Measuring What Matters: Revisiting Internet Exposure of OT Networks

Ricardo Yaben, Mathias Anguita, Emmanouil Vasilomanolakis

[a] *Technical University of Denmark, Kongens Lyngby, Denmark*

Abstract

Many Internet-wide measurement studies report thousands of vulnerable Operational Technology (OT) devices exposed to the Internet. This growing focus on OT is essential for informing new regulations, mitigation techniques, and defense strategies, but many studies overlook false positives in their datasets. Ignoring artifacts such as honeypots, network telescopes, and tarpits leads to misinterpretations and a distorted view of the Internet. This paper revisits the exposure of OT networks to the public Internet. We apply a noise-aware methodology to an Internet-wide scan of networks that expose Modbus, Fox, EtherNet/IP, and IEC 60870-5-104 services, and we filter likely noise from vulnerable devices. Our findings show that noise systematically pollutes datasets and inflates estimates of exposed OT services; across protocols, 7% of the total observations, and up to 20% for particular protocols can be traced reliably to four sources of noise that we term *condensation, displacement, volatility*, and *hostility* using conservative policies with high-confidence signaling classifiers. That said, even after filtering these artifacts, the security landscape of Internet-facing OT devices remains largely unchanged. Devices are still widely affected by misconfigurations, obsolescence, and broader security management issues.

Keywords: vulnerability identification, Internet-wide scans, OT, ICS, Internet measurements, deception, honeypot

1. Introduction

Previous studies on the current state of security in OT networks exposed to the public Internet consistently report the gravity of the situation and the risks of placing critical systems online. Many works demonstrate that commonly used OT protocols were never designed to operate over the Internet and lack basic security properties such as access control, encryption, and protection against data leakage [43, 21, 9]. Other studies highlight widespread security management issues (for example, unrevoked certificates, use of certificates past their end-of-life dates, and deprecated firmware) [60], and a large part of the community focuses on discovering vulnerabilities [54, 56] and characterizing attack trends [15]. The urgency of this topic is justified, as OT systems are both essential and increasingly attractive targets. Incidents such as Stuxnet [27, 31], BlackEnergy [28], Industroyer [30], Havex [36], and LockerGoga [2] have demonstrated the devastating effects of cyber-attacks on critical infrastructure.

However, much of this research has likely substantially over-reported its measurements, overestimating the number of exposed and vulnerable OT systems [39]. According to Srinivasa et al. [53] and Mladenov et al. [39], previous studies may have failed to handle *noise* efficiently (for example, deception systems and network telescopes), and they estimate that as many as one in five observations are either honeypots or systems with highly unlikely characteristics, such as offering more than 30 services. Although these findings do not diminish the importance of monitoring OT networks, they raise new questions about how to measure what matters.

In this work, we revisit the exposure of OT networks on the public Internet and apply a method to detect *noise* based on insights from the literature (for example, fingerprints, suspicion indicators, and unlikely characteristics) and our own criteria (for example, network density, distinct responses to identical probes, and pacifying attempts). Our noise-identification method primarily targets four sources of noise that we describe as *condensation*, *displacement*, *volatility*, and *hostility*. We discuss our definition of each source in the context of the literature, explain what these sources capture, describe how we identify them, and analyze the limitations of our method.

We then use four protocols that are heavily deployed in OT networks as case studies to evaluate our method, and we compare observations before and after applying our noise-identification approach to demonstrate how noise affects results. We choose Modbus and Fox as the two most commonly reported protocols to facilitate comparisons with previous studies, and we include EtherNet Industrial Protocol (EtherNet/IP) and IEC 60870-5-104 (IEC 104) as two additional widely used protocols that are underrepresented or mischaracterized in the literature. In

Email addresses: rmyl@dtu.dk (Ricardo Yaben), s194575@student.dtu.dk (Mathias Anguita), emmva@dtu.dk (Emmanouil Vasilomanolakis)

addition, we describe our experimental setup, probe details, and the manipulations needed to test for particular sources of noise, such as tarpits and hostile hosts. While some criteria in our method work best with OT protocols, the overall noise-identification approach can be applied to other protocols. This paper concludes with a discussion of the limitations of our method and future directions; we intend this method to serve as a baseline for more accurate measurements and for the design and evaluation of deception systems.

Our contributions are as follows:

- We develop a noise detection methodology that extends the current literature to identify four sources of noise, which we term *condensation*, *displacement*, *volatility*, and *hostility*. To the best of our knowledge, our method is the first to explicitly test for network density properties and volatile hosts that may result from deception techniques such as Moving Target Defense (MTD).

- We apply this methodology to four widely used protocols in OT networks: Modbus, Fox, IEC 104, and EtherNet/IP. We conduct an Internet-wide scan of these protocols using stock ZGrab2 probes for Modbus and Fox, and develop and share new probes for IEC 104 and EtherNet/IP. Our analysis indicates that although we are able to annotate 7% of the total observations exhibiting strong noise indicators and filter those observations from suspected vulnerable devices, the overall OT security landscape remains unchanged. The code and configuration used to collect our dataset are publicly available[1]. The dataset hosted on Zenodo with restricted access; the full dataset is available upon request due to ethical considerations related to OT protocols [64].

The rest of this paper is structured as follows. Section 2 describes our data collection methodology and the ethical considerations we took into account during our experimentation. Section 3 introduces the concept of *noise* and provides background for our classification of sources, drawing on the literature on deception techniques and common issues in active Internet surveys. Section 4 analyzes the results of our four case studies: Modbus, Fox, IEC 104, and EtherNet/IP. Section 5 discusses the impact of noise on OT Internet measurements, along with limitations of our study and directions for future work. Section 6 presents related work on active Internet measurements of OT networks and the research gaps that motivate this study. Section 7 concludes the paper.

[1]Data collection code available at `https://github.com/RicYaben/dice-publications/tree/main/computers-and-security-2025`

2. Methods

The measurement campaign took place in October 2025 and lasted 24 hours without interruption. We expect the availability of vulnerable OT devices facing the Internet to vary only modestly over such time scales; consequently, while our results represent a snapshot from a single day, we do not anticipate that the timing of our scans qualitatively changes the patterns we report. This remains an assumption and therefore a threat to validity, which future longitudinal measurement campaigns could examine explicitly. Our data collection and noise-identification methods are, however, readily applicable to such longitudinal studies and to deployments that monitor devices over longer periods, where anomalies may become more apparent over time.

Conducting repeated scans on the same targets can help detect behavioral changes over time. Repeating this procedure may increase the accuracy of our volatility identification method, although with diminishing improvements. Without a more sophisticated approach, these repetitions increase the traffic towards these hosts linearly. For example, further studies could set a maximum boundary of repetitions and filter hosts between iterations that do not seem to change behavior. This study, however, conducts a single subsequent scan as a proof of concept to test this theory.

2.1. Data collection

Our data collection method consists of an active Internet-wide scan using a traditional two-step approach combining ZMap [16] and ZGrab [66], supplemented with crowd-sourced data from RIPE Atlas [48], and Cyber Threat Intelligence (CTI) and IP reputation services GreyNoise [22] and AbuseIPDB [1]. Using exclusively CTI services (e.g., Shodan [35] or Censys [15]) or passively collected traffic would limit our ability to identify sources of noise, which require crafting and manipulating probes. On the other hand, our Internet-wide scans do not provide enough insights to either characterize noise or profile hosts, requiring prefix and Autonomous System (AS) data.

Vantage point. Our experiments were conducted from a single vantage point located in our institution. This vantage point has been used multiple times for similar experiments, which may affect our overall results due to a multitude of blocking strategies and the elevated number of appearances in IP reputation services (e.g., AbuseIPDB, or GreyNoise). While we have not observed significant differences across our experiments, we acknowledge that the vantage point's location and repeated use play a role in our capacity to observe the Internet [58, 15].

Scanners. We use ZMap and ZGrab2 to conduct Internet-wide L4 and L7 sweeps [16, 66], applying a blocklist to avoid scanning certain prefixes. Our blocklist

merges IP ranges from the public Censys repository with prefixes that opted out of our studies. In total, it excludes approximately 20% of the routable IPv4 space; we therefore do not publish it to avoid undermining opt-out protections and enabling misuse. Maintaining high-quality blocklists is a complex task and we encourage further community work in this area. Our measurement follows a two-step workflow. First, we use ZMap to send TCP SYN probes to the default TCP ports of Modbus, Fox, IEC 104, and EtherNet/IP, and treat hosts that respond with SYN/ACK as L4-positive. Second, we perform a stateful L7 scan using ZGrab2 with protocol-specific probes: we use the stock ZGrab2 probes for Modbus and Fox, and we developed new ZGrab2 probes for IEC 104 and EtherNet/IP. To test for volatility, we repeat this scan iteration within the same 24-hour campaign for the set of L4-positive addresses and compare the results across iterations.

Probes. We use a mix of existing (stock) probes and custom probes: Modbus and Fox use existing ZGrab2 modules, while IEC 104 and EtherNet/IP are implemented by us and released with the artifact. Our ZGrab2 probes are designed to test how much information unauthorized users could gain from services without exploiting vulnerabilities. Except for newer standards, the protocols under study do not provide any security features, lacking authentication, access control, and encryption. Our probes do not modify the state of the target host, using exclusively requests to pull information regarding the device's state; as seen throughout this paper, this is the minimal form of interaction sufficient to identify devices and evaluate their security. However, our probes are more intrusive than bare banner-grabbing, falling closer into the category of resource enumeration. Furthermore, we prepared our probes to handle hostility. First, probes are limited to 10 seconds for connecting to remote hosts, and an additional 10 seconds to gather information. These measures prevent issues such as interruptions and consuming excessive resources on a single host, which handles tarpits and excessively slow hosts. The approach prioritizes granularity at the cost of increased traffic. See Section 4 for further details on the individual probes.

Crowd-sourcing. In addition to Internet-wide scans, we enhance our findings with data from RIPE Atlas, and GreyNoise. RIPE Atlas supports our dataset with AS prefix information to group addresses by their ranges. This data is necessary to detect *condensation* noise. Lastly, we use GreyNoise for indicators of suspicious activity from OT networks since unsolicited outbound requests from OT assets to GreyNoise sensors are unusual.

2.2. Ethics

As part of the best practices for conducting ethical Internet measurements via active scanning campaigns, the vantage point used to collect observations continues to host an informational website regarding our scanning activities, including identification signatures to distinguish scanning traffic, a summary of the ports and services probed, tools in use, and contact information. Network administrators can opt out of our studies at any time, effectively removing their networks from our current and subsequent measurements. Complaints to our institution and contact information provided through WHOIS records are also forwarded directly to us. In addition, we continuously revise our agreement with our institution's Internet provider and administrator of Denmark's research network to maintain a common understanding of the experiments we conduct.

Furthermore, the scanning tool ZMap already implements multiple measures to mitigate scanning impact on remote networks, randomizing and ensuring maximum distance between the target addresses [16]. Moreover, the probes included in this paper, including stock ZGrab2 modules and our custom probes, were crafted according to the protocol specifications and do not alter the state of the targeted service, limiting requests and commands to discovery functions, and closing connections gracefully after 10 seconds from the first message.

Though OT networks are believed to be fragile, devices facing the Internet are exposed to constant attacks and unsolicited traffic of sizable volume, often not in conformance with their service expectations. Our scanning methodology attempts to reduce the impact on these networks and mitigate their security issues, but we acknowledge our contribution to the increasing problem of excessive and overly frequent scanning activity.

3. Noise

Reporting on results from Internet measurements is often complicated and filled with caveats. The Internet itself is populated with countless devices and networks built to observe, measure, deceive, mislead, trap, or even retaliate against those who interact with them. We refer to the collection of such networks and devices as *noise*, that is, false positives that pollute datasets and create overly optimistic impressions of reality. However, most Internet surveys ignore this aspect entirely, and only a few studies briefly mention some of these challenges as a limiting factor [51]. To address this issue, we developed a novel approach to annotate datasets with labels from four different sources of noise: *displacement, condensation, volatility, and hostility.* This method primarily builds upon previous studies that focus on network properties, behaviors, and deception detection techniques, combining multiple theories from the literature with our own indicators. Note that Feng et al. [18] and Singla et al. [50] already used this terminology to filter honeypots.

Our noise detection method attempts to answer a challenging question with confidence: *How can we distinguish vulnerable OT devices from noise sources?* Among the most common types of noise in OT environments, we find deception systems (e.g., honeypots, tarpits, echo servers,

sinkholes, and telescopes). Recent studies on OT honeypot fingerprinting suggest that deception is no longer a niche technique [39, 10]. The literature provides strong arguments for authors conducting vulnerability identification studies at scale to consider implementing further measures to address interactions with deception systems. In measurements like ours, where the goal is to identify vulnerable OT systems, including Industrial Control System (ICS) and critical infrastructure, this type of noise bloats results with false positives. This not only impacts negatively on the perception of the issue but also adds unjustified pressure on the reported parties. However, current methods to detect this type of noise are mainly based on heuristics (e.g., unrealistic number of open ports and unlikely locations), lacking proper evaluation and validation. Therefore, we expand the current state of the art with features for detecting deception and other sources of noise using empirical indicators at three scope levels: Internet, network, and host levels.

This section describes the concept of *noise* and its ties to deception techniques and other common events in Internet measurements that impact results. We follow a systematic structure, providing a general definition of the term, a background covering the body of work researching similar concepts, the methodology to identify the type of noise, and the results of our implementation. A description of our classifiers per noise source is shown in Table 1, along with brief descriptions and criteria. In addition, the table shows the number of hosts labeled using each classifier, and the types of noise designed to identify.

To guide readers into the different estimations of noise used throughout the paper, and to clarify on the confidence with which those estimations should be treated, we implement policies separating annotated observations into three depth-levels: *conservative*, *balanced*, and *aggressive*. The first policy, *conservative*, represents the collection of hosts with one or more high-confidence noise signal ($H \geqslant 1$). By default, results are discussed using this policy. The second, *balanced*, includes hosts with at least one high-confidence signal, or more than two labels of any confidence ($(H \geqslant 1)|((H + L) \geqslant 2)$). Lastly, *aggressive*, collects hosts with any noise signal, including low-confidence ($(H + L) \geqslant 1$). Table 2 provides a summary of these policies.

3.1. Condensation

Condensation is the term we use to describe large networks with deeply concentrated clusters of hosts. These clusters are closely allocated hosts with similar characteristics (e.g., exposed services). Condensation measures the probability of finding highly dense networks of similar hosts, and is relative to the size of the network to adjust for the variable prefix lengths advertised on the Internet. Therefore, the metrics used to measure condensation are network density, its volume, and the population of hosts with similar characteristics (e.g., exposed services). Detecting condensation attempts to answer the question:

Is the host located in a network with an unlikely number of similar neighbors for its size?

