, periods of time when the actor has appeared, requests made in each interaction. We then manually check for any interesting behavior.
- *Types of requests*: We run an automated analysis to extract the application-layer types of requests made for each protocol, and compute their relative frequency. We also manually investigate if there are any unique or anomalous requests.
- *Classification of scanning scripts*: After we noticed that we can use some of the fields in the ICS requests to fingerprint versions of scanning scripts, we prepared an automated analysis to classify the requests we received. We defined fingerprinting criteria for part of the protocols, and automatically classified requests and actors to look for any combination of actors that make the same types of requests.

[5]https://virustotal.org

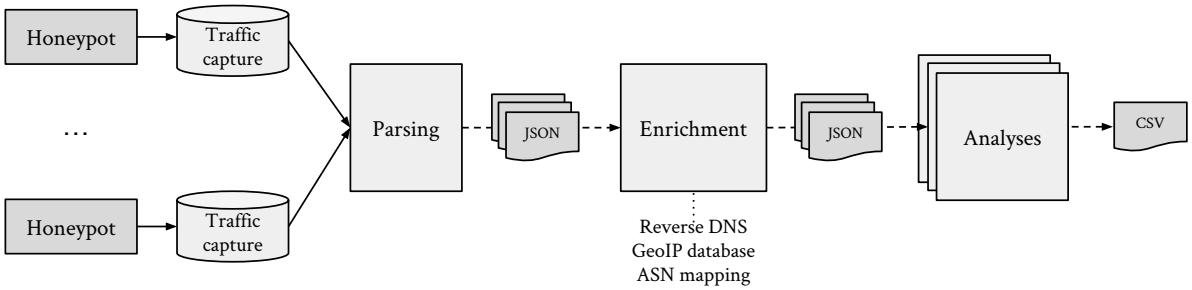

Figure 1: The pipeline used to analyze the data.

3 HONEYPOT DEPLOYMENT

We deployed the honeypot instances using Docker. All instances share the same base `conpot` Docker image; they are differentiated by loading templates and configurations from mounted volumes. The deployment is modular: each honeypot instance is deployed alongside a Docker container using `tcpdump` to collect traffic data.

Templates. We prepared the following templates by replicating actual devices found on Shodan:

- Siemens S7
  - "s7-300-a": Siemens S7-300 PLC, with module type name "CPU 314".
  - "s7-300-b": Siemens S7-300 PLC, with module type name "CPU 317F-2 PN/DP".
- Modbus/TCP
  - "abb-modbus": ABB Stotz Kontakt PLC.
  - "schneider-modbus": Schneider Electric BMX P34 PLC.
- IEC-104
  - "iec104-0": generic IEC-104 device, giving sensor readings for electric voltages.
- Ethernet/IP
  - "ab-micrologix-1100": Allen-Bradley MicroLogix 1100 PLC.
  - "ab-micrologix-1400": Allen-Bradley MicroLogix 1400 PLC.
  - "schneider-automation-pm55xx": Schneider Automation PM55XX PLC.
- BACnet
  - "machprosys-bacnet": Reliable Control Corporation MACH-ProSys BACnet controller.

VPN infrastructure. To simplify the deployment and management of the honeypots, we implemented a VPN-based architecture that redirects all the traffic from publicly exposed hosts to a central host that can be easily managed and has enough computational resources to host multiple honeypots. Given a template, we automated the creation of the honeypot and the configuration of the remote host to correctly forward the traffic through the VPN, thus allowing a scalable deployment of multiple honeypots throughout multiple networks. This architecture is similar to the one proposed by Guarnizo et al. [5], who use SSH tunneling to implement a similar configuration; however, using a VPN rather than SSH tunneling, we are able to forward all the traffic at the network layer. The architecture, depicted in Figure 2, involves the following components:

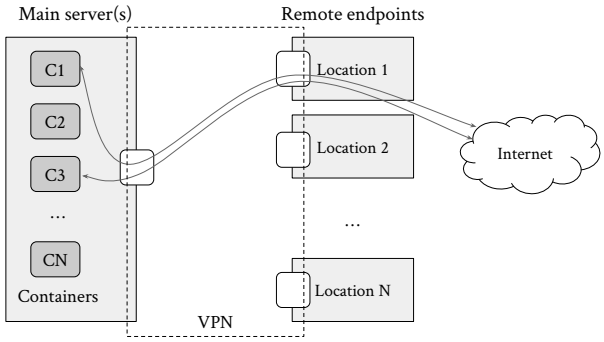

Figure 2: Deploying honeypots through a VPN.

- **Main host**: a centralized host that handles the honeypot instances and the VPN connections.
- **Remote hosts**: the hosts where the public IP addresses are actually accessible. Each remote host is connected to the main host with the VPN; all traffic received on the public IP address is redirected via a destination NAT rule to the VPN IP address assigned to the Docker container of the related honeypot instance, leaving the source IP address intact to allow traffic capture in the main host.
- **Honeypot instances**: the honeypot instances which are to be exposed on the public IP addresses. The instances are deployed on the main host: each honeypot instance is deployed in a dedicated Docker container, and mapped to a specific remote host (we configure the appropriate routing rules to make sure that all traffic leaving each Docker container gets routed through the VPN to the correct host).

Instances. We deployed the following honeypot instances:

- 11 honeypots hosted on a range of IP addresses assigned to a research organization in a research campus network.
  - Two honeypots emulating Modbus/TCP devices. Templates: *abb_modbus*, *schneider_modbus*.
  - Two honeypots emulating Siemens S7 devices. Templates: *s7_300_a*, *s7_300_b*.
  - Three honeypots emulating Ethernet/IP devices. Templates: *ab_micrologix_1100*, *ab_micrologix_1400*, *schneider_automation_pm55xx*.

