DTU Compute
Department of Applied Mathematics and Computer Science

Identifying systemic cyber-security weaknesses in Internet-facing OT and consumer IoT networks

PhD Thesis

Identifying systemic cyber-security weaknesses in Internet-facing OT and consumer IoT networks

PhD Thesis
January, 2026

By
Ricardo Maria Yaben Lopezosa

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Applied Mathematics and Computer Science, Richard Petersens Plads, Building 322, 2800 Kgs. Lyngby Denmark
https://www.compute.dtu.dk/

Approval

This thesis was prepared at the Department of Applied Mathematics and Computer Science at the Technical University of Denmark in fulfillment of the requirements for acquiring a Ph.D degree.

Ricardo Maria Yaben Lopezosa - 230567

.....

Signature

.....

Date

Abstract

Despite the incessant growth of cyber attacks both in frequency and impact, basic security measures are still widely overlooked. A vast number of Internet-facing devices still lack proper access control, encryption, and maintenance. This thesis explores vulnerability identification methods through Internet measurements as a medium to better understand where systemic cyber-security weaknesses occur and how they can be mitigated.

Our work examines cyber-security issues affecting Internet of Things (IoT) and Operational Technology (OT) networks that originate from negligence, abandonment, and obsolescence. We propose more than ten protocol-specific identification methods to detect misconfigurations, fleet-wide vulnerabilities, and poor security maintenance. Furthermore, we investigate the long-tail effects of these issues and highlight the importance of monitoring OT networks. Our work also includes discussions on the results of multiple ethical disclosure campaigns, which reveal that, despite the efforts, advice is often ignored.

Laying the path forward, we present a new framework to orchestrate complex Internet measurements and monitor OT networks, building on the lessons from previous studies and best practices developed over time for reproducible and comparable measurements. Lastly, we present novel methods to remove *noise* from Internet surveys targeting OT protocols, which we estimate account for a fifth of the results reported in the literature.

Abstrakt

På trods af den konstante stigning i cyberangreb – både i hyppighed og omfang – bliver grundlæggende sikkerhedsforanstaltninger stadig i vid udstrækning overset. Et stort antal internettilsluttede enheder mangler stadig korrekt adgangskontrol, kryptering og vedligeholdelse. Denne afhandling undersøger metoder til identifikation af sårbarheder gennem internetmålinger som middel til bedre at forstå, hvor systemiske cybersikkerhedssvagheder opstår, og hvordan de kan afhjælpes.

Vores arbejde undersøger cybersikkerhedsproblemer, der påvirker IoT- og OT-netværk, og som udspringer af forsømmelse, opgivelse og forældelse. Vi foreslår mere end ti protokolspecifikke identifikationsmetoder til at opdage fejlkonfigurationer, sårbarheder på tværs af enheder og mangelfuld sikkerhedsvedligeholdelse. Derudover undersøger vi de langsigtede konsekvenser af disse problemer og understreger vigtigheden af at overvåge OT-netværk. Vores arbejde inkluderer også diskussioner af resultaterne fra flere etiske disclosure-kampagner, som viser, at rådgivning ofte ignoreres trods indsatsen.

For at bane vejen frem præsenterer vi en ny ramme til at orkestrere komplekse internetmålinger og overvåge OT-netværk, baseret på erfaringer fra tidligere studier og best practices udviklet over tid for reproducerbare og sammenlignelige målinger. Endelig præsenterer vi nye metoder til at fjerne *støj* fra internetundersøgelser, der målretter OT-protokoller, som vi estimerer udgør en femtedel af de resultater, der rapporteres i litteraturen.

Acknowledgements

There are many deserving of this page. Family and friends first, who had supported me unconditionally from the beginning, loved and cared for me, respected my pace and – mostly irrational – decisions. The long periods of silence, my own harsh character, and every other perk of my company. This is one of my proudest moments, and I do not regret my choices, but I wish I had more time with you. To my father, **Javier Yaben**, and my grandfather, **Perfecto Lopezosa**, I miss you. **Mercedes Lopezosa**, my mother, always cheered me, even when I only rambled nonsense to her, speaking of computers, hackers, and conspiracy theories. Always there, bending an ear for my fumbles. **Helena Yaben**, my sister and biggest inspiration, the best humor, the smartest, the tallest, etc. It is hard to believe we are family, but I am so happy and proud of it. **Secundina Gonzalez**, my grandmother, who showered me with love, respect, and village sayings (i.e., wisdom). To all the **Yabens** and the **Lopezosas**, one cannot be luckier than carrying your name. To **Carlos Garcia** and **Esther Sierra**, my closest friends, thanks to them I still preserve a bit of sanity.

To **Emmanouil Vaslomanolakis**, for the many years now of believing in me and entertaining crazy ideas. Thanks to you, researching cyber-security has been a fun and passionate challenge. I am exceptionally lucky for having you in my corner. And to my colleagues, the most interesting, smart, and welcoming bunch. The pressure of doing a Ph.D was always bearable thanks to you. I wish I knew how to thank everybody individually.

Last but not least, I am deeply grateful to **Katharina Widebæk**, to whom I owe the power to turn my life and to accomplish the unimaginable. I am not done yet, and I am still excited to reach for far more. What a wild journey, I cannot thank you enough.

Para mi familia,

Por el valor y el apoyo

-con cariño

Contents

1 Introduction

The consequences of treating cyber-security as an afterthought are self-evident. As incidents continue to surge, our ability to protect networks is being stretched to its limits, driving global threat levels to unprecedented heights. Yet, despite the evermore secure systems, basic practices are frequently neglected (e.g., patching, upgrading, and retiring).

Attackers take advantage of this unique opportunity to target both critical infrastructure and consumer environments with ease. This is evident with the proliferation of large-scale IoT botnet attacks infected via brute-force dictionary attacks and known vulnerabilities (e.g., the original Mirai botnet from 2016 [1]), as well as high-impact incidents targeting OT systems lacking access control (e.g., the Colonial Pipeline ransomware attack in 2021).