Mladenov et al. [39] also observed large clusters of similar devices concentrated in a single country and two major Internet Service Providers (ISPs), which they deem suspicious behavior but could not verify. Heidemann et al. [24] offered a possible explanation with an often forgotten aspect of the Internet: multiple addresses can be mapped to a single host; this is a relatively common phenomenon, also known as *aliasing*. Other authors previously reported on similar findings, where devices share similar but not identical or unrealistic characteristics [15, 13], often showing these concentrations are more common but not unique to research networks [40]. Despite being considered a known oddity in the literature, the concept has rarely been openly correlated to deception systems or other forms of noise [40] – with few exceptions, such as Knight et al. [29], which leverages artificial diversity in dense populations as a form of bio-inspired deception. In fact, Durumeric et al. [15] were able to identify a group of Human-Machine Interface (HMI) devices in the US as vulnerable devices facing the Internet, highlighting that what we may consider *noise* or suspicious behavior could be an indicator of further issues. On the other hand, Srinivasa et al. [51] suggests a different direction to this concept: using network telescopes as large deception networks; darknets, network telescopes, sinkholes, and other large sensor networks often refer to unused space that should not receive any traffic, i.e., all incoming traffic should be considered suspicious. Besides providing critical value in many Internet measurements, these types of networks have gained relative traction in the field of deception research, as they provide a unique opportunity to understand large and scattered network events, side/cascading effects, background radiation, etc. [34, 51] As Männel et al. [34] mentions, telescopes are not necessarily empty, and other authors have placed sensors and deception systems within their space or adjacent to them on multiple occasions [37, 20, 47, 42, 7]. As a consequence, a possible explanation is that these large clusters with oddly similar devices are, in fact, large deception networks. To further study this behavior, we denominate this type of noise as *condensation*, i.e., abnormally large clusters of similar hosts.

To detect prefixes with high condensation levels, we fitted a simple linear regression model with two variables: the size of the prefix and its density. Prefix density is calculated as the observed population over its size, as seen in Equation (1) where P is the prefix and $\sum H_P$ is the number of responding hosts during our ZMap scan on that prefix.

$$\text{Density}(P) = \frac{\sum H_P}{2^{32-\ell(P)}} \tag{1}$$

Where:

Table 1: Noise source and classifiers for their identification. The table includes confidence levels to threat results produced from each classifier, an overview of the decision rules triggering the classifier, the number of hosts labeled with each, and their limitations. Each classifier indicates its target noise or capabilities of the classifier: **TA** – Tarpit, **TE** – Telescopes, **H** – Honeypots, **MTD** – Moving Target Defence.

Noise	Classifier	Target	Hosts	Decision rule and limitations
Condensation	Dense▼	TA,TE,H	12,519	Significantly more populated prefixes than others of similar size ($> 95\%$). **Limitations:** Highly sensitive
Displacement	Bloated	TE,H	620	Hosting significantly more services than the population ($> 95\%$)
	Aletheia	TE,H	1,257	Fixed TCP window and scaling factor common in cloud networks and Python servers. **Limitations:** Few known values
	Honeypot[1,2]	H	33	Uses known signatures to fingerprint honeypots **Limitations:** Few known fingerprints
	Odd[1]	H	240	Collision indicators: serial numbers (EtherNet/IP) and data frames (IEC 60870-5-104)
Hostility	Tarpit[1]	TA	62	Connection timeout while receiving data (Modbus and IEC 60870-5-104). **Limitations:** Requires reading from the stream until timeout
Volatility	Intermittent▼	MTD	2,252	Services either become available or unavailable after subsequent scans. **Limitations:** Performance improves with repeated experiments
	Morphed▼	MTD	2,578	Services return banners with different static properties after subsequent scans. **Limitations:** Performance improves with repeated experiments

[1] Protocol-dependent.
[2] Conpot and Honeygrove default configuration signatures for Modbus, EtherNet/IP, and IEC 104 services.
▼ Low-confidence signal (L).

Table 2: Noise policy summary over Exposed hosts. **Conservative:** $H \geqslant 1$; **Balanced:** $(H \geqslant 1)|((H + L) \geqslant 2)$; **Aggressive:** $(H + L) \geqslant 1$. Where H = high-confidence signals, and L = low-confidence signals.

Protocol	Policy		
	Conservative	Balanced	Aggressive
Modbus	41	394	2,475
Fox	395	1,720	5,140
IEC 104	219	706	3,475
EtherNet/IP	784	940	3,344
Total Unique Addresses	1,427	3,775	14,396

P = IPv4 prefix
$\ell(P)$ = prefix length of P
H = host
H_P = host belonging to prefix P

Our test shows these variables (or predictors) have the weakest correlation, as the density of the prefix inevitably decreases as the size increases. Other variables show strong dependence on our dataset, and when comparing models without these variables we observed substantial statistical differences. In particular, AS and country location act as confounding variables in our dataset, since models using these predictors would misclassify prefixes from whole countries or AS. On the other hand, our dataset is highly imbalanced towards smaller prefixes, most of which are in ranges between /16 and /26, with /20 being the most common prefix size, accounting for 15,885 of the total, and heavily skewed towards the smaller prefixes. This means that while the density of the prefix drops exponentially as the prefix size increases, the probability of condensation scales linearly. As an example, condensed /24 prefixes require at least 165 hosts and densities of 0.6, while /20 prefixes require 211 hosts at 0.05 density values, and /19 prefixes require 224 hosts at density values of 0.02. Therefore, this model indirectly imposes a scale penalty that removes smaller prefixes than /24, requiring capacities larger than 100 hosts. Figure 1 illustrates this relationship, and how, despite fitting hosts exposing different protocols, the likelihood of finding OT devices in certain prefixes does not change. The distribution of hosts across prefixes concentrates around prefixes of small to medium sizes, independently of the exposed services. This means that the condensation probability does not depend on the protocols we covered.

This model serves as a baseline to estimate whether some prefixes are denser than others of similar size, helping identify network telescopes, honeynets, and sinkholes, among others. We assume that most OT networks are not exposed to the Internet, and observing them in the wild is rare. Therefore, observing networks overly populated with

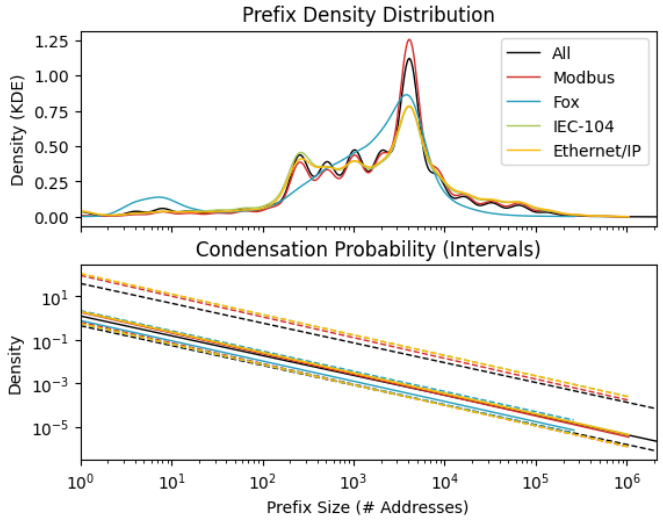


Figure 1: Prefix density distribution (top), and condensation probability ranges for prefixes of each size (bottom). Probability ranges are given between the dashed lines representing the confidence interval between percentiles at the extremes (95% and 5%). Condensed labels are assigned to prefixes with hosts crossing the $p > 95\%$ threshold for their size.

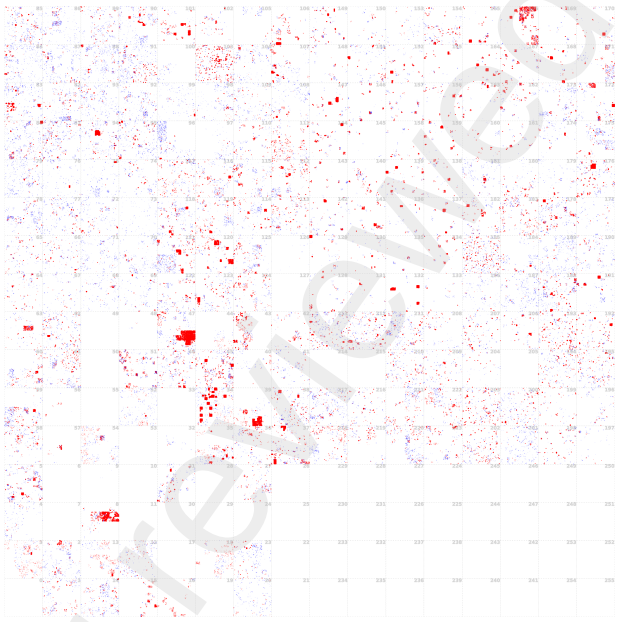


Figure 2: Dense prefix classification over hosts responding to OT probes. The figure shows large clusters of host addresses classified and colored in red, while scattered hosts across prefixes are represented in blue. Four large clusters are particularly dense in octets 8, 34, 39, 47, and 166.

OT devices is a considerably strong reason for suspicion. Applying this model to our dataset consisting of 53,597 unique prefixes identifies 12,463 dense prefixes comprising 3.3 million hosts; only 3% of these hosts are confirmed to run real services under our identification process, yet they account for 65% of the initially assumed exposed OT devices. Figure 2 shows the results from our modeling in a map of the Internet as we observed it using a Hilbert Curve to represent host addresses (classified hosts appear in red). See Section 8 in the Appendix for individual distribution maps.

In terms of limitations, this classification alone does not provide enough evidence to justify disregarding hosts without the presence of other indicators. Instead, it is recommended to be used in combination with others for validation as a sensible indicator of noise. In addition, it may not perform as expected for sparse distributions where there are many large clusters, or their contextual properties make the highly dense clusters the expectation (e.g., large concentrations of websites hosted in cloud providers). Considering additional host properties may produce better results at the cost of complexity.

3.2. Displacement

Besides large clusters of similar hosts, we also consider unlikely observations. Hosts showing unlikely characteristics, either contextual or inherent, we call *displaced*. Contextually unlikely characteristics are those that deviate from their community. One common example in this paper is finding Programmable Logic Controller (PLC) in cloud networks. Moreover, inherent characteristics are those that would make hosts and services extremely unrealistic, such as two devices using the same serial number. Displacement asks:

> *Does the host show unusual characteristics for its context?*

Among the most recent studies, Mladenov et al. [39] shows that exposed ICS are plagued with deception systems using a mixed implementation of heuristics and known signatures to fingerprint ICS honeypots. In addition, the authors use contextual features, including the type of network where the host is located, and inherent properties, such as exposing a large number of ports. Their analysis shows that approximately 90% of ICS expose fewer than 10 ports, with rare exceptions going beyond 30. In addition, their results demonstrate a significant difference in the number of ports depending on the type of network, showing a strong correlation between certain features. Notably, Srinivasa et al. [53] established a similar approach, augmenting host information with network meta-data and correlating their results with other datasets. Their methods include FQDN characteristics to identify domains with self-revealing keywords (e.g., "honeypot" or "test"). These characteristics are mainly anomalies, often unrealistic or plainly obvious. For our own characterization, we summarize these anomalies as a source of noise that we refer to as *displacement*, i.e., properties that make the host highly unlikely within their context.

To identify contextual anomalies, we enhance objects with AS meta-data. This type of information has proven

Table 3: Protocol pair co-exposure rates. Pairwise co-exposure ranges from 0 to 0.0173, indicating low co-occurrence across protocols in our vulnerable set.

	EtherNet/IP	**Fox**	**IEC-104**	**Modbus**
EtherNet/IP	1	0.0003	0.002	0.0173
Fox	0.0003	1	0	0.0003
IEC-104	0.002	0	1	0.0008
Modbus	0.0173	0.0003	0.0008	1

extremely useful in detecting fleet-wide anomalies, as a mitigation mechanism to identify compromised systems, and contingency measures to combat Denial of Service (DoS) attacks. Other authors use this terminology under different names, such as *behavioral features*, or in a different context, e.g., *network anomaly detection* for intrusion detection systems. For our use case, we adapt similar principles to anomalous observations in active measurements. This approach benefits from in-demand interaction with objects, allowing us to survey them as needed. However, it misses temporal features only perceived from continuously monitoring objects. Our method relies on four different classifiers to test for displacement: protocol bloating, the Aletheia method [10], honeypot fingerprints [65], and oddity identifiers.

Bloated. First, we classify hosts exposing an abnormal number of open ports. From our measurements, we observed that positive responses from OT protocols have an almost negligible correlation, i.e., most OT devices have low protocol diversity and expose one to two services (in rare occasions), which aligns with the findings of previous studies [39, 53]. Using these correlation weights, we model a predictor to classify hosts exposing too many ports and unlikely combinations. By reversing the process, i.e., fingerprinting devices first and evaluating the ports afterwards, we can properly model when hosts expose an abnormal number of services, and correlate that to the number of open ports. In summary, our results suggest the probability of finding OT devices with more than two services is extremely low. This classifier yielded 620 hosts exposing more than 2 services. The pair-wise correlation percentages between protocols from positive results can be seen in Table 3, showing no relationships at all between devices exposing Fox and IEC 104, or extremely weak otherwise. The lack of interaction between Fox and IEC 104 is expected, since Fox is mainly used in building automation, and IEC 104 is a protocol used in power grid automation – more common in Europe. Modbus and EtherNet/IP can be used in an array of networks with electronic controllers. The table can be read as the percentage of hosts exposing a particular combination.

Although this step may be unnecessary for our study and a steep threshold at two services seems sufficient for such low correlations, others may benefit from it when considering different protocols or larger studies, which simplifies identifying telescopes and honeypots, especially when paired with condensation labels. To further improve the precision of this method (e.g., in studies with strongly correlated protocols), scans could be coupled with semi-random port checks for other known OT protocols (e.g., IEC 104 and ATG, protocols used to manage fuel tanks [39, 65]).

Aletheia. We implement the signatures described by Cordeiro and Vasilomanolakis [10] to identify cloud environments and Python servers. We expect to observe mainly controllers exposing services not suitable for cloud environments, including but not limited to PLCs, HMIs, Remote Terminal Units (RTUs), Industrial PC (IPC)s, and various gateways. Most of these devices use compiled logical programs with bare-minimum functionality to serve the protocols mentioned here. IPCs and HMI can be considered the exception since these electronics tend to be more capable. This classifier labeled a total of 1,257 hosts exposing OT services, all of which seem to be in cloud environments.

Honeypots. We include two honeypot fingerprinting signatures described in [33, 65, 39, 53] for ICS environments with Conpot [19] instances using default configurations. Conpot is considered the principal ICS honeypot deployed in the wild and has been subject of study in most of the literature measuring ICS exposure. This classifier includes an additional signature to detect Honeygrove honeypots [55] exposing default Modbus services. These signatures detected a total of 33 honeypots.