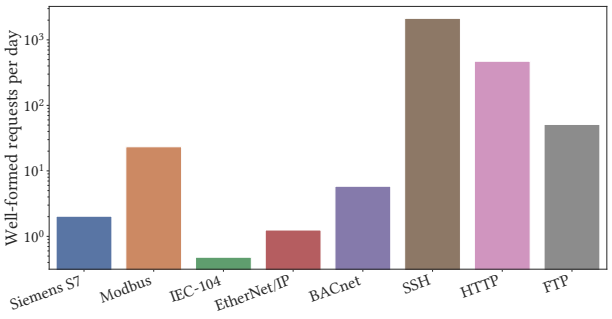

Figure 3: Well-formed requests per day. ICS protocols are much less scanned for than more common protocols.

  - One honeypot emulating a BACNet device. Template: *machprosys_bacnet*.
  - One honeypot emulating a IEC104 device. Template: *iec104*.
  - One honeypot emulating a NAS web interface, acting as a HTTP and FTP baseline.
  - One plain host to gather baseline traffic for SSH.
- 20 honeypots on cloud hosting. 10 honeypots in the United States region, 10 in the Asia region, each with the following set of configurations:
  - Two honeypots emulating Modbus/TCP devices. Templates: *abb_modbus*, *schneider_modbus*.
  - Two honeypots emulating Siemens S7 devices. Templates: *s7_300_a*, *s7_300_b*.
  - Two honeypots emulating Ethernet/IP devices. Templates: *ab_micrologix_1100*, *ab_micrologix_1400*.
  - One honeypot emulating a BACnet device. Template: *machprosys_bacnet*.
  - One honeypot emulating a IEC104 device. Template: *iec104*.
  - One honeypot emulating a NAS web interface, acting as a HTTP and FTP baseline.
  - One plain host to gather baseline traffic for SSH.

This set of deployed instances is designed to cover a diverse number of IP addresses, in different classes (academic, cloud), in different geographical locations (Italy, United States, Asia), with a wide range of templates emulating different devices. We slightly modified the templates for each deployment instance to make it harder for different instances to be recognized as honeypots: each instance has different configurations and values (i.e., serial numbers and device names).

4 RESULTS

In this Section, we show the results obtained by operating our honeypots for 4 months, from February 1st 2019 to June 21st 2019. We received, in total, 4, 986 ICS connections, of which 1, 359 Siemens S7, 1, 332 Modbus/TCP, 222 IEC-104, 1, 584 EtherNet/IP, and 489 BACnet.

Table 1 presents some aggregate statistics of the connections recorded by our honeypot. Statistics for each protocol are computed considering packets directed to the ports assigned to each ICS protocol, i.e., Siemens S7 (TCP 102), Modbus/TCP (TCP 502), IEC-104 (TCP 2404), EtherNet/IP (TCP 44818, UDP 44818), BACnet (UDP 47808), and considering only the honeypot instances that expose that protocol. We consider the following metrics:

- The number of TCP SYN and UDP packets: we use this metric to estimate the amount of traffic received for each protocol.
- The number of interactions, where an interaction includes all the requests made by a source IP address on a specific instance in 24 hours. This metric is designed to let us see how many actors are actually targeting each instance, ignoring the amount of requests made in a single interaction.
- The number of well-formed requests. We define a well-formed requests any request matching the protocol assigned to the destination port (e.g., HTTP requests to the TCP 502 port (assigned to Modbus/TCP), are considered not well-formed).
- The number of well-formed interactions. Analogous to plain interactions, we group all well-formed requests in a single interaction when made by a single IP address to the same instance in 24 hours.

We also list as a "baseline" the aggregate of the number of connections and interactions on hosts that do not implement the protocol, in order to estimate the untargeted traffic, i.e., connections and scans that are performed regardless of whether the device employs the specific protocol, and the targeted traffic, e.g., from actors who verified via other means (e.g., Shodan or previous port scans) that the protocol is actually implemented.

We find that, compared to previous work ([12]), the extent of ICS scanning has generally grown, hinting to an increased interest in targeting ICS protocols. As expected, the amount of traffic we received for the ICS protocols is far lower than the amount of scanning for standard and widespread protocols such as SSH, HTTP and FTP (as we can see in Figure 3).

All protocols have a larger number of requests than interactions, since a single interaction can include many requests. This is more noticeable for the Modbus/TCP and BACnet, as the standard scans for these protocols involve making many requests for different device units and properties. Furthermore, there are differences between scanning activity for different protocols: For instance, Modbus and Siemens S7 receive far more traffic than IEC-104. We guess that Modbus and Siemens S7 are interesting for a wider variety of actors as they are usually used to control any type of PLC, while the IEC-104 protocol is specific to power grids.

Many requests are not well-formed. This is mostly due to scans for protocols different from the ones we are considering, e.g., scans for standard IT protocols like RDP, HTTP, LDAP, Samba, SSDP, SIP, SSH on ICS ports. These requests originate from scanners and botnet workers that indiscriminately scan all ports. We did not find any malformed ICS-specific request, except the BACnet requests sent by the China Hangzhou actor: the requests are all missing one byte, making the request invalid. This error does not seem to be intended to exploit a specific vulnerability, as the rest of the request is identical to the usual scans.

Table 1: Aggregate data of the received ICS traffic (TCP SYN and UDP packets, interactions, well-formed requests, and well-formed interactions per day, respectively).