The lack of basic security has been a recurrent concern for more than a decade [2], and a recurring theme across the body of work [1], [3], [4], [5]. Nevertheless, the growing complexity of cyber-security and the threat landscape continues to raise the bar. Hardening our systems requires a clear knowledge of these evolving challenges and a continuous refinement of the mitigation techniques we built to prevent their endurance.

At the same time, our tools for scanning the Internet and identifying vulnerabilities have evolved significantly [6], [7], and with them, a developed view of the ethical considerations and best practices surrounding Internet-wide measurement [3], [8], [9], [10]. In fact, a decade of research, launched with early experiments measuring embedded devices lacking authentication [11], has produced more than 600 publications, branching into Internet measurements, deception techniques, socio-cultural studies, and beyond.

This thesis builds on the work of active Internet measurements, aiming at identifying systemic vulnerabilities in controllers and embedded devices common in consumer IoT devices and OT networks, including Industrial Control System (ICS) and SCADA networks, critical infrastructure, and automation systems. By focusing on long-standing issues instead of spending efforts discovering new vulnerabilities, we deliver sound methods to gather evidence of these persistent issues with significantly greater detail. Our research pushes towards maturing the scanning and reporting methodology, adding new foundations to evaluate the security of IoT and OT systems. We investigate the following broader research questions, providing an overarching framing for this thesis.

- **RQ1** What cyber-security weaknesses are the most persistent among IoT and OT networks facing the Internet, and how can these be mitigated?
- **RQ2** How can we improve the reliability of current vulnerability identification methods in Internet measurements, and how can we create reproducible and comparable results?

1.1 Motivation

The motivation for this thesis stems from persistent limitations in the study of Internet-exposed IoT and OT systems. The rapid digitization of physical environments and the widespread deployment of Internet-connected sensors and controllers have significantly expanded the attack surface of modern networks. Often designed with limited cyber-security considerations, these systems have become major contributors to large-scale botnets, privacy violations, and critical infrastructure incidents.

Despite more than a decade of research, many of the underlying cyber-security weaknesses affecting exposed devices remain insufficiently understood, poorly measured, or inconsistently reported. These challenges motivate the need for more accurate, reproducible, and methodologically sound Internet measurements.

The high-level research questions presented in Chapter 1 are further specified in this section through a set of more focused research questions, addressing measurement challenges, methodological constraints, and disclosure practices.

- **RQ1a** How can Internet-exposed IoT and OT devices be identified at scale, and how can cyber-security weaknesses affecting them be detected reliably?
- **RQ1b** To what extent are observed cyber-security weaknesses systemic across IoT and OT environments, and what factors contribute to their persistence?
- **RQ1c** How can measurement findings be responsibly communicated to system maintainers, and what role do ethical disclosure practices play in reducing exposure?
- **RQ2a** What limitations affect current Internet measurement methodologies targeting IoT and OT networks, and how can these limitations be mitigated?
- **RQ2b** How can large-scale measurements be made more accurate and reliable, and how can we improve reproducibility?

The initial focus of this work lies in IoT and OT exposure, where devices are often deployed at scale, dynamically addressed, and managed with limited operational oversight. These characteristics complicate long-term measurement and attribution efforts, yet they also reveal widespread vulnerabilities resulting from misconfiguration, neglect, and obsolescence (RQ1a, RQ1b).

Moreover, Internet-wide measurements inevitably interact with unknown and potentially critical systems. Ethical considerations, responsible disclosure, and communication with system operators are therefore integral components of meaningful measurement practice. This thesis examines how ethical disclosure campaigns can influence exposure mitigation efforts (RQ1c).

Experiments targeting both types of networks evidenced that many of the weaknesses observed in IoT environments are equally prevalent in OT networks. Contrary to expectations of stricter control and standardized security practices, OT deployments frequently expose services lacking basic cyber-security, such as Modbus, BACnet, and Ethernet Industrial Protocol (Ethernet/IP). This observation motivates a deeper investigation into OT exposure and measurement challenges (RQ2a).

Measuring IoT and OT networks introduces distinct methodological challenges. While longitudinal monitoring can uncover systemic issues in relatively static OT environments, similar approaches are less effective for IoT networks due to high churn and heterogeneity. These differences require tailored measurement strategies and highlight the limitations of one-size-fits-all Internet-wide scanning approaches (RQ2a, RQ2b).

Beyond technical challenges, the field faces broader limitations, including limited ground truth, poor reproducibility, scarce public datasets, and insufficient reporting of measurement instrumentation. Although the literature contains extensive guidance on Internet measurement practices, many studies remain difficult to replicate and unfeasible to compare. Addressing these issues is essential to advance the reliability of empirical findings (RQ2b).

Building on existing measurement methodologies, this thesis proposes new probing techniques and analytical frameworks to improve accuracy, account for noise introduced by deception systems and other Internet artifacts, and better align measurement observations with real-world security concerns. In doing so, it seeks to contribute toward a more robust and reflective practice of Internet measurement.

1.2 Publications

This thesis draws upon the following co-authored publications produced during the Ph.D., which focus on uncovering systemic vulnerabilities in OT and IoT networks.

[A] R. Yaben et al., "Towards identifying neglected, obsolete, and abandoned iot and ot devices," eng, *Proceedings of the 8th Network Traffic Measurement and Analysis Conference (TMA Conference 2024)*, vol. 1, pp. 1–10, 2024. DOI: 10.23919/TMA62044.2024.10558996

[B] R. Yaben and E. Vasilomanolakis, "Digital ghost ships: Abandoned, neglected, and obsolete iot & ot devices exposed to the internet," *Authorea Preprints*, vol. 1, pp. 1–12, 2025 [Preprint]

[C] R. Yaben and E. Vasilomanolakis, "Drifting away: A cyber-security study of internet-exposed opc ua servers," in *2025 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2025, pp. 195–202. DOI: 10.1109/EuroSPW67616.2025.00029

[D] R. Yaben and E. Vasilomanolakis, "Rolling the dice: A device identification and classification engine to detect vulnerable devices facing the internet," in *2025 9th Network Traffic Measurement and Analysis Conference (TMA)*, 2025, pp. 1–4. DOI: 10.23919/TMA66427.2025.11097013

[E] R. Yaben et al., "Measuring what matters: Revisiting internet exposure of ot networks," Journal preprint, available at SSRN, 2025. DOI: 10.2139/ssrn.5974783. [Online]. Available: https://ssrn.com/abstract=5974783 [Preprint]

While included in the thesis as an appendix, the following paper was also written during the Ph.D. and inspired our most recent publications to consider the impact of cyber-deception in OT measurements and its applications when combined with active scanning methods to detect exposed vulnerable OT devices and ongoing attack patterns.