Odd. Last, we introduce a classifier to identify odd behaviors making services unrealistic. For EtherNet/IP, we check for serial numbers being reused or value 0 – an invalid serial number. In IEC 104, we flag hosts returning identical timestamped readings, and those responding with assigned devices in all the registers we probe (cf. Section 4.4). While simplistic, these classifiers could identify 240 odd hosts.

3.3. Volatility

Volatile objects show extreme reactions to slight changes. In the context of OT, we only expect to observe sudden changes from sensitive networks reacting defensively, e.g., with reactive blocking to unsolicited traffic. Otherwise, this type of behavior should be considered suspicious, including echoed responses and reactive measures common in deception techniques, such as MTD. Volatility detection addresses the question:

Does the host change its shape when observed?

Another common issue in Internet surveys is dealing with the Internet's ever-changing nature. Active scans are particularly susceptible to this issue, where most aspects of the study will produce noticeable differences, trading complexity for on-demand results (e.g., vantage point location and capabilities, experiment duration, time-frame,

and scan method). Issues such as Internet churn may produce double-counting errors or missing-by-chance problems, making regions appear more or less dense than they are. These issues may create the illusion that hosts flicker or change over time. In reality, there are several reasons to believe this behavior: devices are replaced, retired, undergo maintenance, or updated, among others.

However, a full branch of deception, MTD [8], suggests that volatility may in some cases be a deliberate defensive strategy. Far from echo servers that bounce back incoming requests with few to no changes, MTD solidifies the idea of changing properties of a system to deter attackers (e.g., its address or configuration). A key distinction from classical deception techniques is that MTD primarily serves as a proactive resource obfuscation method [8, 32]. MTD does not intend to create or provide false information; instead, it *moves things around* or obfuscates resources to hamper attacks. In this work, we do *not* claim to reliably detect MTD or to distinguish it from benign operational changes. Rather, we treat volatility as a generic indicator of instability that can arise from a variety of causes (e.g., churn, outages, maintenance, or deliberate defenses). We refer to this changing behavior as *volatility*, i.e., hosts, networks, and services that mutate when interacted with.

To test for volatility, we survey hosts multiple times. Multiple scans allow us to identify hosts that disappear, flicker, or mutate. In addition, sweeping the Internet multiple times may help us reduce Internet churn (e.g., the same host appears multiple times in the same prefix) and test for more advanced techniques such as MTD, where the location of the host may change. Given our limited number of snapshots, we only flag such behavior as volatile and refrain from attributing it to specific mechanisms. Volatile hosts are classified by comparing the results from our two scan iterations based on the following criteria.

Intermittent. First, we label hosts that become unavailable from our first scan to our second scan, which reduced the number of potential addresses by 1,340. Second, we label hosts that become available during the second scan, accounting for 911 hosts. This variance is not enough to justify scanning a third time, but may be worth looking over long periods to overcome network congestion and miss-by-chance issues, and temporary blocking measures.

Morphed. Our last criterion considers the properties of our probes and the targeted protocols, annotating hosts that return different values between scans (excepting temporal features). This was the case for: 90 EtherNet/IP hosts that refused to return identities for registered devices, 167 IEC 104 with varying information objects between scans, 13 Modbus devices with different coil values, and 2,308 Fox services mostly returning different ID numbers. In the case of Modbus, the slight changes could be associated with actual coil value changes over time. On the other hand – and to the best of our knowledge, – Fox ID values are static device unique identifiers within their network. Unlike other protocols, Fox device IDs are not generally unique; instead, these are simple indexing values to find devices quickly. Although we could not determine the reasons behind these changes, we observed this behavior quite commonly, always with a change of ID and often with changes in VM UUID. Changes in VM UUID refer to the Java VM, i.e., the instance identifier of the Fox application.

3.4. Hostility

The last behavioral property we differentiate is what we name as *hostility*. Hostile hosts are those that attempt to disrupt the client directly or indirectly without necessarily affecting the communication. Some examples include tarpits, broken responses, infect-back behaviors, and other attempts at pacification that would make hosts unrealistic. Hostile hosts answer the question:

> *Does the host try to actively disrupt the communication?*

Primary studies that conduct Internet-wide scans report that, during their experiments, they observed instances of hosts and networks attempting to disrupt their scanning campaign in a few particular ways: i) trapping connections in endless loops, ii) responding with malformed or flooding messages (e.g., attempting to allocate large or empty buffers for payloads, or responding with unknown options), iii) responding with malware, or iv) redirecting large amounts of traffic their way. Endless connection loops are a deception technique known as network *tarpits* [23]. Tarpits may trap clients in multiple ways, such as delaying communications, slowly responding with random data, or re-hooking clients with unsolicited handshake completions, which the community has labeled as *slowing* tarpits [3]. Some of these tarpits try to confuse stateless network scanners by sending up to a hundred TCP handshake completion packets [23], which could explain some of the behaviors experienced by the community. Other far more aggressive tarpit variants may block functions in their clients [57], commonly known as *sticky*. Sticky tarpits are dangerous, though; they react disproportionally to interaction, potentially disabling clients. This level of hostility is only matched by other malware-infected devices spreading passively (e.g., code injections and passive worms). Otherwise, this type of behavior is associated with pacifying attempts, anecdotally observed while scanning governmental or military networks.

This behavior is exceptionally uncommon in devices exposing real services, although it has been observed in infected devices attempting to propagate passively (e.g., infected websites with code injection vulnerabilities). However, regular infections do not qualify for this classification, as we only consider them as *noise* when the host has been completely replaced with decoys and other forms of malware traps – the objective of this study is to identify vulnerabilities. On the other hand, authors have proposed

similar approaches as means of deception and other sensible networks aimed at consuming resources from attackers, and we are interested in identifying such efforts in the wild, first as an indicator of their implementation and impact on scanning campaigns, and second as a source of noise. While underexplored, avoiding hostile hosts is already a critical factor in most active measurements, but is often discarded or not discussed in detail. Disruptions due to hostile hosts are often perceived as weaknesses in the scanning methodology, and there is a long tail of known measures to avoid them, such as limits to the time of the connection, volume of data received per response, blocking further traffic from the already scanned addresses, returning incomplete results along with raw data on corrupted packets, etc. However, there are cases where hostility can be easily confused with low availability or poor network quality. To distinguish hostile hosts, we expand on our volatility detection method: by requesting the same object multiple times, we can measure response deviations and reliably determine whether we observed a network artifact (e.g., availability issues) or the object shows signs of hostility.

Most of the noise proceeding from hostile hosts is filtered out as part of the incomplete and timeout responses. However, Modbus and IEC 104 services return streams of data frames that may trap scanners in endless connections if precautions are not properly implemented. To evaluate when our scanner is being manipulated into continuing to consume frames ad infinitum, our probes accept this behavior until the scanner forces the connection to time out. Then, we quantify the higher percentile ($> 95\%$) of frames received overall to determine which hosts abused this behavior. To summarize, timed-out IEC 104 communications transmitting more than 165 distinct IOAs with `TypeID` value 36 (stream reads of float + timestamp) are considered as tarpits, to a total of 62 hosts. For reference, most transmissions oscillated between 23 and 93 readings. Unfortunately, the probe used to gather Modbus results is not expressive enough to capture indicators of tarpits. The closest attempt we made was to analyze hosts returning a large number of objects as part of the MEI response, together with the *follow more* flag indicating the client to continue reading from the stream. This alone is not enough to form a conclusion.

4. Results

This section contains use-cases of vulnerability identification with noise detection for Modbus, Fox, IEC 104 and EtherNet/IP. Each use-case follows a systematic analysis approach, describing the probe used to gather results, processing method, and discussion of findings comparing results without noise. This analysis does not attempt to give an accurate representation of the actual number of vulnerable devices facing the Internet, but a method to identify them and consider the impact of noise in Internet surveys.

4.1. Overview

Terminology and counting conventions. We distinguish between (i) host–port observations (an IPv4 address observed on a specific default port) and (ii) unique hosts (unique IPv4 addresses, de-duplicated across ports/protocols). Our measurement follows a two-step workflow: an L4 sweep using ZMap to identify L4-positive host–port observations (SYN/ACK on the protocol's default TCP port), followed by an L7 sweep using ZGrab2 against the L4-positive set. We refer to ZGrab2 responses as host–port observations where a TCP connection was established and a protocol probe was sent (after filtering connection-attempt failures). We refer to exposed hosts as the subset of ZGrab2 responses where the probe completed and returned parseable protocol-specific data characterizing the service. Finally, we refer to potentially vulnerable hosts as the subset of exposed hosts that meet our protocol-specific vulnerability criteria. Unless explicitly stated otherwise, all stage counts reported in this paper are for unique IPv4 addresses, and protocol-specific counts are for host–port observations on that protocol's default port.

Table 4 summarizes the measurement at each stage. In this table, the *ZMap* and *ZGrab2* columns report *host–port observations* on each protocol's default TCP port (i.e., an IPv4 address can appear once per protocol). The *Total Unique Addresses* row de-duplicates IPv4 addresses across all protocols, so per-protocol row sums can exceed the total.

For ZMap, the total of 3,401,011 corresponds to unique IPv4 addresses that responded with a SYN/ACK on *at least one* of the four scanned ports (union across protocols). For ZGrab2, 2,533,236 denotes unique IPv4 addresses for which at least one ZGrab2 connection attempt succeeded (union across protocols), after filtering connection-attempt failures. Many of these successful connections still yield application-level errors, access denial messages, or responses that our probes cannot handle; these cases are included in ZGrab2 but excluded from Exposed. The remaining 19,183 Exposed hosts completed the probe exchange and returned parseable protocol-specific data. Finally, 18,558 hosts were classified as Vulnerable according to our protocol-specific criteria.

The criteria to classify hosts as Exposed, and subsequently as Vulnerable, are summarized in Table 5; Sections 4.2-4.5 provide algorithmic definitions. Our methodology is simple in nature, testing for authentication barriers preventing anonymous clients from establishing a communication channel, and sending discovery or otherwise informational requests to determine whether there is any form of access control. Note that our probes never attempted critical operations that would alter the state of the device, such as writing data to addresses or uploading documents to the device. We restate that only newer versions of the legacy Fox standard implement authentication and access control. In principle, all Exposed services allowing anonymous clients to communicate with the device

Table 4: Results by protocol and stage. **ZMap** and **ZGrab2** report host–port observations on the protocol's default TCP port (not de-duplicated across protocols). **Exposed** and **Vulnerable** report unique IPv4 addresses per protocol for which the probe completed and returned parseable data (Exposed) and that meet protocol-specific vulnerability criteria (Vulnerable). **Noise** columns count vulnerable IPv4 addresses flagged by each noise label; labels are *not mutually exclusive*. The final row (**Total Unique Addresses**) de-duplicates IPv4 addresses across all protocols.

Protocol	Port	Scan Results		Exposed	Vulnerable	Noise			
		ZMap	ZGrab2			Condensation	Displacement	Hostility	Volatility
Modbus	502	2,858,739	1,235,841	3,278	3,213	2,348	41	-	603
Fox	1911	2,818,650	10,264	8,516	8,516	3,618	395	-	2,688
IEC 104	2404	2,913,767	1,544,463	3,578	3,578	3,417	157	62	1,049
EtherNet/IP	44818	2,788,438	1,416,534	3,868	3,304	3,177	784	-	497
Total Unique Addresses	-	3,401,011	2,533,236	19,183	18,558	12,519	1,365	62	4,827

should be considered Vulnerable, since nothing prevents clients from using the device at their will. However, we noticed that some networks apply security features to selected commands.

Lastly, the intersection between exposed and GreyNoise yielded 201 hosts that recently scanned GreyNoise's network, with 127 of them sending malicious requests (e.g., attempting to authenticate to Telnet, SSH services, etc.) and 74 that scanned, contacted, or enumerated their networks. It appears that none of the hosts exposing EtherNet/IP were observed by GreyNoise's networks; the 201 malicious or suspicious hosts exposed primarily IEC 104 services, with few instances exposing Modbus (14) or Fox (4).

We note that GreyNoise correlation is performed at IP granularity and cannot attribute scanning activity to the specific OT service we observed. We therefore interpret matches only as 'this IP has been observed scanning,' not as direct evidence that the OT device initiated scans or is compromised.

4.2. Modbus

We use the default ZGrab probe as-is. This probe sends a *Read Device Identification* request for a single unit and object's ID (0), and handles the first Read Device Identification (MEI) response from the stream. The probe disregards Industrial PC (IPC) responses that do not fit in a single frame, which coincidentally avoids getting stuck in a stream loop. Without extending the probe to read more responses, we cannot induce a state where we can test for tarpitted connections abusing this vector. Figure 3 shows the differences between the request and response structures. Responses may include one or more objects, and fields to track the state of the stream, indicating whether the client should continue reading from the connection for further objects, and a tracking ID to order packets.

Similar to other OT probes, enumerating addresses is a trivial task beyond the first request. In addition, a single request may produce a stream of readings, adding unnecessary strain on remote devices. Therefore, the exercise of probing each address is left for further studies with such needs, and this paper is limited to a single address. To improve the precision of our noise detection method, we

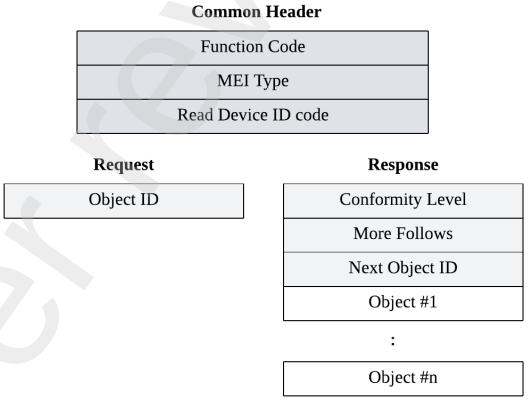


Figure 3: Modbus frame structures of reading requests and MEI responses with common headers. MEI responses may contain one or more objects, and may be split into multiple frames.

recommend that authors probe between two and four additional addresses, as adding randomness to the probe helps identify deception systems that respond to far more addresses than actual devices normally would. The major tradeoff from this approach is handling Modbus time delays between frames, which could lead to mistaking regular connection timeouts with tarpits (false negatives).

Algorithm 1 illustrates the procedure followed to classify Modbus services as exposed and vulnerable. In summary, devices responding with non-empty MEI responses are classified as exposed, and those including internal fields from where device information can be derived, such as firmware version and type of device, are classified as vulnerable, since these allow unauthorized clients to perform operations on the device and leak internal data that should only be available to their maintainers.