	Siemens S7	Modbus	IEC-104	EtherNet/IP	BACnet
Baseline SYN, UDP	1.84	2.28	0.71	2.02	1.73
Baseline inter.	1.58	1.64	0.63	1.99	1.35
SYN and UDP	7.58	33.22	3.01	3.62	9.22
Interactions	2.36	2.00	0.79	1.92	1.43
WF requests	2.42	27.95	0.83	1.12	8.95
WF interactions	0.66	1.24	0.33	0.99	1.40
Mirian et al. [12]	1.98	1.40	–	–	0.37

4.1 Actors

Across the traffic we received, we have seen requests to the ports assigned to ICS protocols from 1,469 distinct IP addresses, of which 832 (57%) made at least a well-formed request to one of the supported ICS protocols. To understand the actors behind the requests, we group IP addresses using reverse DNS (PTR) records, aggregating by top-level domain (e.g., "shodan.io"); if no DNS PTR record is present, we group IP addresses by Autonomous System (AS). We also check if the port TCP 80 is open on the source IP addresses, in case a web page is present to signal and explain scanning activity.

Among the IP addresses we recorded, we identify 97 distinct actors. Only 44 of these actually made any well-formed ICS requests, of which 23 for Siemens S7, 28 for Modbus, 14 for IEC-104, 20 for EtherNet/IP, 21 for BACnet. Only 7 actors made well-formed requests for all the supported protocols.

Many actors are public scanners, i.e., legitimate research organizations or companies who perform periodic internet-wide scans for research purposes. Most public scanners scan a wide variety of network ports, and interact with the supported protocols to collect basic information. Among the public scanners, Beijing University of Telecommunication performed only connections to the S7comm port, with no connection attempt on any other ports; instead, the other scanners performed ICS requests as part of a wider range of ICS and non-ICS protocols. The public scanners we identified and made well-formed requests for at least one of the five protocols we are considering are reported in Table 2. The remaining actors are not easily attributable to individuals or organizations: They all have no reverse DNS record, and the IP addresses are assigned to private ISPs or cloud hosting services.

We list the most prominent actors (both public scanners and unknown actors) attempting connections to ports assigned to ICS protocols in Table 3, and actors that actually make well-formed ICS requests in Table 4.

4.2 Countries

Public scanners make often use of cloud hosting services to run scans. We therefore ignore connections made by public scanners in this analysis, since the origin of the connections is already known.

Considering only the ICS connections not made by known scanners, we find that 60% of connections come from the AS assigned to Blackhost, a United States based bulletproof hosting service.

If we exclude all cloud and hosting services, since they give no information about the country of origin of the actual actor, we find that 89% of the connections made by unknown actors have source IP addresses located in China, 5% from Vietnam, 3% from the United States. The remaining fraction of the connections are distributed among a handful of countries: Germany, Bulgaria, Hong Kong, Singapore, South Korea, the Netherlands and the United Kingdom. If we exclude all cloud and hosting services, since they give no information about the country of origin of the actual actor, we find that 90% of the connections made by unknown actors have source IP addresses located in China, 4% from Vietnam, 3% from the United States. The remaining fraction of the connections are distributed among a handful of countries: Germany, Bulgaria, Hong Kong, Singapore, South Korea, the Netherlands and the United Kingdom. The connections from China are equally distributed across a range of ASs owned by private ISPs. As the requests received from the different Chinese ASs are different (Section 4.6), we classify them as distinct actors.

Connections received from other countries are very few, and usually limited to a handful of connections per source IP address.

4.3 Actors Behavior

We notice different behaviors among the actors we identified.

Recurrent scanners. Many actors remain present over long periods of time and perform scans regularly, sometimes with fixed intervals. Other instead appear only once, scan one or multiple honeypot instances, then disappear and never perform any other request. We call the first ones "recurrent" scanners and the second "occasional" scanners. An overwhelming majority of requests are made by recurrent scanners: 83% of the ICS connections we received were made by four actors alone (Net Systems Research, Blackhost, Shodan, Censys). If we consider the top 10 actors, we reach 92% of the connections. The distribution of connections among actors can be seen in Figure 4. A much smaller part of the requests we collected are from occasional scanners, which we presume to be individuals or organizations doing targeted scans.

We notice that only public scanners do periodic scans at fixed intervals; Shodan, Censys, F-Secure and Kudelski Security perform weekly scans, while Stretchoid and Rapid7 appear monthly, but none of the actors that are unidentified show any clear periodic activity. Despite not having periodical characteristics, several unidentified actors show nonetheless recurrent behavior, scanning reliably for long periods of time. Among these types of actors we see a good portion of the unidentified actors from Chinese ISPs, specifically: China Telecom (AS4134), China Sichuan, China UCloud Beijing, China Telecom Chengdu. On the other hand, most of the requests generated from cloud IP addresses appear in short-lived bursts that do not last more than one or two days, targeting all our honeypot instances; this is consistent with what we would expect for actors that are renting cloud hosts to launch one-time wide-range scans.

Campaigns over time. Some of the actors doing recurrent scans appear and disappear over time: we see Kudelski Security doing ICS scans only up to April, then stopping; Alphastrike appears in

Table 2: Public scanners making well-formed ICS requests

Name	Autonomous System	Scanned Protocols
Alphastrike	AS25504	BACNet, EtherNet/IP, S7, Modbus
Beijing University of Telecommunications	AS4538	S7
Binaryedge	AS14061, AS63949	EtherNet/IP, S7, Modbus
Censys	AS237	BACNet, S7, Modbus
F-Secure (Inverse Path)	AS42708	S7
Kudelski Security	AS42570	BACNet, Modbus
Net Systems Research	AS36351, 50562, 60781	BACNet, EtherNet/IP, Modbus
Onyphe	AS12876, 16276, 63949	Modbus
Rapid7 (Project Sonar)	AS10439, 13213, 29302	BACNet
Shodan	AS10439, 174, 29073, 32475, 50613, 9009	BACNet, EtherNet/IP, S7, Modbus, IEC-104
Stretchoid	AS14061	S7, Modbus

Table 3: Distribution of attempted connections per actor.