[1] Mongardini et al., "A Systematic Meta-Survey of Cyber Deception: Unified Taxonomy and Research Directions," 2025 [Preprint]

1.3 Outline and Contributions

This dissertation aims to develop a deeper understanding of widespread cyber-security weaknesses affecting IoT and OT networks facing the Internet. By choosing active measurements as our method to survey the Internet, this dissertation contributes to the body of work studying the Internet's security landscape. Our studies enhance existing identification techniques with new probes to detect insecure devices across more than ten widely used protocols in IoT and OT. This thesis offers practical guidance to address such weaknesses and proposes alternative approaches for communicating findings to device owners.

From a practical standpoint, this thesis introduces new methods to analyze systemic cyber-security weaknesses and addresses key limitations in the field through a framework

that enables reproducible and comparable complex measurements. Our framework summarizes best practices and considerations targeting sensitive networks, including noise filtering capabilities and novel approaches to detect signs of negligence, obsolescence, and abandonment.

Chapter 2 establishes the necessary foundations on Internet measurements and vulnerability assessment in IoT and OT environments. The core contributions of this dissertation are presented in Chapters 3 to 7, which correspond to the publications produced during this Ph.D. These chapters are organized into two thematic parts, each addressing a distinct subset of the research sub-questions introduced in Section 1.1.

Part I: Identifying Systemic Weaknesses in Internet-Facing Systems

Part I (Chapters 3 to 5) addresses research sub-questions derived from **RQ1**, focusing on the identification and characterization of persistent cyber-security weaknesses in Internet-facing IoT and OT systems. Through large-scale Internet measurements covering widely deployed protocols, these studies analyze security management failures, software obsolescence, and device abandonment, providing empirical evidence of their prevalence and long-term persistence. In addition, it explores mitigation strategies and alternative disclosure practices aimed at reducing exposure and improving communication with affected stakeholders.

- **Paper** A (Chapter 3) Addresses **RQ1a and RQ1b**, introducing our first set of methods to detect vulnerable devices facing the Internet. This paper develops eight probes to detect access control and certificate management issues in IoT and OT Internet-facing systems. The data collected from these devices reveal several systemic issues associated with mismanagement and obsolescence. The results of this study suggest that these issues are not unique to one particular sector. Lastly, we conduct a responsible disclosure campaign to notify over a hundred maintainers and operators to address **RQ1c**; however, we received merely five responses.
- **Paper** B (Chapter 4) This paper expands on Paper A with a longitudinal study of the evolution of these issues to further investigate the research questions **RQ1a and RQ1b**, and briefly evaluate the effects of our previous disclosure campaigns and address **RQ1c**. This study highlights the persistence of cyber-security weaknesses over time, showing that 25% of IoT and up to 50% OT vulnerable devices have not received any maintenance, despite efforts to inform system operators.
- **Paper** C (Chapter 5) This paper examines the security posture of Internet-exposed OPC UA servers. It primarily addresses research questions **RQ1a** and **RQ1b**, with a specific emphasis on OT environments. The study demonstrates that security-aware protocol design alone is insufficient when deployment and maintenance practices are flawed. Our findings reveal that most exposed OPC UA services are affected by misconfigurations, including unsuitable security policy combinations and the continued use of deprecated authentication and encryption mechanisms, effectively negating built-in protections.

Part II: Advancing Measurement Methodology and Mitigation Strategies

Part II (Chapters 6 to 7) aims towards addressing **RQ2** by examining the methodological

limitations of current Internet measurements targeting OT networks. This part introduces new approaches for monitoring exposure over time, evaluates sources of measurement bias and noise, and proposes a unified framework to improve reproducibility and comparability.

- **Paper** D (Chapter 6) This paper introduces DICE, a device identification and classification engine that draws on the literature for conducting effective Internet measurements to conduct complex scans and produce comparable and reproducible results. This work is a direct attempt to address **RQ2a and RQ2b**, providing a framework to build upon that researchers can use to evaluate methods and share instrumentation calibration details.
- **Paper** E (Chapter 7) This study analyzes the impact of noise on active Internet measurements targeting OT networks. It addresses research questions **RQ2a and RQ2b** by focusing on improving measurement accuracy and reliability. We introduce noise-aware probing techniques and detection rules to identify deception systems, network telescopes, and other Internet artifacts that inflate large-scale measurement results. Despite filtering these sources of noise, our findings show that exposed OT devices continue to exhibit widespread vulnerabilities, including outdated software, obsolete deployments, and reliance on legacy protocols lacking basic security features.

Finally, Chapter 8 synthesizes the findings across all studies, reflecting on their implications for Internet measurement research. The chapter concludes by outlining open challenges and directions for future work.

2 Internet measurements: Methods and Security Applications

Internet measurements originate from calibration experiments to understand and optimize paths over the network by characterizing the end-to-end dynamics of the Internet, studying routing, network quality properties, bandwidth, and packet loss [17], [18]. Measurements were initially thought to analyze traffic as it expanded from a few thousand interconnected networks to millions of computers competing for address space. The pace at which the Internet has evolved means that it is difficult to untangle its complexities, modeling our understanding of the Internet landscape faster than we can digest it. Part of the challenge, as John et al. [19] explains, is that the Internet is not governed by a centralized entity controlling the expansion of its infrastructure. In practice, each Autonomous System (AS) – of the many Internet management entities – borrows address space and decides on the policies it implements, such as cross-border and internal routing (i.e., the paths interconnecting addresses within the AS, and how they connect with others), prefix advertisement and sub-letting, robustness of the network, etc. While overly simplified, this explanation gives a glimpse at the complexities arising from such a wildly heterogeneous ecosystem.