Our dataset contains responses from 3,278 Modbus stations with devices registered in the first address and responding to our device identification request. The information returned from each station varies depending on the vendor implementation, with few information objects in common. From these, we distinguish four objects that appear with relative frequency: vendor name, product code, firmware revision, and unit ID. These details provide sufficient information to profile linked devices; in total, our

Table 5: Protocol-specific classification criteria to consider services exposed and vulnerable. Except for newer versions of Fox, none of the protocols in the table implement security features by themselves, relying on firewalls, VPNs, and other external security measures to secure communications. Classification criteria follow the procedure of testing for authentication and access control.

Protocol	Classification	Criteria
Modbus	Exposed	Responds to Read Identification Requests
	Vulnerable	MEI response includes device information
Fox	Exposed	Hello response communication established
	Vulnerable	Hello response includes sensitive data only available to authorized clients
IEC 104	Exposed	Valid response to General Interrogation requests
	Vulnerable	General Interrogation contains ASDUs with IOA data
EtherNet/IP	Exposed	Successful ListIdentity requests
	Vulnerable	Identities includes vendor and product internal information

Algorithm 1: Classification algorithm for Modbus exposed and vulnerable services

Input: Response
Output: $(Exposed, Vulnerable)$
$Exposed \leftarrow 0$;
$Vulnerable \leftarrow 0$;
$Internal \leftarrow [vendor, product\_code, revision]$;
if *exists(Response.Objects)* **then**
  $Exposed \leftarrow 1$;
  if $Response.Objects \cap Internal \neq \emptyset$ **then**
    $Vulnerable \leftarrow 1$;
  end
end

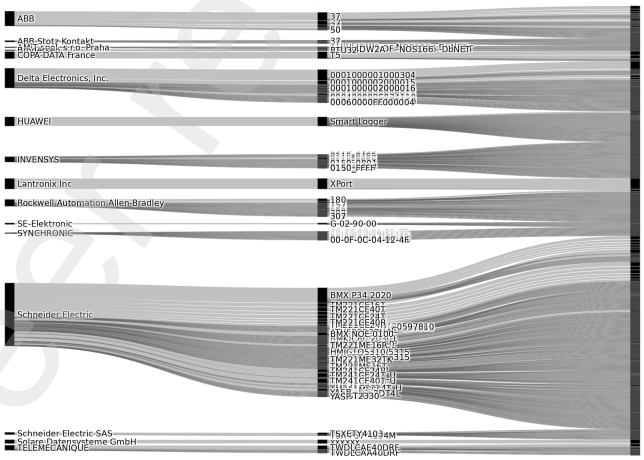


Figure 4: Modbus sankey diagram of vendors, products, and distribution of revisions occupying 75% of the dataset.

dataset contains 92 unique vendors and 279 products with 659 different firmware revisions. Figure 4 shows a representation of the most frequent vendor-product-revision relationships found in the datasets (95% of the results linked to just four vendors). Comparing these results with our previous study in [61] reveals minimal differences, mainly associated with a wider spread of vendors but lower product and revision diversity. The most significant difference is a reduction in obsolete products. However, sector-specific devices controlling power stations, solar panels, wind turbines, etc. are still found facing the Internet and not decreasing in numbers (e.g., Huawei SmartLoggers increased from 181 to 195). In addition, we still observe largely outdated devices, such as 325 out of the 328 `BMX P34 2020` PLC running on vulnerable firmware versions between `v1` and `v2`, while current revisions surpass `v3`. Though small number variations could be caused by our experimental setup, device availability, or other common limitations from Internet measurements, the overall picture remains in a similar state: outdated and vulnerable devices controlling critical systems are still facing the Internet.

As seen in Figure 5, the effect of noise is significantly lower than in other protocols. Our most reliable classifiers detected 41 hosts that appear to be in cloud environments, all of which present features that make Modbus services unrealistic. However, we could identify various gateways and VPNs that appear to be meant for the cloud. Open VPNs and other redirecting devices, such as gateways and routing systems exposing Modbus services to the wild are still counted as vulnerable, opening doors for attackers into their networks. These misclassifications are false negatives, giving reason for further inspection regardless of the assigned noise labels. About volatility, only 13 hosts responded with different values between scans, largely due to changes in reading values. The rest of the volatile hosts appeared intermittently, most of which became unavailable during the second scan. This raises two possibilities: either the services were in fact unavailable at the time of our experiments, or these networks actively dropped connections from our vantage point to their Modbus services. Since the intermittent behavior is three times higher in hosts missing during the second scan than during the first, the most likely explanation is that these hosts take an active role in disregarding connections. Whether this behavior is the result of implementing MTD techniques or merely in-place firewalls is still unknown. In summary, there were 41 hosts annotated under the conservative policy, 394 with balanced policy, and a total of 2,475 hosts

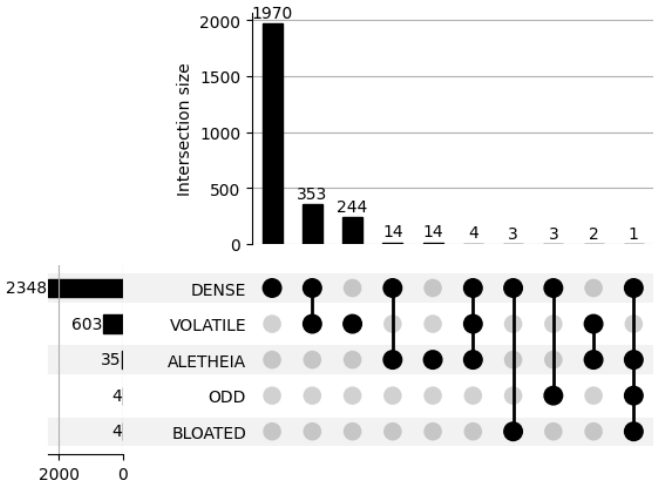


Figure 5: Modbus correlation diagram of noise labels.

using the aggressive policy.

4.3. Fox

Similarly to Modbus, we use the stock ZGrab2 probe to detect Fox services facing the Internet on Fox's default insecure port. This probe sends a static request with a client hello string (`fox a 1 -1 fox hello`) to gather basic device information. Responses are expected to be prefixed (`fox a 0 -1 fox hello`), indicating whether the server is a Fox service. Valid responses typically contain details, such as the name of the application running and various unique identifiers. Unsuccessful requests will include authorization errors or empty banners. Algorithm 2 shows the classification pipeline that handles the collected responses, where services are classified as exposed and vulnerable simultaneously as long as responses are valid and contain meta-data identifiers; this behavior is only possible when Fox services allow clients to communicate directly with the service, treating anonymous clients as authenticated.

Algorithm 2: Exposed Fox services classification

Input: Response
Output: $(Exposed, Vulnerable)$
$Exposed \leftarrow 0$;
$Vulnerable \leftarrow 0$;
if *exists(Response.version)* **then**
  $Exposed \leftarrow 1$;
  $Vulnerable \leftarrow 1$;
end

Fox is a building automation protocol for industrial environments to control alarms and security devices, sensors, switches, etc. One of the main distinctions with other OT protocols is that most versions of Fox implement authentication, role-based access control, and encryption (through TLS and WebSockets, named FOXS and FOXWSS respectively). Unfortunately, these features are optional, and while base Fox is not recommended (nor suitable) to communicate with remote and Internet-facing stations and workbenches, our results suggest that over 8K devices do not enable any of these security features. On the other hand, the addition of access control features and allowing for guest users limits the extent to which we can verify that these devices allow unknown users to control them. It is possible these devices allow guests to establish a communication channel, but do not allow them to perform any further action. Nevertheless, even allowing unknown guest clients to communicate with these devices poses significant risks, and only trusted clients should be able to communicate with them.

The identified Fox extensions are running on one of three operating systems: QNX (2393), Linux (361), and Windows (186). QNX is an operating system for embedded devices, indicating that those devices are outstations. In addition, all the identified devices were running on QNX versions below `v6.5`, preceding the next major version released in 2017. The same can be said for Linux-based stations, with the most current version observed of the kernel being `v4.4`. QNX `v6.5` and Linux `v4.4` reached their end-of-life in 2022. Those using Windows are spread through versions from XP to Windows 10, and various deprecated versions of Windows Server. Except for 48 Windows 10 workbenches, the rest run on deprecated and vulnerable Windows releases. Figure 6 shows the distribution of operating systems and their version found in Fox devices. Our dataset contains only 13 observed workbenches; the rest were identified as outstations. Anecdotally, a superficial analysis of station names reveals how the protocol is used to automate shopping stores and recognizable businesses, using their brand names for the devices, followed by the station's physical location. However, allowing unauthorized users to access this information comes with significant security and privacy risks.

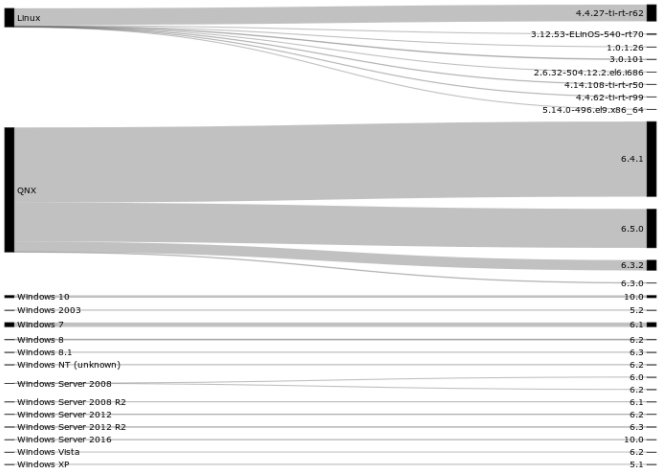


Figure 6: Fox distribution of operating systems and their versions.

Regarding noise, Fox stations appear to be more spread across the IPv4 space than other OT protocols. Stations are mostly located in US prefixes, with the largest cluster in ISPs offering data center solutions and cloud hosting services, and corresponding to 90% of the addresses labeled as *dense*. Despite this, the Aletheia method could only verify 14 addresses as located in cloud environments. In addition, all the stations labeled as volatile had changed their identifier during the second scanning round. To the best of our knowledge, this is not a common behavior in Fox applications; however, we are aware of these being encapsulated into Java Virtual Machines (Java VMs). Since we did not observe a pattern where the VM identifier would change alongside other identifiers, we cannot establish a definitive conclusion; our best estimation is that devices instantiate a separate application per connection, which would explain the changes in some identifiers. If that were the case, our volatile classifier for this protocol would need further adjustments. Figure 7 shows the correlation between the assigned labels. Overall, conservative policy annotates 395 hosts, combined signals in the balanced policy sum to 1,720, and 5,140 using the aggressive policy.

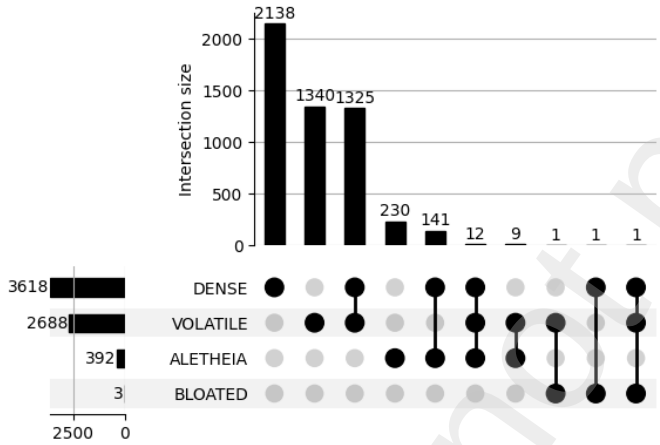


Figure 7: Fox correlation diagram of noise labels.

4.4. IEC 104

The probe to identify IEC 104 sends three different requests: i) a `TestFR` test frame, ii) a `StartDT` frame to establish a connection and confirm the server is willing to receive APDU frames, and iii) a General Interrogation command to request values of all objects in an address. In the third request, we scan for the following Common Addresses (CAs): 1, 2, 10, and 65535. These addresses represent the first two, the tenth, and the last valid address. The expectation is to observe different or empty values in the first and second addresses, none in the tenth, and, in most instances, a summary from the last address – some RTU implementations use the last address as a wildcard. Querying for more than one CA improves our chances of identifying deception systems that respond with random or identical values across all addresses, or attempt to trap our connection in a loop of potentially infinite APDUs. To better differentiate between honeypots and tarpits, we intentionally read all APDUs the server sends until we receive a termination ASDU or the connection times out. For illustration, Figure 8 shows the probe interrogation process to identify IEC 104 services, while Figure 9 describes the content structure of the APDU.

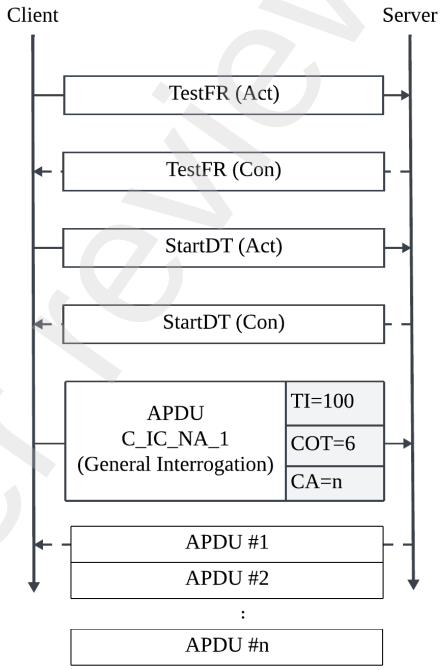


Figure 8: IEC 104 probe communication flow between the vantage point and remote hosts, sending `TestFR` and `StartDT` requests followed by a General Interrogation command.

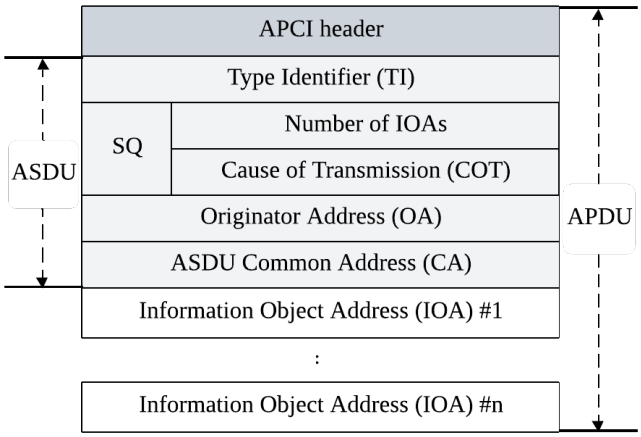


Figure 9: IEC 104 APDU structure of major fields with expanded ASDU section. A single frame may include multiple APDUs, each with an arbitrary number of IOAs, within the limits of the allowed frame size.