	Siemens S7	Modbus	IEC-104	EtherNet/IP	BACnet	Overall
Alphastrike	0.42%	0.55%		0.32%	0.52%	0.42%
Binaryedge	10.46%	2.99%	5.49%	1.94%		3.39%
Hosting: Blackhost		26.94%		36.91%	32.91%	25.30%
Hosting: Capitalonline		0.26%	1.90%			0.19%
Hosting: Carinet	0.30%	0.60%	0.53%	0.17%	0.26%	0.34%
Censys	11.99%	10.00%			10.34%	6.63%
ISP: China Sichuan	0.97%	0.63%	2.64%	0.60%	0.74%	0.83%
ISP: China Telecom (AS4134)	3.05%	1.96%	5.28%	1.75%	1.23%	2.13%
ISP: China Telecom (AS23724)	0.34%	0.16%	0.53%	0.02%	0.23%	0.18%
ISP: China Telecom Chengdu	2.20%	1.28%	4.96%	1.19%	1.52%	1.68%
ISP: China Telecom Jiangsu	0.85%					0.13%
Hosting: China UCloud Beijing	3.39%	2.33%	9.08%	2.72%	1.42%	2.86%
Hosting: China UCloud Shangai	1.61%	1.15%	2.43%	0.73%	0.19%	0.98%
Hosting: DigitalOcean	0.80%	0.92%	2.01%	0.41%		0.62%
F-Secure	1.86%					0.30%
Hosting: Google Cloud	7.67%	0.50%				1.35%
Ipip	5.34%					0.85%
Kudelski Security		2.72%			3.10%	1.35%
Hosting: Leaseweb			0.74%			0.05%
ISP: MCI Comm. Services	0.55%	0.34%	1.48%	0.30%		0.36%
Media Land LLC	0.25%	0.18%	0.84%	0.11%		0.18%
Hosting: Megaservers.de	1.57%	0.39%	2.32%	0.04%		0.51%
NetSystemsResearch		23.75%		35.07%	28.78%	23.04%
Hosting: OVH	1.27%	0.47%				0.32%
Onyphe	4.36%	3.19%				1.51%
ISP: Quasi Networks	0.72%	0.31%	0.84%	0.24%		0.32%
Rapid7					3.75%	0.78%
Hosting: Selectel	3.64%	2.23%	2.43%			1.31%
Shodan	30.07%	10.66%	50.69%	14.07%	12.37%	17.72%
Hosting: Softlayer		1.28%		1.86%	1.71%	1.27%
Stretchoid	0.59%	0.37%				0.19%
ISP: Vietnam CHT	0.38%	0.18%	0.95%	0.17%	0.23%	0.27%
Others (<0.5% for all protocols)	5.04%	3.51%	4.33%	1.36%	0.71%	2.55%

Table 4: Distribution of well-formed connections per actor.

	Siemens S7	Modbus	IEC-104	EtherNet/IP	BACnet	Overall
Alphastrike	0.69%	0.13%		0.04%	0.52%	0.29%
Hosting: Amazon AWS		0.51%			0.03%	0.07%
Baidu Netcom	0.69%		1.89%	0.08%	0.10%	0.13%
Beijing Univ. of Telecom.	0.69%		0.94%	0.04%		0.06%
Binaryedge	5.84%	3.67%		0.84%		0.99%
Hosting: Blackhost		33.00%		40.26%	33.33%	34.00%
Hosting: Capitalonline		0.51%	5.66%			0.15%
Hosting: Carinet	0.34%	0.13%	0.94%	0.27%	0.26%	0.26%
Censys	7.56%	4.80%			10.47%	5.54%
ISP: China Hangzhou Alibaba	1.03%	0.63%	1.89%			0.15%
ISP: China Sichuan	3.44%	2.02%	9.43%	1.07%	0.75%	1.27%
ISP: China Telecom (AS4134)	5.84%	2.40%	13.21%	1.99%	0.07%	1.52%
ISP: China Telecom (AS23724)	0.69%		0.94%		0.23%	0.15%
ISP: China Telecom Chengdu	7.56%	2.53%	15.09%	0.92%	1.54%	1.88%
Hosting: China Ucloud Beijing				1.45%	1.44%	1.20%
Hosting: China Ucloud Shangai	9.97%	4.30%	16.98%	0.27%	0.20%	1.37%
Hosting: Ehost			0.94%			0.01%
F-Secure	9.97%					0.42%
Hosting: Google Cloud	9.62%	0.38%				0.45%
Kudelski Security		2.65%			3.14%	1.71%
NetSystemsResearch		24.40%		33.49%	29.15%	28.57%
Onyphe		2.91%				0.34%
ISP: Rajfa Vesolak	1.03%					0.04%
Rapid7					3.79%	1.69%
Shodan	29.90%	10.37%	30.19%	17.15%	12.53%	15.05%
Hosting: Softlayer		1.26%		1.61%	1.73%	1.53%
Stretchoid	2.41%	1.01%				0.22%
ISP: Vietnam CHT	0.34%	0.25%	0.94%	0.23%	0.23%	0.25%
Hosting: World Hosting Farm	1.03%	0.51%	0.94%	0.08%		0.15%
Others (<.0.5% for all protocols)	1.37%	1.64%		0.23%	0.49%	0.55%

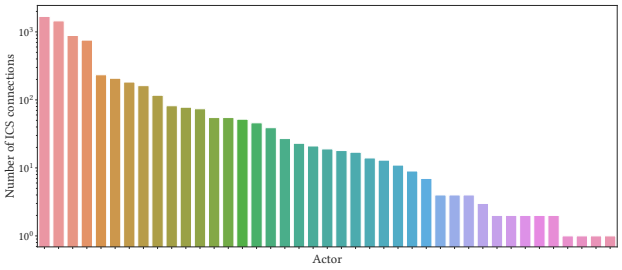

Figure 4: Number of connections per actor (log scale). A handful of actors generates the majority of the ICS traffic.

two distinct scanning campaigns from March 19th to March 25th, and from April 1st to April 8th.