The term *measurements* is rather precise: the objective is to describe the functioning of the Internet through observation. Similar to others, Internet measurements require measuring points [18], frequently referred to as *vantage points*, units of measurement or metrics (e.g., number of OT exposed devices), a methodology to collect observations, instrumentation, etc. Over time, Internet measurements have branched to study different aspects of the Internet dynamics. In fact, cyber-security constitutes a significant margin of the work. Safaei Pour et al. [20] provide an excellent high-level taxonomy for Internet measurements from a cyber-security perspective, describing the core concepts and summarizing the most common techniques to study the different aspects of Internet. In addition, the authors reflect on the general issues discussed throughout the literature that continue to hold back the field from moving forward. Their work highlights how small method variations produce significant differences, adding further complexity to one of the major cornerstones of Internet measurements: ground truth, reproducibility, and comparisons.

Many like us have focused on the security and privacy implications of applications exposed to the open, while others specialize in mitigating network abuse (e.g., BGP and DNS poisoning, amplification, and denial-of-service attacks), monitoring outages, the impact of cyber-warfare, socio-cultural aspects of cyber-security, censorship, etc. Broadly, much of the quantitative work studying cyber-security can be grouped into three categories: i) Internet infrastructure, ii) network security, and iii) protocol and application security. The first includes core components routing and transporting Internet traffic, such as Internet exchange switches, DNS servers, and other indispensable technologies building the Internet's backbone. Then, studies focused on Network security frequently examine topics such as topology properties and their performance, and alternative networks, overlays, private communities, and virtual networks. Lastly, the work on protocol and application security often investigates exposure and attack vectors weakening the communication or compromising the endpoints directly. Examples include encryption coverage and systems used over the Internet evaluating whether Internet-facing services encrypt their communications, and if so, how secure are those.

While the main goal of the measurements is to gain quantitative insights through empirical experiments, studies in this area often include mixed-methods combining qualitative data on observed behavioral patterns with quantitative results by some form of traffic analysis. One of the benefits is the opportunity to open discussions on topics such as the ethics behind Internet measurements [21]. The literature describes two main methods to gather quantitative results: i) active measurements, and ii) passive measurements.

This thesis is only concerned with protocol and application security for the particular case of IoT and OT networks facing the Internet, and supplements the literature with privacy implications and societal behavioral aspects contributing to the proliferation of widespread cyber-security weaknesses. Therefore, this chapter builds on the concept of Internet measurements and its cyber-security applications to mitigate weaknesses in IoT and OT networks. Specifically, this chapter dives into combined active and passive measurements through Internet-wide scans and honeypot deployments to detect vulnerable and infected devices, describing techniques used in remote device profiling as means to evaluate their security. Section 2.1 introduces the main differences between active versus passive measurements, and cases where both methods have been combined. Section 2.2 examines active measurements and introduces a taxonomy design scanning campaigns. Section 2.3 covers the evaluation metrics for active measurements. Section 2.4 dives into the sub-field of Internet-wide scans and the nuances particular to this approach in comparison with scoped scans. This section includes a discussion of the gaps and limitations of the field, and the contributions of this thesis to their mitigation.

2.1 Methods

Traditionally, quantitative methods for Internet measurements are often grouped into active and passive methods, with the main difference between these two groups being the role of the observer in the measurement; either active and attempting to establish communication channels with remote networks, or passive and mainly receiving these attempts or quantifying traffic passing through the observation point (i.e., vantage point). These two groups are not exhaustive or exclusive; while active and passive measurements can be considered the most representative forms, hybrid variations and other combinations exist, but often one of the two methods takes the primary role of the study. A representation of this classification and major studies can be seen in Figure 2.1. An example used throughout the literature is semi-active (reactive) measurements, a form mainly seen as part of deception deployment studies where the goal is to understand the origin of the attacks (i.e., reverse identification) rather than studying the attacks themselves, a particularly useful technique widely used to uncover botnet infection vectors. In our view and in contrast with the taxonomy provided by Safaei Pour et al. [20], this classification also includes studies relying on Cyber Threat Intelligence (CTI) and crowd-sourced data, since these too used active, passive or hybrid methods to collect observations. This section examines these groups, describing the methods, highlighting the most notorious examples found in the literature, and linking the method to its main applications through this thesis.

Active Measurements. In active measurements, vantage points have an active role, attempting to initiate communications with other hosts over the Internet. These often include some form of scoped or full-range Internet scanning using specially crafted probes. In this context, *probe* is the name given to the process of communicating with another host – other fields use the term *probe* as a synonym for vantage point. This term is often used to describe protocol-specific requests; however, the terminology is vaguely defined and may require context to understand the concept. Therefore, we define the act of sending a probe as a vantage point attempting to communicate with another host in

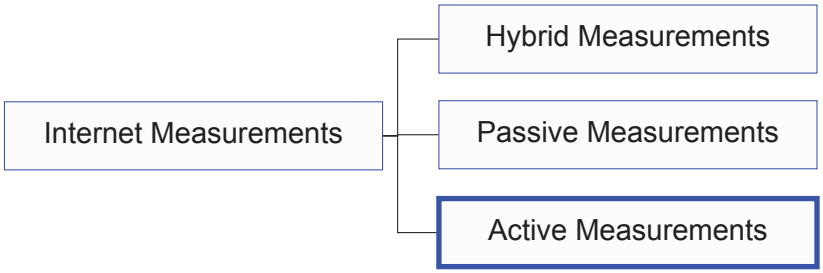


Figure 2.1: Categorization of Internet measurements. Active measurements highlighted as the main method used in this thesis.

a particular way. Commonly, the goal of *probing* a host is to isolate as many variables as possible to produce a deterministic outcome (i.e., *does the host behave in a particular way when interrogated with a set of questions?*). A common example of scanning and probing used throughout this thesis is known as *banner-grabbing*, which is the minimal form of interaction between a vantage point and a host to identify exposed services, often terminating after the initial handshake.