Algorithm 3 shows the classification pipeline for exposed and vulnerable devices. Exposed include valid responses for the probe's `TestFR`, `StartDT`, and General Interroga-

tion request; vulnerable devices respond with IOA data for one or more registered addresses.

Algorithm 3: Classification algorithm for IEC 104 exposed and vulnerable services

Input: Response
Output: $(Exposed, Vulnerable)$
$Exposed \leftarrow 0$;
$Vulnerable \leftarrow 0$;
if *exists(Response.Interrogation, Response.TestFR, Response.StartDT)* **then**
  $Exposed \leftarrow 1$;
  if $Response.Interrogation.APDUs \neq \emptyset$ **then**
    $Vulnerable \leftarrow 1$;
  end
end

Despite having gathered approximately 116K valid responses, only a fraction responded with APDUs. It is important to clarify that we did not enumerate all addresses, and the last address wildcard is only available in a handful of implementations. Therefore, devices using other addresses and not using the wildcard address will appear in our dataset as valid but empty responses. However, enumerating addresses is a trivial exercise once it is known the device exposes an IEC 104 service. Therefore, it is not advisable to either disable wildcard addresses or use random addresses to register devices as a security measure.

Further, in this paper only addresses that respond to arbitrary commands from unknown sources are considered vulnerable, for a total of 3,578 hosts exposing IEC 104 services in their default port. Probing with a General Interrogation command (`C_IC_NA_1`) is sufficient to prove the willingness of those services to accept commands. Additionally, while this command does not have side effects on the service, abusing it could cause DoS issues. The same is true for the rest of the commands, though, accepting others could change the device's state and pose major safety risks. A common example is the combination of `C_RP_NA_1` commands with `C_CD_NA_1`, which would reset the state of the device to default and delay requesting data from paired RTUs indefinitely.

Figure 10 shows the distribution of addresses the IEC 104 servers responded with. This distribution is mainly possible due to the use of the wildcard address 65535, returning a summary of the assigned addresses and streaming readings. As seen, most occupied addresses are allocated in the first 25 slots. It is important to remember that our probe explicitly requests information from the first, second, tenth, and last CAs (in that order), which are, in fact, the most represented addresses here. The rest of the CA values are artifacts returned as part of the wildcard address response. These results support our hypothesis of observing noise instead of real devices, considering the low probability of finding servers with all the requested addresses allocated.

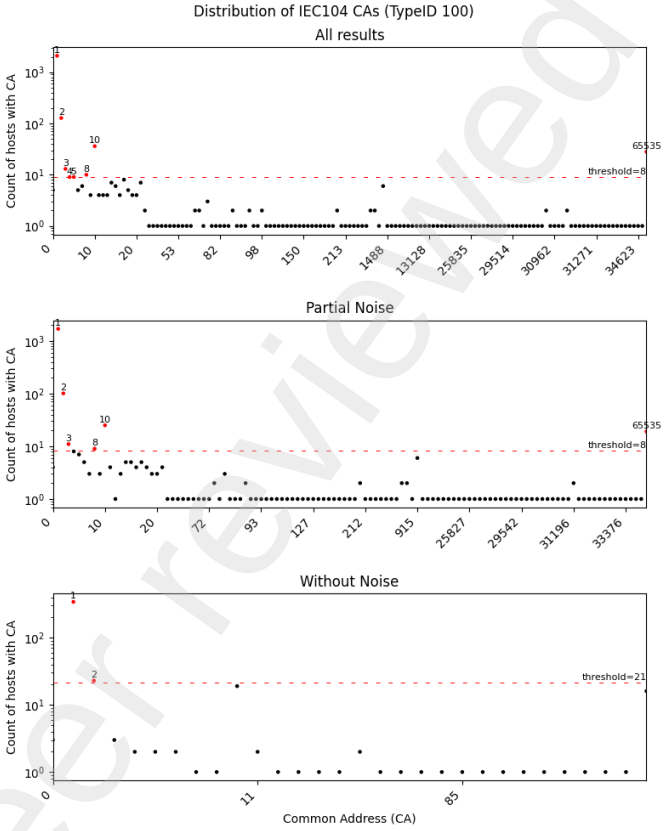


Figure 10: Distribution of IEC-104 CAs in: i) all results, ii) partial noise removed (w/o dense or volatile), and iii) removing all noise. Annotated CAs represent the most commonly found in our dataset across hosts.

Excepting the *dense* tag, the majority of tagged IEC 104 services include more than one tag. Figure 11 shows the correlation between tags, with the most common combinations of dense with Aletheia (cloud environments), tarpits, and known honeypots. Considering hosts not only tagged with the dense tag sums up to a total of 219; however, removing these hosts has no noticeable effect on the distribution of CAs, indicating that probing the first tens of addresses is still a reliable method to identify IEC 104 services, but restricting scans to a single address reduces the visibility of the scan by roughly 50% – $\tilde{5}$0% of our results are accumulated between the first two addresses. On the other hand, including all the tagged hosts effectively removes most services responding with random addresses past the 10th CA. To that, we must add 167 hosts returning different CAs between scans, and 882 showing intermittent behaviors, 393 of which stopped responding after the first scan. Volatile hosts have similar correlations as dense with other tags.

Overall, 219 of IEC 104 services can be safely considered noise using the conservative policy, 706 of the total show strong indicators of suspicion with two or more noise labels (balanced), and up to 3,475 if an aggressive policy is applied. A common trait among suspicious services is answering to random addresses beyond the tens, as we

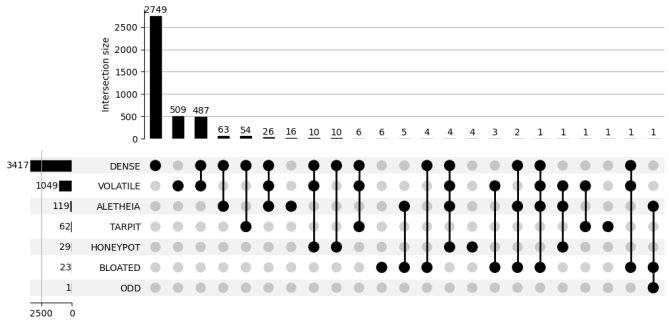


Figure 11: IEC 104 correlation diagram of noise labels.

observed a clear tendency to utilize lower addresses to register devices.

A distinctive characteristic from other protocols used in SCADA environments is that IEC 104 does not include dedicated discovery commands to request service descriptors. Unfortunately, this restricts the extent of our analysis and limits our ability to diagnose and discuss other security weaknesses in devices using IEC 104. The IEC 104 standard considered in this study (60870-5) is widely known for lacking basic security features for open networks, missing on access control measures to authenticate and (de-)authorize commands, and encryption to avoid rogue intermediary nodes listening for connections in transit. The newer IEC 62351 series of standards aims to provide these security features for protocols used in power systems and smart grids (e.g., Modbus, DNP3, and IEC 104) by implementing support for TLS and role-based access control at the gateways. Adopting standards with security in mind has proven to be challenging, and the literature has yet to study its progress. This poses an opportunity for further studies on the transition of OT legacy protocols to meet security standards (e.g., on the currently assigned secure ports for IEC 104 TCP/19998, and Modbus TCP/802). While the current standard under study continues to phase out, the security recommendations for these protocols remain the same: protect OT services behind VPNs and firewalls, segment networks with exposed devices, and disconnect or remove devices from the public Internet when Internet connectivity is not strictly needed.

4.5. EtherNet/IP

Our EtherNet/IP probe sends encapsulated Common Industrial Protocol (CIP) *ListIdentity*, *ListServices*, and *RegisterSession* requests, querying the device for general information (i.e., device serial number, manufacturer, and product name), capabilities, and access to resources [45]. This probe includes an identifier in the Sender Context field within the Encapsulation Header, which we use to notify servers of our presence and disregard servers that modify the field. Figure 12 provides a representation of the overall structure of EtherNet/IP response frames, showing the major fields included for device identities.

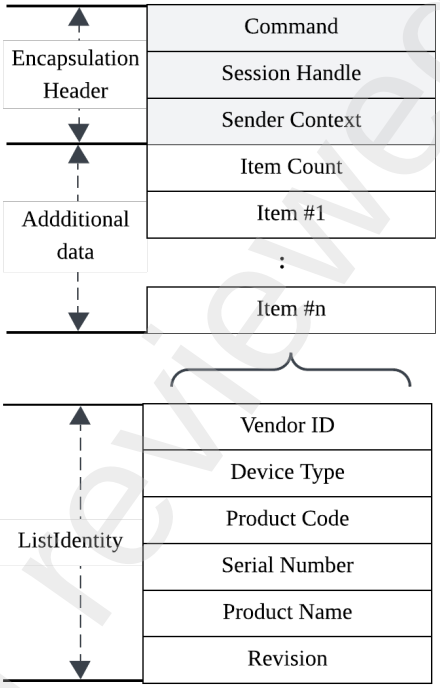


Figure 12: EtherNet/IP frame structure with major fields with response payload as *Additional data*. Additional data response structures vary for each command; the figure shows `ListIdentity` items and their informational fields.

We classify EtherNet/IP services as exposed when servers respond to the probe's `ListIdentity` request (i.e., one or more items) with valid response statuses. Responses only populate the Additional data when they accept incoming requests, and they can perform the command's action. Therefore, vulnerable devices are those that include one or more items in the Additional data field, and those items contain identifiable information, such as the vendor ID. The classification algorithm can be seen in Algorithm 4.

Algorithm 4: Classification algorithm for EtherNet/IP exposed and vulnerable services

Input: Response
Output: $(Exposed, Vulnerable)$
$Exposed \leftarrow 0$;
$Vulnerable \leftarrow 0$;
$Internal \leftarrow [vendor\_id]$;
if *exists(Response.Payload)* **then**
  $Exposed \leftarrow 1$;
  if $Response.Payload.Identities \cap Internal \neq \emptyset$ **then**
    $Vulnerable \leftarrow 1$;
  end
end

During our scans, we observed 3,868 hosts that responded positively to our probes with exactly one device

identity (i.e., a descriptor). Device identities always include static vendor and product type IDs, which Open DeviceNet Vendors Association (ODVA) assigns to certified members [46]. Vendors can choose to leave these fields empty or use placeholder IDs, while non-members may not adhere to this standard. Additionally, identities may include further product information, such as serial numbers, product names, and versioning. This information is useful for forming an impression of the population of devices exposing EtherNet/IP services to the Internet and helps us evaluate the risks their owners face, including details on their network and common behaviors. However, while this information has many benefits (e.g., asset discovery), EtherNet/IP device identities reveal critical information that is easily weaponized, highlighting the need for authentication and authorization.

Devices that freely respond to our unauthenticated probe with details of their internal infrastructure constitute a non-negligible risk. However, this factor alone is not sufficient to conclude on the severity. The EtherNet/IP standard specifies that certain types of devices can act as gateways or brokers for other devices, which should broadcast discovery requests to the devices on their network. By contrast, all addresses in our dataset responded with a single device identity regardless of the device type, likely due to some level of contingency – Rockwell Automation specifies that this configuration can be disabled [5]. However, a single controller may have a large number of adapters and other devices, such as switches and HMIs [44]. It should be noted that the majority of device serial numbers in our dataset were unique, which we use as a factor to determine when we encounter honeypots – duplicated serial numbers are a reason to be suspicious of an address. These behaviors can be used as additional heuristics to identify honeypots.

Overall, our dataset contains a total of 13 types of devices (5 that we could identify), 28 different vendors, and 382 distinct products. Figure 13 shows the distribution of vendors and types of devices (general distribution in dotted black). The distribution is skewed towards certain types of devices from particular vendors, denoting a trend among devices exposed to the Internet; however, it should not be confused with the actual popularity of a vendor or product, since our dataset only represents the observable Internet (i.e., devices responding to our probe sent from our particular vantage point). On the contrary, this type of trend helps us identify systemic security issues, such as common configurations or signs of widespread compromises.

In terms of the products themselves, Figure 14 offers a quick view of the most common identities in our dataset, creating a hierarchy that bundles devices by their vendor, product, and revision. Our dataset contains a wide heterogeneity of products known to be at different stages of their life-cycle. In addition, we find that their versions also vary wildly, from devices using early firmware versions to some updated to their latest version. Some of these firmware

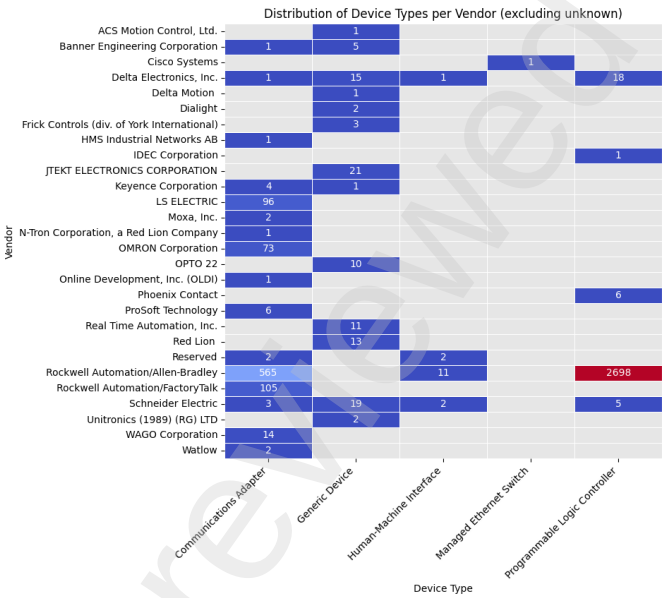


Figure 13: Distribution of EtherNet/IP vendors and device types.

versions contain vulnerabilities that have been known for years, with critical-level risk scores, and enabling attackers to take full control of the device with minimal interaction. Notably, approximately 73% of the products in our dataset were running on vulnerable firmware. In addition, only 58% of devices were currently receiving updates; the rest of the devices were either obsolete (e.g., the Rockwell Automation 1763-L16 product series, with more than 9 different associated Common Vulnerabilities and Exposures (CVEs) and discontinued in 2017) or had their end-of-life discontinuation date already announced. It is worth noting that only products in the active state of their life-cycle had no associated CVEs. Figure 15 shows the life-cycle of the top 10 products found in our dataset, where 7 out of the 10 products had their latest firmware update 5 years ago. ODVA issues a Declaration of Conformity to lines of products that pass their internal testing, and includes details such as dates for when these products were certified, their latest revision, and links to the vendor home page, where one can find details on the specific firmware versions, and product maturity level.

Furthermore, we identified 15 HMIs open to the Internet. Unprotected HMIs typically provide access to internal infrastructure, allowing easy control of other managed devices, such as valves and actuators. While these devices are known to be a common attack vector, no evidence leads us to think that these open HMIs pose a higher risk than PLCs or communication devices. However, since those devices give access to larger networks, the potential damage they could cause is equal to or larger than that of the other types of devices. Recent attacks on Internet-facing HMIs indicate that their exposure can be associated with symptoms of additional security issues.