Particularly interesting is the traffic we receive from the Blackhost actor (a bulletproof hosting service): we see only low intensity port scans (with no ICS traffic) targeting all hosts until April 8th, after which the scans greatly increase in volume and include ICS protocols; peculiarly, these more recent scans only target the honeypot instances we deployed on cloud. Figure 5 depicts the volume of ICS-related and general traffic generated by the Blackhost actor, showing this phenomenon.

Ephemeral IP addresses. Recurrent scanners usually have a large number of IP addresses at their disposal to make scans. We noticed two main ways which actors behave when launching a scan:

- "Persistent" IP addresses: the actor has a fixed pool of IP addresses, and these IPs are reused regularly. This is the case for Shodan, for example.
- "Ephemeral" IP addresses: the actor allocates a unique IP address for every single scan. This is possible if the actor's address pool is large enough, or more commonly by using cloud hosting services.

It is worth noting that ephemeral IP addresses are harder to attribute to a specific actor, since they are usually only seen once. In these cases, it is sometimes still possible to group IP addresses together, for instance by looking at multiple IP addresses being picked consistently from the same subnet. Actors that we notice

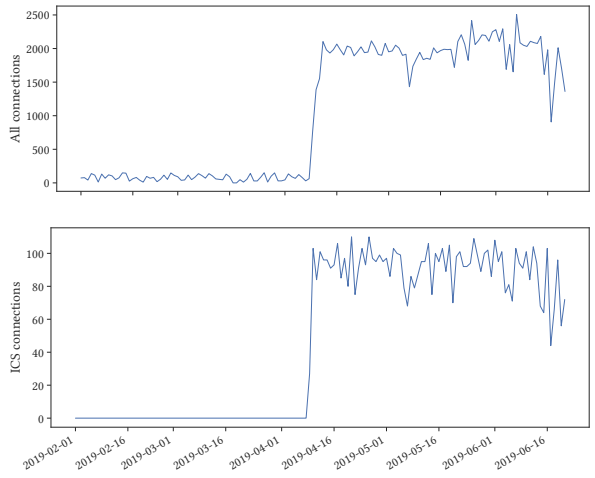

Figure 5: Number of connections received from Blackhost over time. Traffic increases and ICS requests appear at the beginning of April.

using ephemeral addresses are Binaryedge (which we can identify using DNS records) and China Telecom (AS4134).

Scans directed to open ports only. We recorded on multiple occasions new IP addresses making well-formed requests to ports that are open on the specific honeypot instance, without doing any previous port scan. By cross-checking on different instances, we found that these IP addresses do targeted well-formed scans on different hosts. These types of scans are tailored to open ports, using the correct protocol for the port. The actors doing scans in this way are presumably trying to avoid potential scan filters, if for example the target has a system to detect SYN scans and blacklist the IP addresses doing the first scan sweep. There are two ways for an actor to accomplish this: either the actor has access to multiple IP addresses, and uses one IP to find open ports on the target and a different one to do the application-layer well-formed scans, or the actor is using a search engine like Shodan or Censys to gather information about open ports on the target hosts. We noticed this behavior from IP addresses assigned to China Telecom (AS4134) and China Hangzhou Alibaba (AS37963).

Differences between targeted instances. Some actors do not scan the whole IPv4 range equally. Specifically, we notice that Blackhost does scans for the BACnet, EtherNet/IP and Modbus protocols on the honeypot instances that were exposed on cloud IP addresses, but avoids the instances exposed on the honeypots in the network belonging to the research organization; we notice that the actor does port scans for other ports on all instances: the only difference is on the ICS-specific ports. On the other hand, we notice that the Vietnam CHT actor does the opposite: the actor performs ICS-specific scans on the instances belonging to the research organization, while ignoring the cloud instances.

Although we exposed honeypot instances simulating different device models for each protocol, we did not find any significant difference in how actors target them.

4.4 Types of Requests

The set of application-layer request types that we have received is very limited. All the requests (except for the interactions detailed in Section 4.5) are described in Table 5. Notably, for Modbus, we received several requests for "Unity," a request specific to Schneider Electric devices, usually sent to the honeypot instances that advertise themselves as Schneider Electric PLCs. Unity is an extension for the Modbus protocol supporting many different functions, but the only one we have seen is for requesting additional information about the device model. Among the other Modbus requests, "Report Slave Id" requires that a valid slave id is passed as parameter, with value ranging from 0 to 255: Some scanners (e.g., Shodan) send a request for each possible value to collect as much information as possible. Almost all of the EtherNet/IP requests we have seen are "List Identity", which asks for a small set of information about the device; a single actor, Alphastrike, is more accurate to the EtherNet/IP specification by additionally sending a "Request Session" request to obtain a session token before the "List Identity" request. For BACNet, the request "Read Property" allows to request a specific property among object-identifier, vendor-identifier, vendor-name, firmware-revision, application-software-version, object-name, model-name, description, location, system-status. "Read Property Multiple" is a batched version of the request (used with the same properties): We noticed that, usually, if this request fails, scanners fall back to using multiple single "Read property" requests.