Active measurements strive to minimize the side-effects of the Internet's chaotic nature, aiming towards negligible packet loss, overcoming churn to mitigate double-counting issues, and reducing the impact of external factors on the observations, such as variance produced by the timing of the measurement, or dataset false-positive pollution from noise, such as honeypots and network telescopes.

Passive Measurements. Passive measurements leverage traffic in transit and incoming unsolicited requests to study Internet dynamics. Vantage points in this context are also referred as *sensors*, and, while the term is widely associated with deception, it is also used in other fields and proven exceptionally useful to study Internet radiation and network performance properties. However, cyber-deception remains the main user of passive methods, often consisting of simple decoy systems, in cases occupying whole prefixes (e.g., network telescopes), or single hosts through fully flagged honeypot systems emulating real services. Cyber-deception has taken many forms and covered a wide range of use-cases. These

Hybrid Measurements. Hybrid measurements combine passive and active approaches to measure the Internet. This type of study often weighs towards one of the two approaches. However, vantage points are rarely combined. Instead, hybrid measurements will use dedicated machines for each type of measurement, and combine findings afterwards. Most hybrid studies have been used to examine attack patterns and their volume, giving an overview of the ongoing risks; then, experiments would include a scanning campaign to measure exposure.

A special case of this type of measurement is where all communications happen between vantage points with the same owner. For example, network measurements studying throughput optimization problems typically involve two or more vantage points communicating in a closed circuit and capturing different metrics, such as packet loss, bandwidth, traffic loads, delays, etc.

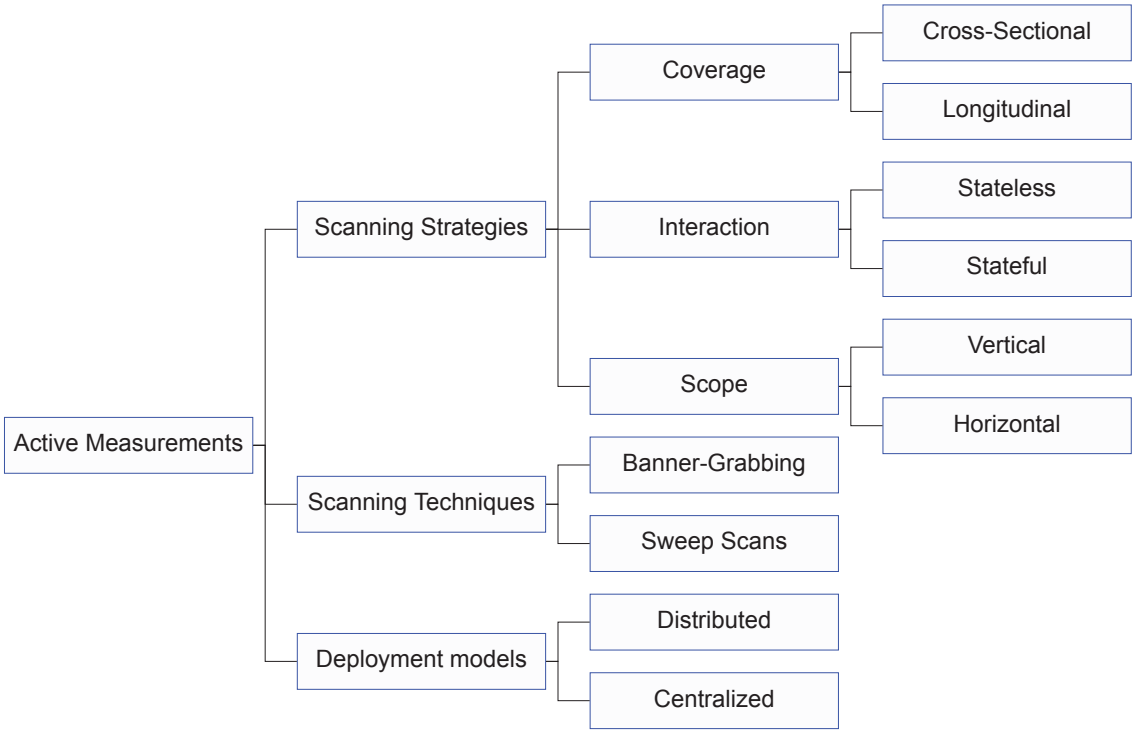


Figure 2.2: Active Internet measurements taxonomy of scanning strategies, techniques, and deployment models. Active measurements combine multiple nodes from all levels.

2.2 Active Measurements in Depth

This section introduces a classification of techniques, strategies, and deployment models to conduct Internet measurements as shown in Figure 2.2. This taxonomy describes the process of designing and carrying out active measurements for different objectives, giving an overview of the elements under consideration, their limitations, and how maximizing for one metric disputes with the rest. It is important to mention that the elements of this taxonomy are not exclusive, and active measurements will combine most leaves. The scope of the experiments carried out during this thesis combines stateful and stateless probes in multi-staged scans, and while horizontal in nature (Internet-wide scans), our measurements included some verticality with multiple protocols across the transport and application layers (e.g., stateless TCP and UDP sweeps with stateful banner-grabbing strategies for OT protocols).

2.2.1 Scanning strategies

Internet scanning has undergone significant evolution over the years. Access to new tooling and strategies has opened new possibilities to more nuanced and complex scans. We distinguish three dimensions that define current scanning strategies: scope, interaction, and dynamics.

Scope. One of the main considerations when conducting scans is the scope, which defines the portion of the Internet being covered and determines the extent of data required from that population.

• *Vertical.* Traditionally, scanning campaigns have primarily focused on less than 10% of the Internet at the time (e.g., surrounding addresses, specific AS, or physical locations) [22], a dated but common strategy that allows for accurate snapshots of a narrow portion of the total population. By focusing on less of the available address space, scanning

campaigns can evaluate more aspects of their targets, which often translates into more ports being probed or studying adjacent characteristics. This is what we know as *vertical* scanning, where granularity is often favored over other metrics.