The high concentration of devices in the USA may be

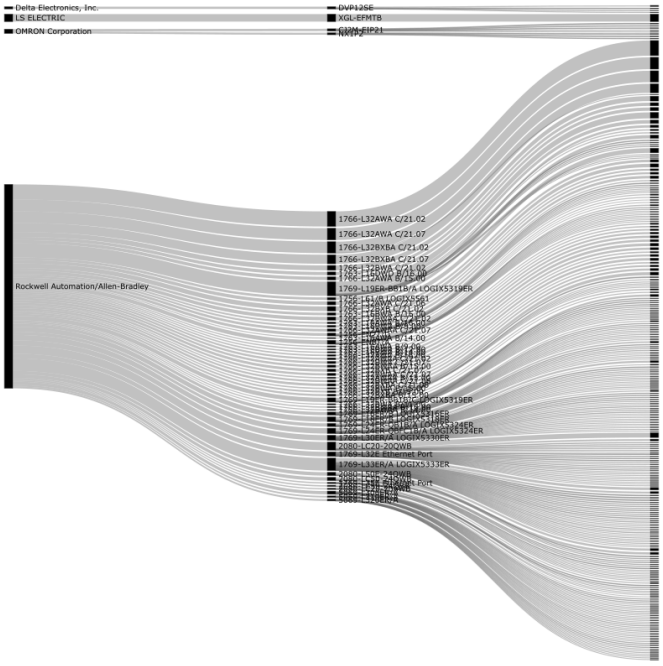


Figure 14: Distribution of EtherNet/IP vendors and products.

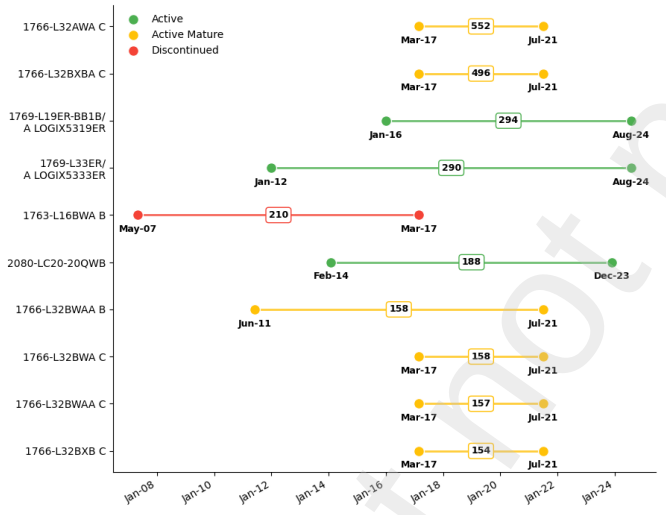


Figure 15: State of maturity of EtherNet/IP products found in our dataset.

linked to the market dominance of Rockwell Automation and the widespread standardization of EtherNet/IP in PLC products. Industrial software, such as FactoryTalk or Studio 5000, is tightly integrated with EtherNet/IP, which may explain why addresses expose both EtherNet/IP and other services simultaneously, such as web portals, OPC UA services, or gateway controllers in other ports.

Devices exposing EtherNet/IP interfaces to the Internet without the protection of any form of authentication, authorization, or encryption pose a significant risk in their network. These devices leak internal information easily exploitable, such as the name of the product and the firmware version. CIP Security specifies TLS (and DTLS for UDP communications), so deployments that leave EtherNet/IP services unprotected are operationally similar to legacy systems and are generally unsuitable for direct Internet exposure without compensating controls. Besides upgrading systems to enforce TLS or DTLS on Internet-facing devices – and not just support, – a simple remediation for this issue is to hide those devices behind a VPN, filtering unsolicited connections from legitimate ones, and providing encryption and authentication. In addition, maintainers can disable EtherNet/IP discovery services to control the level of information these devices give away. Moreover, our analysis revealed that a large majority of Internet-facing devices speaking EtherNet/IP suffer from various known vulnerabilities, are missing several updates, or are commonly in a state of obsolescence. Our recommendation to maintainers is to implement a security cycle of updating, upgrading, and eventually retiring devices before they reach their end-of-life. Generally, OT requires further security measures to keep devices protected, such as contingency areas, segmentation, firewalls, constant monitoring, and systems to identify ongoing threats and malware campaigns. Security in OT cannot be a one-time job.

In terms of noise, EtherNet/IP represents a particular case where correlations between sources are more common than in other protocols. Figure 16 illustrates these. The relationship between Aletheia's method, together with the bloated and dense labels, shows that deception techniques are more prevalent and easier to spot in EtherNet/IP. However, these devices are prefixes apart; the largest concentrations with these labels contain up to 13 hosts. Lastly, the hosts identified as volatile either responded with different status values or did not contain identities. In total, 784 are annotated following the conservative policy, 940 with balanced, and 3,344 using the aggressive policy.

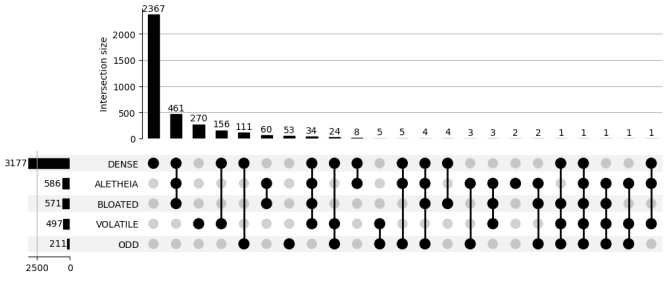


Figure 16: EtherNet/IP correlation diagram of noise labels.

5. Discussion

5.1. Implications for OT Internet measurements

Internet measurements that report on the security state of exposed devices should account for the presence of noise in their datasets. As shown in this paper, not all hosts

are real, and determining which ones are requires multiple iterations of analysis and evaluation. Fortunately, the effects of noise (and thus false positives) can be mitigated significantly.

The effects of simpler deception techniques with no interaction, such as telescopes and low-interaction honeypots, can be largely dismissed with our current probes. In our measurements, OT devices usually expose a single service at a time (and at most two in rare cases), and certain protocol combinations instantly raise suspicion (e.g., Fox and IEC 104 on the same host). In cases where more than one port is open at a time, we introduced noise detection classifiers for bloating that can be used to study these indicators further. In our dataset, the bloated classifier turned out to be a highly reliable label, identifying hosts with hundreds of open ports or multiple unlikely services combined. The same is true for our honeypot and odd classifiers for displacement, which mostly flagged honeypots with known signatures and services that were clearly not real, including repeated serial numbers in the case of EtherNet/IP and responses to more addresses than expected for IEC 104.

Not all classifiers, however, are equally reliable. The Aletheia method correctly classified all labeled hosts, but it missed many other known cloud networks. We consider this a limitation of the method more than anything else. Together with the bloated and condensation labels, these classifiers were intended to identify telescopes, cloud networks, and large clusters of unrealistic services. Our dense classifier, in particular, is currently too sensitive to be used alone and is better suited as a supporting label, i.e., as a warning. One possible explanation is that many of the services we observed are genuinely clustered in a few larger prefixes (e.g., EtherNet/IP and IEC 104), reflecting actual deployment choices. This in turn suggests that OT services may be hosted predominantly in particular prefixes and managed by a small number of ISPs. If this is the case, security negligence is not only a device-level problem but also a broader issue that extends to the operators of these prefixes.

Regarding volatility, the effects of this type of noise are better studied in longitudinal studies. Intermittent behaviors were not symmetrical in our snapshot, with far more hosts failing to respond during the second scan than during the first. Multiple possible reasons make this analysis inconclusive, such as blocking behaviors and availability issues. We also could not identify the reasons behind hosts responding differently between scans. While this could be linked to MTD, since the behavior does not apply to the majority of hosts, our method is not sufficient to be certain, and we interpret volatility as an indicator of instability rather than definitive evidence of MTD.

Even without considering hosts labeled only with dense or volatile classifiers (low-confidence signals), and focusing only on the more reliable noise classifiers (high-confidence), we observed significant noise levels for all protocols, ranging between an overall 7% across hosts, and raising up to 20% for particular protocols of the hosts that we suspect are vulnerable, depending on the service (i.e., lacking access control, lacking encryption, having known vulnerabilities, or being obsolete). Considering hosts annotated with more aggressive policies rise these numbers significantly. These percentages should be used as benchmarking baselines for future studies that aim to account more thoroughly for the impact of noise when reporting on vulnerability assessments of exposed OT networks.

Overall, characterizing security weaknesses is challenging, but it plays a crucial role. Comparing and contrasting the state of a given network against the visible Internet produces valuable insights that help mitigate vulnerabilities early (e.g., information leaks, a map of the attack surface, and similarities with other exposed devices). However, deception systems and other sources of noise can distort our view of the Internet. False positives pollute datasets and create a misleading representation of the Internet; reporting on these findings puts undeserved strain on network administrators, vendors, and device owners. At the same time, the opposite problem also exists, with many observations tending toward inconclusive results, which in the worst cases can produce false negatives. Some may argue that in the case of OT networks, and particularly for critical infrastructure, overestimating risk may be more beneficial than being conservative, even at the cost of receiving false alerts.

For the benefit of the field, and to produce more reliable measurements, future studies should provide additional guarantees for their results. With further improvements, the noise classifiers provided in this study could be used to provide more robust measurements and to better mitigate the threats posed by insecure OT networks exposed to the Internet.

5.2. Threats to validity and limitations

Our study has some limitations that should be taken into account when interpreting the results. First, all scans were conducted from a single institutional vantage point that has been used in multiple prior measurement campaigns and is visible in external reputation services. As a consequence, some networks may block or throttle our traffic, or treat it differently from other scanners, which can bias both the set of observable hosts and their behavior. In addition, we honor a blocklist that covers a nontrivial fraction of the routable IPv4 space, meaning that OT deployments behind those prefixes remain outside our visibility.

Second, our results are based on a snapshot spanning two Internet-wide scan iterations within a relatively short time window. This limits our ability to characterize long-term dynamics and, in particular, to cleanly separate transient churn, maintenance windows, and temporary blocking from deliberate mechanisms such as MTD. For this reason, volatility should be interpreted as an indicator of instability rather than as proof of active deception or configuration changes.

Third, our noise classifiers are heuristic by design. The condensation classifier, for example, is intentionally aggressive and should be interpreted as a warning signal rather than a definitive honeypot or telescope detector, especially in environments where dense OT deployments may be legitimate. Similarly, Aletheia misses some known cloud networks, and we do not provide a systematic quantification of false positives and false negatives for each classifier. Our labels therefore represent strong indications of noise, not ground truth.

Fourth, we focus on services exposed on the default ports of Modbus, Fox, IEC 104, and EtherNet/IP. Devices using nonstandard ports, being shielded by VPNs or jump hosts, or deployed behind NAT, as well as deployments using proprietary or vendor-specific extensions, are not captured by our scan. As a result, our measurements characterize the exposed Internet-facing surface of legacy OT deployments rather than the full population of deployed devices.

Finally, we restrict ourselves to four widely used OT protocols. Other industrial or building automation protocols, as well as higher-layer application logic and organizational processes, may exhibit different exposure patterns and noise characteristics. We therefore caution against overgeneralizing our quantitative estimates beyond the protocols and time frame studied here. Nevertheless, the methodology and noise taxonomy we propose are applicable to other protocols and future measurement campaigns.

6. Related Work

This section covers the body of work in Internet measurements, with a particular focus on studies investigating OT exposure and its security. The methods and techniques used to survey the Internet have advanced at an increased pace since Internet-wide L4 scanners became widely available [14]. The literature now contains hundreds of studies using L4 and L7 active scanning tools (e.g., Masscan, ZMap, and ZGrab2), results from CTI services (e.g., Shodan, Censys, GreyNoise), and passively collected datasets through network telescopes, honeypots, and other privileged vantage points. These collective efforts contributed to develop best practices for designing experiments and conducting Internet measurements [41].

While gaining momentum, the current state of the literature contains a moderate number of publications discussing the security of OT facing the Internet [49]. The scope of the work and terminology referring to OT systems has evolved significantly over the years, whereas the earlier work focused solely on SCADA networks, then expanded to ICS, and is currently moving towards the more general term of OT. Many of these studies use passive scanning approaches (e.g., traffic through Internet backbone infrastructure and network telescopes) [17, 42, 6] or use deception systems to collect information (e.g., ICS honeypots) [33, 52, 26]. Others have measured the use of deception systems in OT [65, 53, 39], and even fewer conduct active Internet surveys using stateful probes (i.e., sweep scans followed by banner-grabs) [60, 11, 12]. Authors have raised concerns about OT environments being too sensitive for traditional scans, and that communicating with these networks could be catastrophic [18]. However, this theory does not explain how these systems can persist for long periods exposed to the Internet and remain unnoticed by their owners. Coffey et al. [9] found no evidence of network degradation or abnormal behavior from using aggressive banner grabbing tools such as Nmap. One of the main weaknesses of the argument is that studies using passive collection methods observe large traffic loads towards services commonly used in OT – and this trend increases with every new study. Another is that limiting our view of the Internet to passive methods results in a poor understanding of the security issues not observed passively, which reduces security to a reactive approach. Passive and active measurements are complementary to one another, and both are necessary to understand and mitigate the ongoing security challenges particular to OT.

The majority of OT security issues we continue studying today have been known and exploited for decades: lack of access control and encryption, data leaks, use of legacy devices, device fragility, etc. Igure et al. [25] covered some of these for currently widely used protocols with thousands of devices facing the Internet today (i.e., Modbus, EtherNet/IP, IEC 104). Ghosh and Sampalli [21] echo these lessons on a recent survey of security of SCADA networks, expanding on possible attacks, their effects, and countermeasures. Their survey also contains a comparison of security standards, which the authors critique for lacking encryption schemes safe against quantum attacks. While DoS had been one of the strongest focuses in the literature, the work of Nicholson et al. [43] reemphasizes the core security issues in OT, summarizing some of the most relevant attacks, their vectors, and consequences on a study of SCADA security in cyber-warfare (i.e., lack of authentication, misconfigurations, and outdated software). In addition, the authors include a daring analysis of the security posture of device vendors and manufacturers, showing that some of the major incidents in OT, such as Stuxnet, were partly due to vendor/manufacturer bad security practices (e.g., failing to fix critical vulnerabilities in time and hardcoding credentials).