All the ICS requests listed in this section are identical or very similar to the standard Nmap [10] scanning scripts (NSE) for the respective protocols, specifically the scripts named "s7-info", "modbus-discover", "iec-identify', "enip-info", "bacnet-info". We suppose the actors are doing automated scans using these scripts, sometimes with small tweaks (e.g., requesting more properties in the BACnet scan). All of these requests are harmless: their only purpose is to ask the target device for information, like the device model or the name of the facility. The request for IEC-104 does a reading of the sensors, but is otherwise harmless too. We never see any harmful or destructive request. This is understandable, as issuing potentially destructive requests does not benefit a generic malicious actor. Unlike SSH or HTTP, the ICS protocols do not offer a way to gain resources from the device, like computing power or bandwidth. Destructive requests with a benefit for an attacker are far more targeted: for instance, a state actor targeting critical infrastructure like a power grid could try to damage it by targeting exposed devices. If this happened, it would be very unlikely for a low interaction honeypot to record the event, both because it is likely to be an extremely rare occasion, and because it would be an attack targeted for very specific, well-vetted devices. There is a possibility that some actors are doing scans to find feasible targets for later attacks, but we were not able to confirm it reliably.

4.5 Interesting Requests

In only a few occasions, we received ICS requests that are not similar to the ones made by standard scanning scripts. The requests are not destructive, but instead of looking for information about the device (e.g., device name), they query the internal state of the device. These interactions show that not all ICS requests received in the wild are from common scanners: actors capable of creating specific and new

Table 5: Types of request received by the honeypot

Name	Requests	Description
Siemens S7		
Setup Communication	40.4%	Starts a new connection
Read SZL / Module Id.	34.8%	Basic module information
Read SZL / Component Id.	23.3%	Component information
Read SZL / Read All	1.9%	All available system information
Modbus		
Read Device Identification	51.5%	Requests vendor and model name, revision number
Report Slave Id	48.0%	Type, state, identification of one of the devices connected to the PLC
Unity	0.5%	Schneider Electric-specific request
IEC-104		
TESTFR	41.4%	Checks if the host is active
STARTDT	32.4%	Enables data transfer
C_IC_NA_1	26.1%	General Interrogation Command (returns the current sensor readings)
EtherNet/IP		
Request Session	0.4%	Requests a session token (optional)
List Identity	99.6%	Basic information, e.g., vendor ID, device type, model, serial number
BACNet		
Read Property	96.0%	Asks for a specific device property, e.g, device name, model name, location
Read Property Multiple	4.0%	Batched version of Read Property

custom scripts exist. Moreover, we see these interactions for a very limited amount of time, suggesting that the actors were focusing on finding specific targets, instead of running wide-range persistent scans. Since in all cases (except one) the actors tried to find out the status of the device, we can speculate that the requests could have been a reconnaissance step to find a suitable target, potentially followed by an actual malicious attack. Unfortunately, being low-interaction, the honeypot was not completely accurate, especially for uncommon requests like this ones: We cannot conclude anything about what the next steps of the attacker would have been if the honeypot had answered in a completely faithful way.

Modbus - Read Holding Registers. We recorded four Modbus interactions issuing a series of "Read Holding Registers" requests is made—a class of request that is not usually included in the scans we have seen. The requests were targeted to the honeypot instances with a working Modbus implementation. Each honeypot was targeted from a different source IP address, all of them assigned to a specific AWS region; grouping them together was feasible since the scan behavior is really similar. Each scan had the same behavior: ten "Read Holding Registers" requests, each one targeted to a different slave unit (from 0 to 9). The request lets a user read the PLC internal registers, i.e., find its current state.

The honeypot responded to the requests with an "Illegal data address" Modbus exception. Since the scan was targeted to the honeypot instances that had a working Modbus implementation, it is reasonable to assume that the actor had access to information about potential targets, for example with tools like Shodan.

EtherNet/IP - Identity / Get Attributes All. On two occasions, we received a EtherNet/IP interaction from an IP assigned to a hosting service (M247 LTD). The interaction included 16 consecutive CIP (Common Industrial Protocol) “Identity / Get Attributes All” requests, one for each address from 1 to 16. The request is functionally equivalent to the more common “List Identity,” requesting standard device information like device name, serial number, status, which the honeypot returned as normal.

S7 - Read Var. We received a single Siemens S7 request using the “Read Var” function, a type of request that is not included in standard scans. The source IP address was a public Tor exit node, indicating that the sender was purposefully trying to stay anonymous. The intended functionality of the “Read Var” request is to read from the memory of the PLC at a specified address; the request we received was meant to read one byte at address 0x0.

S7 - Read and Write Var. We received a single interaction using the “Write Var” function, with a source IP assigned to China Telecom (AS4134). The interaction was composed of three steps: first, three “read var” requests (read 2 bytes at address M100, 1 bit at address M100, 1 byte at address M100); then a “write var” request (write 0 at address M100, bit 1); then, the same three “read var” requests to read 2 bytes, 1 bit and 1 byte at address M100. In general, a write var request may be dangerous, as it modifies the state of the device; due to the simplicity of the request itself, we suppose that the request purpose was reconnaissance, i.e., performing fingerprinting or honeypot detection by checking whether the device was working correctly. Unfortunately, when this request arrived, our honeypot did not implement the "write var" functionality correctly, hence we do not know what the next steps of the interaction would

have been; after receiving this request, we implemented the write var functionality following the specification, but we did not record any other subsequent request using it.

S7 - SZL 0x232 (Communication Status Data). We received three consecutive requests for the field 0x232 in the System Status List (SZL) from a IP address assigned to China Unicom. The request for the field 0x232 at index 4 returns: CPU protection level, operator control settings and version ID/checksums. Our honeypot answered with protection level 0, meaning "no protection" (i.e., hardware configuration and blocks can be read and modified by anyone). The interaction is very interesting, as it actually shows an actor interested in knowing the security state of the ICS device. Nevertheless, the interaction did not continue; we assume the actor was only performing reconnaissance.