• *Horizontal.* On the other hand, *horizontal* scans have a larger scope, but their results are far coarser. Horizontal scans cover larger spaces with less detail, targeting fewer ports, and digesting fewer details in general. Over the years, horizontal scans have become more accessible thanks to developments in probes and tooling, to the point where simpler Internet-wide scans can be done within a few hours [23]. In turn, nowadays horizontal scans can offer a greater level of detail without sacrificing other aspects. Nevertheless, choosing the right level of height over depth highly depends on the goal of the scan. This thesis mainly conducts Internet-wide scans targeting a limited set of protocols on their default ports.

Interaction. The rise of horizontal scanning is closely linked to improvements in probing mechanisms used to interrogate large numbers of hosts. Probing constitutes one of the most significant bottlenecks in scan design, as it governs the level of interaction required per host. This consideration underpins the distinction between *stateless* and *stateful* probing approaches.

• *Stateless.* Stateless probes are optimized to take full advantage of the available bandwidth and maximize the number of hosts being interrogated. To achieve this, scanners speak directly with the vantage point's network card, sending raw packets towards their targets without blocking the socket or tracking the communication. This method is less precise and reliable than regular stateful probes. Stateless scanners mitigate these issues by implementing a short tracking window, retransmitting the probe multiple times per host, and adding identifiers to the outgoing packets. While this scanning technique requires careful calibration and the tradeoffs are considerable, it is effective and widely used to test for livingness (i.e., whether the host responds).

• *Stateful.* Stateful probes are equivalent to normal communications and are often described in terms of the level of interaction or intrusion. Interaction can be measured as the combination of metrics such as the number of packets required for the communication, the average time required to complete the probing process, conformance with the protocol standard, or properties describing the intention of the probe (e.g., banner-grabbing versus vulnerability testing or resource enumeration).

Coverage. Besides interaction and scope, active Internet measurements are mainly reported as either cross-sectional or longitudinal studies. We find exceptions to this rule, however. The work of Durumeric et al. [7] and trends among CTI services merge both terms in a third category, *continuous*, that optimizes for accuracy instead of coverage. They aim to give an accurate representation of the Internet at present, instead of examining the state of the Internet at a given time or showing its evolution over a period.

• *Cross-Sectional.* Cross-sectional scans are the most widely used strategy found in the literature. This strategy provides single-point snapshots of the Internet at a given time. Snapshots help measure the population of a service and estimate the size of the Internet, detecting outages, and identifying systemic issues from observing the Internet as a whole (or parts of it). However, the duration of the scan and other dynamics are important factors shaping its accuracy. Among other biases, slower scans risk probing the same host twice due to Internet churn, and the timing of the scan may jeopardize the experiment's ability to observe its true population.

Table 2.1: Layers of the OSI model with protocol examples

Layer	Name	Protocols
7	Application	Modbus, OPC UA, MQTT
6	Presentation	TLS
5	Session	SQL, RPC
4	Transport	TCP, UDP
3	Network	ICMP, TCP/IP
2	Data Link	Ethernet, Bluetooth, WiFi, NFC
1	Physical	

• *Longitudinal.* Longitudinal studies offer a contiguous view of the Internet over time, emphasizing evolution trends over fixed-point observation details. These studies are complex. Comparing Internet measurements is a delicate and meticulous task that requires substantially more resources than cross-sectional ones. However, longitudinal scans are considered more complete, since they have the potential to mitigate much of the variability in Internet measurements. For example, scanning multiple times over a short period can capture otherwise exposed services that failed to respond during one or more iterations due to network degradation, packet loss, or availability issues.

2.2.2 Scanning Techniques

Remote probing techniques can be mapped to the upper layers of the OSI model (cf. Table 2.1), namely the network ($L3$), transport ($L4$), and application layers (mainly $L7$). Techniques vary depending on the protocols under study and the objective of the measurement. Some of the most common $L3$ protocols used to understand Internet dynamics include ICMP and IPX. These protocols have been widely used to understand topological aspects of the Internet, such as routing and pathing, and counting networks facing the Internet. For this thesis, we focus on TCP/IP protocols to test for responsive hosts and evaluate service exposure.

Bou-Harb et al. [10] collected the most comprehensive list of scanning techniques to date for protocols in the $L3$ and $L4$ layers. Active scanning techniques in these layers have not changed significantly; the most notorious advancements have been focused on the instrumentation and protocols in layers $L5$ to $L7$. The community has contributed with evaluation methods for these probes, showing that $L3$ and $L4$ measurements do not contain enough information to evaluate Internet exposure or evaluate application security issues. These limitations gave way to a further categorization of Internet scanning techniques, splitting measurements into multi-staged approaches combining lower-layer protocols to detect occupied and exposed addresses (sweep scans), with upper-layer protocols to measure service exposure at a deeper level (banner-grabbing).

Sweep Scans. Lower-layer protocol scans are jointly known as *sweep scans*, where TCP SYN-ACK and UDP scans do most of the work at the first stages. The probes used in sweep scans are designed for coverage over accuracy, indicating whether hosts are *alive* (exposed and responding). Other analytics have emerged relying on this data. Shamsi et al. [24] proposed new Operating System (OS) fingerprinting methods leveraging single-packet probing. Cordeiro and Vasilomanolakis [25] recently developed a method to detect hosting providers among OT deployments to identify honeypots and odd systems.