Regarding Internet measurements, the work of Mirian et al. [38] is a notorious example for their contributions bringing attention to the state of exposed ICS networks, covering multiple widely used protocols such as Modbus and Fox, among others. In their work, the authors uncovered more than 60,000 vulnerable systems from a wide range of organizations, including critical infrastructure. The authors complement their findings with a network telescope to provide an analysis of ongoing attacks exploiting the protocols covered in their study, showing that, at the time, most of the traffic they could observe originated from research institutions and security firms. Feng et al. [18] covered 17 ICS protocols and implemented some

of the first honeypot fingerprinting techniques; however, these deception systems are treated as noise and never quantified.

In a study on the security of OT and Internet of Things (IoT) exposed systems, Dahlmanns et al. [12] found that only 6.5% of Internet-facing OT devices – speaking one of 7 different protocols – encrypted their communications, and 42% of those were insecurely configured. The authors claim that deception systems and other sources of noise do not affect their results. Yaben et al. [60] further develops this by identifying vulnerable systems exhibiting symptoms of precarious security management, searching for misconfigured, seemingly abandoned, or obsolete devices. Their major contribution is the granularity of their analysis and worrying message: most devices they found show critical vulnerabilities that do not require complex exploitation methods (e.g., lack of authentication, encryption, or are missing major updates). Their work only filters deception systems as classified by Shodan. Others dedicate their efforts to more complex protocols, as the case of Dahlmanns et al. [11], where they revealed that 92% of OPC UA servers were misconfigured, exhibiting problems such as disabled security, reliance on deprecated cryptographic primitives, or unauthenticated access. Yaben and Vasilomanolakis [62] conducted a similar study revisiting the state of OPC UA, diving into further details and comparing results with previous studies, showing that nearly 25% of vulnerable servers remain unchanged year after year. While both studies use a similar data collection method, neither considers noise in their datasets.

In a similar vein, there is a number of studies that rely on the results from CTI services for their analysis. One of the most notorious examples is the work of [15], which uses data from Censys to report on their findings. [59] leveraged Censys instead of active probing, tracking five ICS protocols between 2015–2017 and identifying nearly 68,000 devices, with a clear trend of increasing exposure over time. Another study [4] examined Shodan, Censys, and BinaryEdge using both vendor- and protocol-specific queries. The results demonstrated tool-specific strengths: Censys returned the highest number of banners for vendor based queries, whereas BinaryEdge was more effective for protocol based queries. Identified devices were further classified into categories such as PLCs, RTUs, HMIs, and SCADA systems. It is important to mention that Shodan and Censys do not share their deception identification methods, and their evaluation is reported using a confidence percentage or labels (e.g., tarpit, or honeypot).

OT networks urgently need methods to evaluate and monitor their security remotely, on demand, and at scale. We recently proposed a framework aimed at this particular issue [63]; this study uses a simplified version of the proposed one to analyze results offline. An obvious contribution towards this goal is to create probes targeting OT protocols. To this date, only a fraction of the most commonly used OT protocols are included in the literature, such as Modbus, S7, DNP3, BACnet, or Fox. On the other hand, other protocols with similar adoption were largely understudied, such as EtherNet/IP and IEC 104 – which we covered in this study. However, while developing new methods to cover more systems and with better accuracy has proven highly beneficial to advance our knowledge of the Internet, the findings of Srinivasa et al. [53] and Mladenov et al. [39] call into question the results reported from studies choosing to ignore the prevalence of deception systems and unrealistic observations. Therefore, we should revise our current methods to evaluate OT exposure and measure what matters.

7. Conclusion

This paper revisited the exposure of OT networks on the public Internet through a noise-aware lens. Using Internet-wide scans of Modbus, Fox, IEC 104 and EtherNet/IP, complemented with AS-level metadata and CTI sources, we proposed and instantiated a taxonomy of noise in terms of condensation, displacement, volatility and hostility, and showed that a non-trivial fraction of ostensibly exposed OT services are artifacts of honeypots, telescopes, tarpits and other deceptive or anomalous infrastructures. Even after filtering out 7% of the total observations, and up to 20% for particular protocols as likely noise, we still observed thousands of legacy and misconfigured devices directly reachable from the Internet, which confirms that the OT attack surface remains dangerously large. These percentages are specific to our vantage point and time window and should be interpreted as indicative baselines rather than universal constants; we expect noise levels to vary across networks and over time. Our classifiers and probes are made publicly available to support more reliable future measurements, although several are conservative and would benefit from longitudinal validation and extension to additional protocols. We hope this work encourages both researchers and operators to treat noise as a first-class concern when quantifying OT exposure and to base risk assessments and mitigations on measurements that better reflect what matters in the real Internet.

Acknowledgment

This work is part of the project *Digital ghost ships: unveiling the threat of misconfigured and obsolete systems*, funded by the Independent Research Fund Denmark (grant number: 2035-00030B).

References

[1] AbuseIPDB. 2025. AbuseIPDB. `https://www.abuseipdb.com/` [Online; accessed 2025-12-11].

[2] Alexander Adamov, Anders Carlsson, and Tomasz Surmacz. 2019. An Analysis of LockerGoga Ransomware. In *2019 IEEE East-West Design & Test Symposium (EWDTS)*. IEEE, Batumi, Georgia, 1–5. `https://doi.org/10.1109/EWDTS.2019.8884472`

[3] Lance Alt, Robert Beverly, and Alberto Dainotti. 2014. Uncovering network tarpits with degreaser. In *Proceedings of the 30th Annual Computer Security Applications Conference* (New Orleans, Louisiana, USA) *(ACSAC '14)*. Association for Computing Machinery, New York, NY, USA, 156–165. https://doi.org/10.1145/2664243.2664285

[4] Travis Ashley, Sri Nikhil Gupta Gourisetti, Newton Brown, and Christopher Bonebrake. 2022. Aggregate attack surface management for network discovery of operational technology. *Computers & Security* 123 (2022), 102939. https://doi.org/10.1016/j.cose.2022.102939

[5] Rockwell Automation. 2025. Rockwell - EtherNet/IP Network Devices. https://literature.rockwellautomation.com/idc/groups/literature/documents/um/enet-um006_-en-p.pdf [Online; accessed 2025-08-30].

[6] Giovanni Barbieri, Mauro Conti, Nils Ole Tippenhauer, and Federico Turrin. 2021. Assessing the Use of Insecure ICS Protocols via IXP Network Traffic Analysis. In *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, Athens, Greece, 1–9. https://doi.org/10.1109/ICCCN52240.2021.9522219

[7] Kevin Borders, Laura Falk, and Atul Prakash. 2007. OpenFire: Using deception to reduce network attacks. In *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*. IEEE, Nice, France, 224–233. https://doi.org/10.1109/SECCOM.2007.4550337

[8] Jin-Hee Cho, Dilli P Sharma, Hooman Alavizadeh, Seunghyun Yoon, Noam Ben-Asher, Terrence J Moore, Dong Seong Kim, Hyuk Lim, and Frederica F Nelson. 2020. Toward proactive, adaptive defense: A survey on moving target defense. *IEEE Communications Surveys & Tutorials* 22, 1 (2020), 709–745.

[9] Kyle Coffey, Richard Smith, Leandros Maglaras, and Helge Janicke. 2018. Vulnerability analysis of network scanning on SCADA systems. *Security and Communication Networks* 2018, 1 (2018), 3794603.

[10] Arthur Cordeiro and Emmanouil Vasilomanolakis. 2025. Towards Agnostic Operational Technology (OT) Honeypot Fingerprinting. In *2025 9th Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, Copenhagen, Denmark, 1–4. https://doi.org/10.23919/TMA66427.2025.11097018

[11] Markus Dahlmanns, Johannes Lohmöller, Ina Berenice Fink, Jan Pennekamp, Klaus Wehrle, and Martin Henze. 2020. Easing the Conscience with OPC UA: An Internet-Wide Study on Insecure Deployments. In *Proceedings of the ACM Internet Measurement Conference* (Virtual Event, USA) *(IMC '20, Vol. 1)*. Association for Computing Machinery, New York, NY, USA, 101–110. https://doi.org/10.1145/3419394.3423666

[12] Markus Dahlmanns, Johannes Lohmöller, Jan Pennekamp, Jörn Bodenhausen, Klaus Wehrle, and Martin Henze. 2022. Missed Opportunities: Measuring the Untapped TLS Support in the Industrial Internet of Things. *Asia Ccs 2022 - Proceedings of the 2022 Acm Asia Conference on Computer and Communications Security* 1 (2022), 252–266. https://doi.org/10.1145/3488932.3497762

[13] Nicholas DeMarinis, Stefanie Tellex, Vasileios P. Kemerlis, George Konidaris, and Rodrigo Fonseca. 2019. Scanning the Internet for ROS: A View of Security in Robotics Research. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, Montreal, QC, Canada, 8514–8521. https://doi.org/10.1109/ICRA.2019.8794451

[14] Zakir Durumeric, David Adrian, Phillip Stephens, Eric Wustrow, and J. Alex Halderman. 2024. Ten Years of ZMap. In *Proceedings of the 2024 ACM on Internet Measurement Conference* (Madrid, Spain) *(IMC '24)*. Association for Computing Machinery, New York, NY, USA, 139–148. https://doi.org/10.1145/3646547.3689012

[15] Zakir Durumeric, Hudson Clark, Jeff Cody, Elliot Cubit, Matt Ellison, Liz Izhikevich, and Ariana Mirian. 2025. Censys: A Map of Internet Hosts and Services. In *Proceedings of the ACM SIGCOMM 2025 Conference* (São Francisco Convent, Coimbra, Portugal) *(SIGCOMM '25)*. Association for Computing Machinery, New York, NY, USA, 147–163. https://doi.org/10.1145/3718958.3754344

[16] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast internet-wide scanning and its security applications. *Proceedings of the 22nd Usenix Security Symposium* 1 (2013), 605–619.

[17] Claude Fachkha, Elias Bou-Harb, Anastasis Keliris, Nasir D Memon, and Mustaque Ahamad. 2017. Internet-scale Probing of CPS: Inference, Characterization and Orchestration Analysis. In *NDSS*. NDSS, San Diego, CA, USA, 1–15.

[18] Xuan Feng, Qiang Li, Haining Wang, and Limin Sun. 2016. Characterizing industrial control system devices on the Internet. *Proceedings - International Conference on Network Protocols, Icnp* 2016- (2016), 7784407. https://doi.org/10.1109/ICNP.2016.7784407

[19] MushMush Foundation. 2025. Conpot: ICS/SCADA honeypot. https://github.com/mushorg/conpot [Online; accessed 2025-11-26].

[20] Jérôme Francois, Olivier Festor, et al. 2008. Activity monitoring for large honeynets and network telescopes. *International Journal on Advances in Systems and Measurements* 1, 1 (2008), 1–13.

[21] Sagarika Ghosh and Srinivas Sampalli. 2019. A Survey of Security in SCADA Networks: Current Issues and Future Challenges. *IEEE Access* 7 (2019), 135812–135831. https://doi.org/10.1109/ACCESS.2019.2926441

[22] GreyNoise. 2025. GreyNoise. https://viz.greynoise.io/ [Online; accessed 2025-12-11].

[23] Harm Griffioen and Christian Doerr. 2023. Could you clean up the Internet with a Pit of Tar? Investigating tarpit feasibility on Internet worms. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, USA, 2551–2565. https://doi.org/10.1109/SP46215.2023.10179467

[24] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. 2008. Census and survey of the visible internet. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement* (Vouliagmeni, Greece) *(IMC '08)*. Association for Computing Machinery, New York, NY, USA, 169–182. https://doi.org/10.1145/1452520.1452542

[25] Vinay M. Igure, Sean A. Laughter, and Ronald D. Williams. 2006. Security issues in SCADA networks. *Computers & Security* 25, 7 (2006), 498–506. https://doi.org/10.1016/j.cose.2006.03.001

[26] Arthur Jicha, Mark Patton, and Hsinchun Chen. 2016. SCADA honeypots: An in-depth analysis of Conpot. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. IEEE, Tucson, AZ, USA, 196–198. https://doi.org/10.1109/ISI.2016.7745468

[27] Stamatis Karnouskos. 2011. Stuxnet worm impact on industrial cyber-physical system security. In *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, Melbourne, VIC, Australia, 4490–4494. https://doi.org/10.1109/IECON.2011.6120048

[28] Rafiullah Khan, Peter Maynard, Kieran McLaughlin, David Laverty, and Sakir Sezer. 2016. Threat analysis of BlackEnergy malware for synchrophasor based real-time control and monitoring in smart grid. In *Proceedings of the 4th International Symposium for ICS & SCADA Cyber Security Research 2016* (Belfast, United Kingdom) *(ICS-CSR '16)*. BCS Learning & Development Ltd., Swindon, GBR, 1–11. https://doi.org/10.14236/ewic/ICS2016.7

[29] John Knight, Jack Davidson, Anh Nguyen-Tuong, Jason Hiser, et al. 2016. Diversity in cybersecurity. *Computer* 49, 04 (2016), 94–98.

[30] Pavel Kozak, Ivo Klaban, and Tomáš Šlajs. 2023. Industroyer cyber-attacks on Ukraine's critical infrastructure. In *2023 International Conference on Military Technologies (ICMT)*. IEEE, Brno, Czech Republic, 1–6. https://doi.org/10.1109/ICMT58149.2023.10171308

[31] Ralph Langner. 2011. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE security & privacy* 9, 3 (2011), 49–51.

[32] Cheng Lei, Hong-Qi Zhang, Jing-Lei Tan, Yu-Chen Zhang, and Xiao-Hu Liu. 2018. Moving target defense techniques: A survey. *Security and Communication Networks* 2018, 1 (2018), 3759626.

[33] Sam Maesschalck, Vasileios Giotsas, and Nicholas Race. 2021. World wide ICS honeypots: A study into the deployment of conpot honeypots. In *Industrial Control System Security Workshop*. ICSS, virtual, 1–10.

[34] Alexander Männel, Jonas Mücke, K. C. Claffy, Max Gao, Ricky K. P. Mok, Marcin Nawrocki, Thomas C. Schmidt, and Matthias Wählisch. 2025. Lessons Learned from Operating a Large Network Telescope. In *Proceedings of the ACM SIGCOMM 2025 Conference* (São Francisco Convent, Coimbra, Portugal) *(SIGCOMM '25)*. Association for Computing Machinery, New York, NY, USA, 826–841. https://doi.org/10.1145/3718958.3754347

[35] John Matherly. 2025. Shodan: The Search Engine for the Internet of Things. https://www.shodan.io/ Accessed: 2025-08-17.