4.6 Classification of Scanning Scripts

By analyzing the ICS requests, we notice that most are very similar, but differ only for some specific parameters. These parameters, like the EtherNet/IP "context" field, do not change the semantics of the requests and are usually ignored by the ICS devices. We find that requests made by the same actor have a consistent behavior: if a parameter has a fixed value, this never changes across requests. Furthermore, by studying popular scanning tools (e.g., Nmap), we find out that these parameters are usually fixed in the scripts themselves, and that unique values for the parameters indicate that the actor is using a different version of the scripts.

Since usually these parameters are consistent for a specific actor, we can make use of these small differences between requests as a further tool to identify distinct actors. We group IP addresses by AS if no reverse DNS record is available, but this is imprecise if an actor uses IP addresses from multiple ASs. Assuming that each actor uses only one scanning tool, we can group different ASs if the requests that they make have the same characteristics.

We focus on a small set of fingerprintable parameters, detailed in Table 6: the transaction ID for Modbus, QOI (Qualifier of Interrogation) for IEC-104, and context for EtherNet/IP, and we group the actors as detailed below. Interestingly, we find that no Chinese actor has the same parameters for all the protocols; we have therefore reason to claim that they are all actually distinct actors. Furthermore, each actor we identified never uses more than one version of each type of scan, suggesting that each actor is atomic and not divisible in further distinct actors. We also confirm that scanning traffic received from IP addresses that share the same DNS PTR records is consistent: there are no cases among the actors we identified with this method where the scans use different parameters, showing that DNS PTR records are a good criteria for distinguishing actors. Curiously, Shodan and Vietnam CHT have scripts that use the same parameters, but the Vietnam CHT actor does not have any reverse DNS record linking it to Shodan, which are usually always set. The Vietnam CHT actor is a very active recurrent scanner: it may very well be that it corresponds to misconfigured Shodan hosts.

5 RELATED WORK

Security of ICSs and, more in general, of cyber-physical systems, is a widely studied research area, and issues in this field have been discussed extensively [6, 7]. The most common ICS network protocols were originally designed to work in closed and secured networks: They have few, if any, security mechanisms, and have been shown to be easy to exploit to the point that an accessible open port can provide full unauthenticated control to the ICS device [11, 17]. On the field of defense, efforts concentrated on offering standardized security guidelines for ICS and SCADA systems [8, 18], suggesting network segmentation and firewalls. Despite this, recently many ICS devices have been exposed on the Internet with little to no protection. Several pieces of research tried to estimate the amount of exposed ICS devices with full-range scans [3, 12], finding that more than 100,000 ICS devices are exposed on the Internet, which are prone to attacks. This inevitably attracts actors interested in finding vulnerable devices and potential targets.

The goal of this paper is to study any traffic made by actors interested in finding vulnerable ICS devices. Several approaches have been used in literature to estimate scanning activity. One of the most widespread tools are network telescopes (i.e., "darknets"), large ranges of IP addresses which are supposed to be unused and do not contain legitimate traffic. As they are a useful tool to study unsolicited scans made indiscriminately to all IP addresses, they have been used before to measure port scans and botnet activity [13], also in the ICS field [12]. The drawback of network telescopes is that they are passive, thus they cannot be used to study full interactions with scanners, and are therefore unable to give much information about the application layer. A different approach is to analyze real traffic passing through a central internet vantage point; Nawrocki et al. [14] use this approach in the context of ICS.

A different way to study scanning activity is by means of honeypots. Indeed, honeypots are often used to find how particular classes of devices and protocols are targeted in various contexts, ranging from the deployment of a worldwide distributed network of honeypots [9], to studying threats against Internet of Things devices [20]. In the ICS context, Mirian et al. [12] deploy a default configuration of conpot (thus easy to fingerprint and detect), as part of their study using data from a network telescope. Serbanescu et al. [15, 16] use low interaction conpot honeypots to measure scans, focusing on Shodan scans and how they influence other scans by publishing exposed hosts. To overcome the drawbacks of low interaction honeypots, Antonioli et al. [1, 2] propose a more complex architecture, using honeynets and simulating entire networks, including the physical process; they only discuss the design and implementation of the system, without discussing the results of an in the wild deployment.

Honeypots are also widely used to study threats affecting IoT systems. In this spere, Vervier et al. [20] deploy a combination of high- and low-interaction honeypots to perform a comprehensive study of unsolicited requests targeted to IoT systems, including malware targeting IoT devices; Tambe et al [19] define a pipeline including deployment, enrichment, analysis, using high-interaction honeypots, and propose using public VPNs to obtain new non-suspicious network vantage points; Guarnizo et al. [5] propose "wormholes" to redirect traffic from remote network vantage points to locally hosted high-interaction honeypots. We use a similar approach, using a VPN to forward all traffic at the network layer.

Table 6: Protocol fields used by various actors.