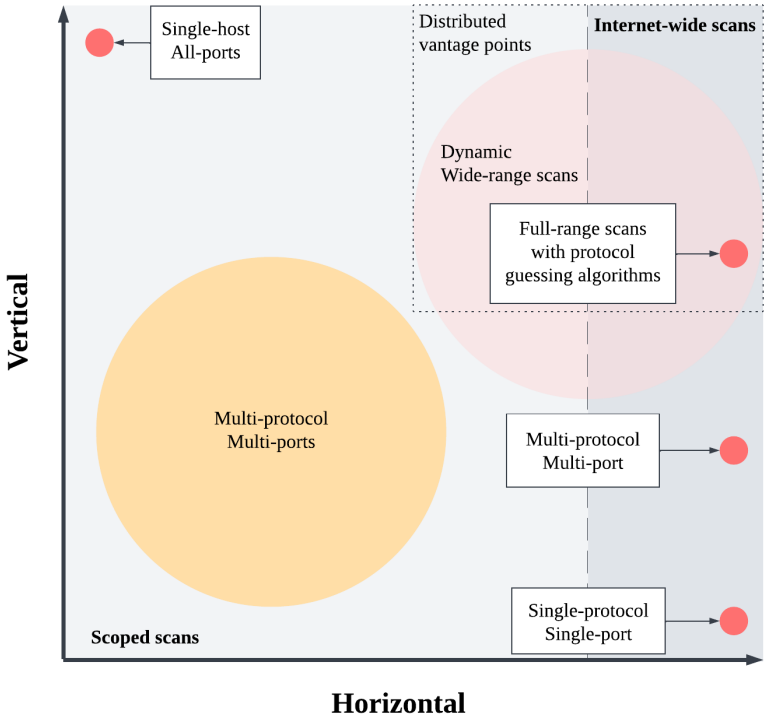


Figure 2.3: Schema of active measurements deployment models for different objectives. This thesis covers the lower-right quadrant, conducting Internet-wide scans targeting single or multiple protocols in their default ports for regular and encrypted communications. Complexity grows exponentially over the diagonal from the bottom-left quadrant to the top-right corner. The top-right quadrant currently requires distributed deployment models and dynamic measurements to achieve completeness.

Banner-Grabbing Scans. Upper-layer protocol scans are widely known as banner-grabbing scans; however, this terminology is vague and does not specify the intent or intrusion level of the probe, with many of which sequence multiple requests before closing the communication, as opposed to traditional banner-grabs, where the connection terminates after initial service handshakes. One of the most notorious examples using this method to examine ICS exposure is the work of Mirian et al. [8].

2.2.3 Deployment Models

Vantage points are an integral part of the measurement. The reliability of a measurement is often related to the number and location of vantage points used in a study. Linking to the experiment's coverage, the visibility of a vantage point is directly affected by its physical location and Internet routing options. Vantage points distributed around the globe will inevitably observe the Internet with different perspectives, and their own capabilities will affect it. However, larger and more complex measurements require a combination of diversity mixed with advanced scanning strategies. Figure 2.3 shows a matrix of how active scanning campaigns require different deployment models. The requirements for the deployment model increase with the complexity of the scan. In short, we distinguish between centralized and distributed deployment models.

Wan et al. [9] recently explained how the location of the vantage point determines the scope of the experiment, limiting the capacity of observing the Internet to only endpoints within reach. The choice of one versus many, geographically distributed, or simply increasing diversity can have a significant impact on the precision of the measurement.

Durumeric et al. [7] explains how vantage point seeding with marginal overlaps improves the visibility and increases accuracy. In addition, their work achieves greater coverage than most measurements by applying dynamism to their methodology, where the Internet is monitored instead of captured in snapshots. For this, the authors schedule scans at multiple levels, where ports are split into batches with different relevance. Lastly, authors apply protocol-guessing algorithms to identify services running on random ports. In addition, the authors show how other properties can alter our view and the differences created from ignoring their effects. Their work aims to give a continuous and accurate representation of exposed services, demonstrating how the frequency of their measurements paints a different picture of the Internet compared to others. By capitalizing on metrics that require large resources to gather, such as the breadth of their scans, the authors offer a unique view that serves as a baseline for many other studies with a narrower scope and detailed analysis.

Centralized. Centralized deployments use a single vantage point as the source for the measurements. This deployment model is widely used across the literature, requiring fewer resources than its counterpart. However, while centralized deployments introduce several non-negligible biases (cf. Section 2.3), expanding to distributed approaches increases the complexity of the measurement significantly. Instead, many authors opt for dynamic and temporal features in their scans to improve coverage and completeness (e.g., repeated scans).

Distributed. Distributed deployments scan the Internet from multiple locations. The scanning methodology varies, with parallel deployments being more common than sequential executions. Overlaps between scanners may be introduced to support measurement evaluation. Full overlap allows results from different locations to be compared, providing insight into vantage-point quality and reliability. At the opposite end, non-overlapping deployments focus on maximizing coverage and reducing scan time. While centralized deployments can apply these strategies by increasing the number of scanning machines in the vantage point (e.g., François et al. [26] used one vantage point, parallelizing operations across eight virtual machines), the gains differ from those obtained through distributed deployments.

2.3 Evaluation Metrics for Active Measurements

Lacking true ground truth, active measurements seek alternative metrics for their evaluation. As Heidemann et al. [27] explains, the performance of active measurements can be inferred from their collection methods (including bias and representation), accuracy, completeness, and reproducibility of the study. While dated, their methodology already considered the limitations and nuances of Internet surveys versus censuses (i.e., Internet sampling and population studies). The measurements included in this thesis are considered Internet surveys under this terminology. Figure 2.4 shows a representation of the metrics used to evaluate measurements, building on the lessons from the literature [7], [9], [27]. Similar to other fields, studies in this area strive for an accurate and complete view of the Internet (completeness and accuracy), and reproducibility. In addition, we identify three threats of bias: vantage point, probes used, and instrumentation and calibration.

Reproducibility. Lacking a complete ground truth does not imply that measurements should not strive for transparency. Internet measurements' reproducibility is often reported as part of the methods used to collect observations, sharing the details on the experimental setup, data sources, and other relevant nuances particular to the study. While