[36] Yassine Mekdad, Giuseppe Bernieri, Mauro Conti, and Abdeslam El Fergougui. 2022. The Rise of ICS Malware: A Comparative Analysis. In *Computer Security. ESORICS 2021 International Workshops*, Sokratis Katsikas, Costas Lambrinoudakis, Nora Cuppens, John Mylopoulos, Christos Kalloniatis, Weizhi Meng, Steven Furnell, Frank Pallas, Jörg Pohle, M. Angela Sasse, Habtamu Abie, Silvio Ranise, Luca Verderame, Enrico Cambiaso, Jorge Maestre Vidal, and Marco Antonio Sotelo Monge (Eds.). Springer International Publishing, Cham, 496–511.

[37] Lionel Metongnon and Ramin Sadre. 2018. Beyond Telnet: Prevalence of IoT Protocols in Telescope and Honeypot Measurements. In *Proceedings of the 2018 Workshop on Traffic Measurements for Cybersecurity* (Budapest, Hungary) *(WTMC '18)*. Association for Computing Machinery, New York, NY, USA, 21–26. https://doi.org/10.1145/3229598.3229604

[38] Ariana Mirian, Zane Ma, David Adrian, Matthew Tischer, Thasphon Chuenchujit, Tim Yardley, Robin Berthier, Joshua Mason, Zakir Durumeric, J. Alex Halderman, and Michael Bailey. 2016. An Internet-wide view of ICS devices. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, Auckland, New Zealand, 96–103. https://doi.org/10.1109/PST.2016.7906943

[39] Martin Mladenov, László Erdődi, and Georgios Smaragdakis. 2025. All that Glitters is not Gold: Uncovering Exposed Industrial Control Systems and Honeypots in the Wild. In *2025 IEEE 10th European Symposium on Security and Privacy (EuroS&P)*. IEEE, Venice, Italy, 133–152. https://doi.org/10.1109/EuroSP63326.2025.00017

[40] Shun Morishita, Takuya Hoizumi, Wataru Ueno, Rui Tanabe, Carlos Gañán, Michel J.G. van Eeten, Katsunari Yoshioka, and Tsutomu Matsumoto. 2019. Detect Me If You... Oh Wait. An Internet-Wide View of Self-Revealing Honeypots. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, Arlington, VA, USA, 134–143.

[41] David Myers, Ernest Foo, and Kenneth Radke. 2015. Internet-wide scanning Taxonomy and Framework. *Conferences in Research and Practice in Information Technology Series* 161 (2015), 61–65.

[42] Marcin Nawrocki, Thomas C. Schmidt, and Matthias Wählisch. 2020. Uncovering Vulnerable Industrial Control Systems from the Internet Core. arXiv:1901.04411 [cs.NI] https://arxiv.org/abs/1901.04411

[43] A. Nicholson, S. Webber, S. Dyer, T. Patel, and H. Janicke. 2012. SCADA security in the light of Cyber-Warfare. *Computers & Security* 31, 4 (2012), 418–436. https://doi.org/10.1016/j.cose.2012.02.009

[44] ODVA. 2011. Securing EtherNet/IP™ Networks. https://www.odva.org/wp-content/uploads/2020/05/PUB00269R1.1_ODVA-Securing-EtherNetIP-Networks.pdf#page=6.66 [Online; accessed 2025-09-01].

[45] ODVA. 2016. The Common Industrial Protocol (CIP™) and the Family of CIP Networks. https://www.odva.org/wp-content/uploads/2020/06/PUB00123R1_Common-Industrial_Protocol_and_Family_of_CIP_Networks.pdf [Online; accessed 2025-10-20].

[46] ODVA. 2025. ODVA Vendor ID data. https://marketplace.odva.org/vid.dat [Online; accessed 2025-08-30].

[47] Morteza Safaei Pour, Joseph Khoury, and Elias Bou-Harb. 2022. HoneyComb: A Darknet-Centric Proactive Deception Technique For Curating IoT Malware Forensic Artifacts. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium* (Budapest, Hungary). IEEE Press, Budapest, Hungary, 1–9. https://doi.org/10.1109/NOMS54207.2022.9789827

[48] RIPE. 2025. RIPE Atlas - Dashboard. https://atlas.ripe.net/ [Online; accessed 2025-12-12].

[49] Scopus. 2025. Scopus - Document search results. https://bit.ly/44saqQd [Online; accessed 2025-12-14].

[50] Rishabh Singla, Shreyas Srinivasa, Narasimha Reddy, Jens Myrup Pedersen, Emmanouil Vasilomanolakis, and Riccardo Bettati. 2023. An Analysis of War Impact on Ukrainian Critical Infrastructure Through Network Measurements. In *2023 7th Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, Naples, Italy, 1–10. https://doi.org/10.23919/TMA58422.2023.10199005

[51] Shreyas Srinivasa, Jens Myrup Pedersen, and Emmanouil Vasilomanolakis. 2021. Open for hire: attack trends and misconfiguration pitfalls of IoT devices. In *Proceedings of the 21st ACM Internet Measurement Conference (IMC '21)*. Association for Computing Machinery, New York, NY, USA, 195–215. https://doi.org/10.1145/3487552.3487833

[52] Shreyas Srinivasa, Jens Myrup Pedersen, and Emmanouil Vasilomanolakis. 2022. Interaction matters: a comprehensive analysis and a dataset of hybrid IoT/OT honeypots. In *Proceedings of the 38th Annual Computer Security Applications Conference* (Austin, TX, USA) *(ACSAC '22)*. Association for Computing Machinery, New York, NY, USA, 742–755. https://doi.org/10.1145/3564625.3564645

[53] Shreyas Srinivasa, Jens Myrup Pedersen, and Emmanouil Vasilomanolakis. 2023. Gotta catch'em all: a multistage framework for honeypot fingerprinting. *Digital Threats: Research and Practice* 4, 3 (2023), 1–28.

[54] Chee-Wooi Ten, Chen-Ching Liu, and Govindarasu Manimaran. 2008. Vulnerability assessment of cybersecurity for SCADA systems. *IEEE Transactions on Power Systems* 23, 4 (2008), 1836–1846.

[55] UHH-ISS. 2025. UHH-ISS/honeygrove: A multi-purpose, modular medium-interaction honeypot based on Twisted. https://github.com/UHH-ISS/honeygrove?tab=readme-ov-file [Online; accessed 2025-11-27].

[56] Darshana Upadhyay and Srinivas Sampalli. 2020. SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations. *Computers & Security* 89 (2020), 101666.

[57] Sebastian Walla and Christian Rossow. 2019. MALPITY: Automatic Identification and Exploitation of Tarpit Vulnerabilities in Malware. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, Stockholm, Sweden, 590–605. https://doi.org/10.1109/EuroSP.2019.00049

[58] Gerry Wan, Liz Izhikevich, David Adrian, Katsunari Yoshioka, Ralph Holz, Christian Rossow, and Zakir Durumeric. 2020. On the Origin of Scanning: The Impact of Location on Internet-Wide Scans. In *Proceedings of the ACM Internet Measurement Conference*. ACM, Virtual Event USA, 662–679. https://doi.org/10.1145/3419394.3424214

[59] Wei Xu, Yaodong Tao, and Xin Guan. 2018. The landscape of Industrial Control Systems (ICS) devices on the internet. *2018 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, Cybersa 2018* 1 (2018), 8551422. https://doi.org/10.1109/CyberSA.2018.8551422

[60] Ricardo Yaben, Niels Lundsgaard, Jacob August, and Em-

manouil Vasilomanolakis. 2024. Towards identifying neglected, obsolete, and abandoned IoT and OT devices. *Proceedings of the 8th Network Traffic Measurement and Analysis Conference (TMA Conference 2024)* 1 (2024), 1–10. https://doi.org/10.23919/TMA62044.2024.10558996

[61] Ricardo Yaben and Emmanouil Vasilomanolakis. 2025. Digital ghost ships: abandoned, neglected, and obsolete IoT & OT devices exposed to the Internet. *Authorea Preprints* 1 (2025), 1–12.

[62] Ricardo Yaben and Emmanouil Vasilomanolakis. 2025. Drifting Away: A Cyber-Security Study of Internet-Exposed OPC UA Servers. In *2025 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, Venice, Italy, 195–202. https://doi.org/10.1109/EuroSPW67616.2025.00029

[63] Ricardo Yaben and Emmanouil Vasilomanolakis. 2025. Rolling the DICE: A Device Identification and Classification Engine to detect vulnerable devices facing the Internet. In *Proceedings of the 9th Network Traffic Measurement and Analysis Conference (TMA conference 2025)*. IEEE, Copenhagen, Denmark, 1–4. https://doi.org/10.23919/TMA66427.2025.11097013

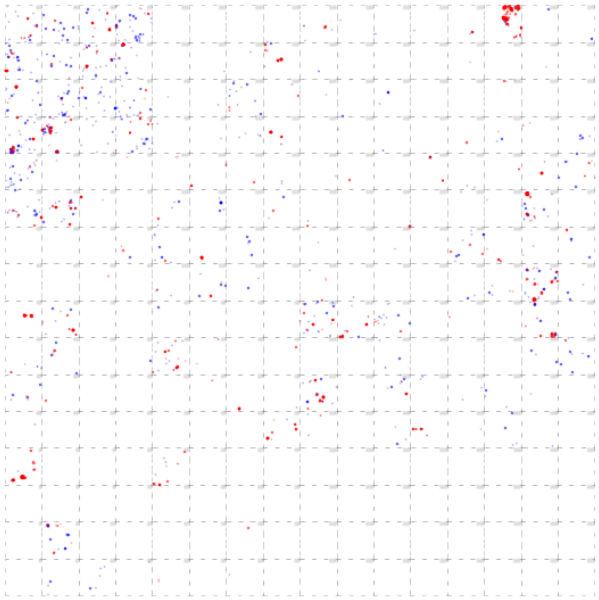
[64] Ricardo Yaben, Emmanouil Vasilomanolakis, and Mathias Anguita. 2025. *Measuring What Matters: Revisiting Internet Exposure of OT Networks*. Technical University of Denmark. https://doi.org/10.5281/zenodo.17977303

[65] Mohammad-Reza Zamiri-Gourabi, Ali Razmjoo Qalaei, and Babak Amin Azad. 2019. Gas what? I can see your GasPots. Studying the fingerprintability of ICS honeypots in the wild. In *Proceedings of the Fifth Annual Industrial Control System Security (ICSS) Workshop* (San Juan, PR, USA) *(ICSS)*. Association for Computing Machinery, New York, NY, USA, 30–37. https://doi.org/10.1145/3372318.3372322

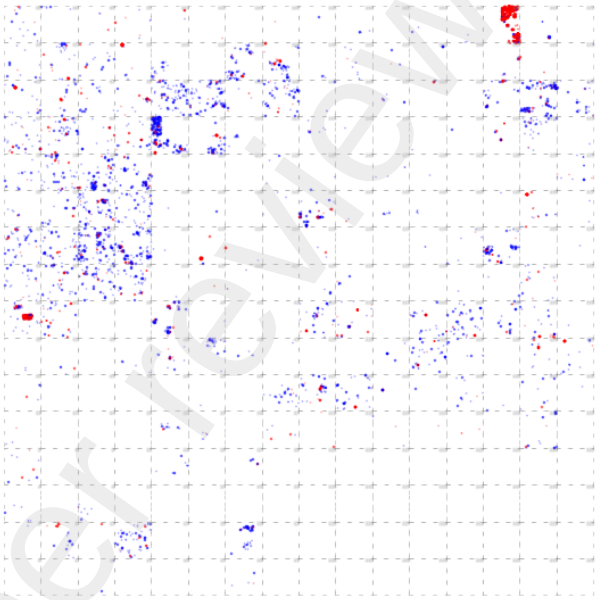
[66] ZMap Project. 2025. *ZGrab2: Application-Layer Scanner*. The ZMap Project. https://github.com/zmap/zgrab2

8. Host distribution

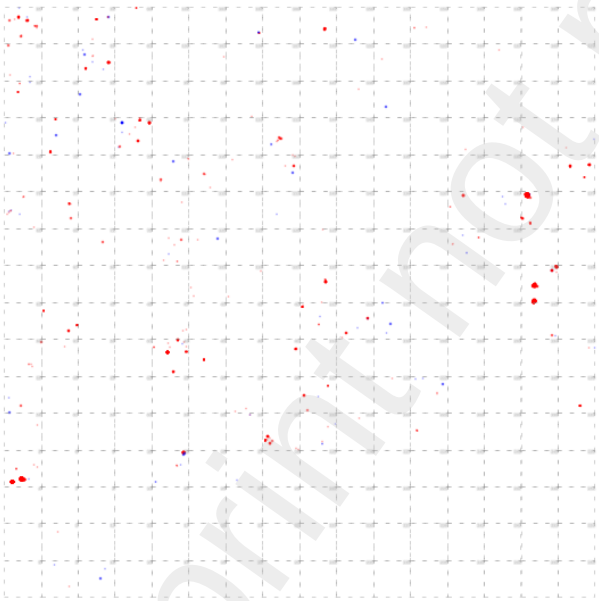
Figure 17 breaks down the host-address-space visualization by protocol. Each subfigure shows the set of identified hosts for that protocol projected into two dimensions using the same IPv4 ordering as in the main text (i.e., nearby points typically correspond to nearby addresses and shared prefixes). Hosts flagged by at least one of our noise classifiers are highlighted in red, while the remaining observations are shown in blue. These maps provide an at-a-glance view of whether a protocol's apparent exposure is dominated by a small number of dense networks (large contiguous clusters) or is more dispersed across the address space.



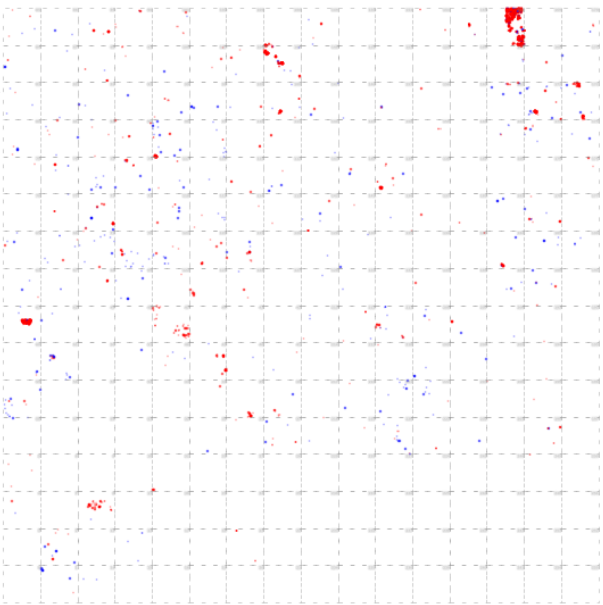
(a) Modbus



(b) Fox



(c) IEC-104



(d) EtherNet/IP

Figure 17: Protocol-specific host distributions in IPv4 address space. Red points indicate hosts flagged as likely noise by our classifiers; blue points indicate hosts without noise flags.