Value	Actors
Modbus: transaction ID identification number for a Modbus request; usually ignored	
`0` (used by Nmap)	ABCDE Group, AS Data, Binaryedge, Capitalonline, China Telecom (AS4134), China Hangzhou Alibaba, China Sichuan, DCS Pacific Star, Dahai Network, Kudelski Security, Shodan, Tamatiya, Vietnam CHT
`1`	China Telecom Chengdu
`4919` (i.e., `0x1337`)	Alphastrike, Blackhost, Censys, NetSystemsReseatch, Onyphe, Softlayer, Stretchoid, Vultr
`23111`	World Hosting Farm
random value	China Ucloud Shangai
IEC-104: QOI used in the IEC-104 General Interrogation Command to further specify the type of sensors to read from; usually ignored.	
`0`	Capitalonline, Shodan, Vietnam CHT, World Hosting Farm
`20` (used by Nmap)	Baidu Netcom, China Hangzhou Alibaba, China Sichuan, China Ucloud Shangai, EHOSTIDC
EtherNet/IP: context works as an identification number for the EtherNet/IP request	
`0`	Alphastrike, Blackhost, NetSystemsResearch, Softlayer
`0xc1debed1` (used by Nmap)	Baidu Netcom, Binaryedge, China Telecom (AS4134), China Sichuan, China Ucloud Shangai, China Wenzhou, Velia, World Hosting Farm
`0x6a0ebe64`	Shodan, Vietnam CHT

6 CONCLUSIONS

In this paper, we proposed a scalable low-interaction honeypot architecture, augmented with an automated extensible analysis pipeline to study the ICS-targeting traffic, we deployed a set of honeypots in various configurations (protocols and networks) resembling real ICS devices, and we analyzed the results of running our honeypots for several months, to understand what types of requests are made in-the-wild and who are the actors 'doing scans'. First, we found that all the (non-targeted) requests made in ICS-specific scans are requests for information on the device, or sensor readings. As a matter of fact, we found that, except a few exceptions, the requests matched standard scanning scripts like Nmap. Second, as opposed to common IT protocols, where most of the traffic is generated by botnets, we found that ICS traffic is dominated by a few recurrent scanners. We found that the number of distinct actors is very limited (less than 100) and a majority of them are known benign public scanners. Excluding these, we find that most of the actors are hard to identify, as the IP addresses are assigned to large ISPs or cloud hosting services. Furthermore, we showed how requests fields can be used to further distinguish actors. Finally, we show some interactions that we have seen making unique requests, suggesting that although most actors targeting ICS devices are innocuous scanners that are reusing scripts, actors capable of creating specific and new custom scripts exist.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 690972. The paper reflects only the authors' view and the Agency and the Commission are not responsible for any use that may be made of the information it contains.

REFERENCES

[1] Daniele Antonioli, Anand Agrawal, and Nils Ole Tippenhauer. 2016. Towards high-interaction virtual ICS honeypots-in-a-box. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*. ACM, 13–22.

[2] Daniele Antonioli and Nils Ole Tippenhauer. 2015. MiniCPS: A toolkit for security research on CPS networks. In *Proceedings of the First ACM workshop on cyber-physical systems-security and/or privacy*. ACM, 91–100.

[3] Xuan Feng, Qiang Li, Haining Wang, and Limin Sun. 2016. Characterizing industrial control system devices on the internet. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. IEEE, 1–10.

[4] Dor Green. 2019. Pyshark. https://github.com/KimiNewt/pyshark.

[5] Juan David Guarnizo, Amit Tambe, Suman Sankar Bhunia, Martín Ochoa, Nils Ole Tippenhauer, Asaf Shabtai, and Yuval Elovici. 2017. Siphon: Towards scalable high-interaction physical honeypots. In *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*. ACM, 57–68.

[6] Vinay M Igure, Sean A Laughter, and Ronald D Williams. 2006. Security issues in SCADA networks. *computers & security* 25, 7 (2006), 498–506.

[7] Johannes Klick, Stephan Lau, Daniel Marzin, Jan-Ole Malchow, and Volker Roth. 2015. Internet-facing PLCs-a new back orifice. *Blackhat USA* (2015), 22–26.

[8] William Knowles et al. 2015. Assurance techniques for industrial control systems (ics). In *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*. ACM, 101–112.

[9] Corrado Leita et al. 2008. The leurre. com project: collecting internet threats information using a worldwide distributed honeynet. In *2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing*. IEEE, 40–57.

[10] Gordon Lyon. 2019. Nmap security scanner. https://nmap.org/.

[11] Brian Meixell and Eric Forner. 2013. Out of control: Demonstrating scada exploitation. *Black Hat* (2013), 2013.

[12] Ariana Mirian et al. 2016. An internet-wide view of ICS devices. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 96–103.

[13] David Moore, Colleen Shannon, Geoffrey Voelker, Stefan Savage, et al. 2004. *Network telescopes: Technical report*. Technical Report. Cooperative Association for Internet Data Analysis (CAIDA).

[14] Marcin Nawrocki, Thomas C Schmidt, and Matthias Wählisch. 2019. Uncovering Vulnerable Industrial Control Systems from the Internet Core. *arXiv preprint arXiv:1901.04411* (2019).

[15] Alexandru Vlad Serbanescu, Sebastian Obermeier, and Der-Yeuan Yu. 2015. ICS threat analysis using a large-scale honeynet. In *Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research*. BCS Learning & Development Ltd., 20–30.

[16] Alexandru Vlad Serbanescu, Sebastian Obermeier, and Der-Yeuan Yu. 2015. A scalable honeynet architecture for industrial control systems. In *International Conference on E-business and Telecommunications*. Springer, 179–200.

[17] A Soullie. 2014. Industrial control systems: Pentesting PLCs 101. *BlackHat Europe* (2014).

[18] Keith Stouffer, Joe Falco, and Karen Scarfone. 2011. Guide to industrial control systems (ICS) security. *NIST special publication* 800, 82 (2011), 16–16.

[19] Amit Tambe, Yan Lin Aung, Ragav Sridharan, Martin Ochoa, Nils Ole Tippenhauer, Asaf Shabtai, and Yuval Elovici. 2019. Detection of Threats to IoT Devices using Scalable VPN-forwarded Honeypots. (2019).

[20] Pierre-Antoine Vervier and Yun Shen. 2018. Before Toasters Rise Up: A View into the Emerging IoT Threat Landscape. In *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 556–576.