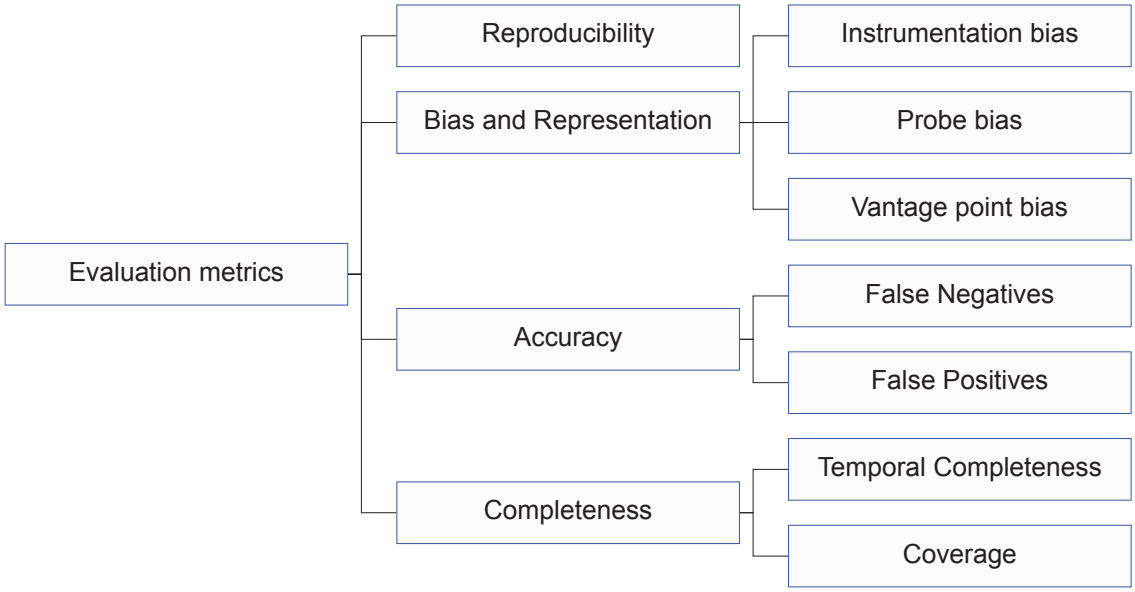


Figure 2.4: Evaluation metrics for active Internet measurements.

comparisons between methods and datasets are often thought to be unfeasible, there have been several examples measuring the same events with consistent results across studies (e.g., the Mirai botnet [1], [28], [29]). However, this process is not straightforward, and there is no consensus on which metrics should be used to improve reproducibility. It is the case that datasets produced from studies examining cyber-security vulnerabilities in the wild often contain sensitive information, making authors reluctant to share them directly. Moreover, the tooling is rarely shared: tools are heavily tailored for the job, proprietary, or merely described. Durumeric et al. [23] highlights this limitation based on a decade of experience with the widespread adoption of the ZMap ecosystem. The authors call into question the community's contributions to reproducibility, noting that despite over 300 publications relying on these tools, probes are rarely shared or actively maintained. To mitigate this issue, the probes crafted throughout this thesis are widely available, and we are working towards merging them with the official tool branches.

Completeness. Due to the dynamic nature of the Internet, the best we can achieve through active measurements is an imperfect ground truth; in other words, a degree of temporal completeness through the lens of the observable Internet across time.

• *Coverage.* Coverage is often reported as the percentage of the Internet space under observation. While most measurements conduct Internet-wide experiments, it is common to use blocklists that reduce the coverage by a significant margin. Abuse reports, agencies that opt-out, sensible ranges, and reserved address spaces may crop the measurement's visibility of the Internet. On the other hand, determining which networks actively blocked measurement traffic is challenging, and identifying the root causes may not be feasible in most cases, as it may stem from selective censorship, network abuse triggers, firewalling, network degradation, etc.

• *Temporal Completeness.* Temporal completeness captures the extent to which an active Internet measurement observes the target population across time, rather than at a single instant. As mentioned, Internet dynamics make capturing snapshots of the Internet challenging, missing a significant portion of the target population due to volatile aspects, such as churn, low availability, dynamic blocking, etc. To mitigate these effects, Internet

measurements can increase the frequency of their experiments to measure differences across multiple scans.

Accuracy. Accuracy is understood as the fraction of measurement results that correctly correspond to the targeted measurement. Often, Internet measurements strike a balance between accuracy and coverage, as the accuracy of an experiment inevitably decreases when optimizing for coverage, and vice versa. This effect occurs due to the imperfections particular to Internet measurements, as snapshotting the Internet frequently across all ports is currently unfeasible. In addition, authors have demonstrated that, at least for a few widely used protocols, less than 10% of the exposed services live in their default ports [30], [31]. Therefore, measurements require careful analysis to test for false positives and false negatives, as the opposite also occurs in particular cases where deception and odd systems may pollute datasets.

• *False positives.* Internet measurements inherently produce large volumes of false positives, which we denote as *noise*. For measurements studying exposure, noise has been shown to account for over 90% of responses [30], largely due to interference from filtering mechanisms, deception systems, telescopes, and other services operating on the targeted ports. As Mladenov et al. [32] and Srinivasa et al. [33] report, deception systems degrade the accuracy of a measurement for about 5-20% of the observations depending on the protocols under study. In terms of volume, prior work has shown that deception systems are more likely to mimic either OT networks or expose an arbitrary number of services. Additionally, measurements that merely test for liveness often find their findings misrepresenting the landscape of active hosts, not accounting for multi-homed and aliased networks, telescopes, and an array of filtering systems. False positives lead to large overestimations, often by orders of magnitude.

• *False negatives.* If false positives reduce the accuracy of a measurement, a double failure is not to adjust the sensitivity of noise detection systems, leading to mistakenly taking true positives for noise. This is particularly relevant for vulnerability identification systems, where the expectation bias of the classification system may deem services as unlikely due to the sheer amount of vulnerabilities found in the host, misinterpreting actual vulnerable devices for honeypots, for example.

Bias and Representation. The decisions taken during the design of an active Internet measurement have a direct impact on the data representation. As discussed, the choice of vantage point, scanning technique, and strategy plays a crucial role in observing the Internet.

• *Vantage point bias.* Active measurements heavily depend on the vantage points' capacity to observe the visible Internet. The decision between one versus many, geographically distributed, and the deployment method (e.g., parallel, sequential, scheduled, continuous, with overlaps, etc.) provides a unique perspective on the Internet, revealing significant differences [9]. Another aspect often neglected is whether vantage points have been previously used for similar ends. Because Internet scans spray networks with unsolicited traffic, vantage points are regularly reported to IP reputation services. Hosts relying on those services can manage firewall rules accordingly to treat traffic originating from reported addresses differently. It is now understood that measurements can vary significantly due to this phenomenon [7], [9], [27]. For once, the impact of adaptive firewalls on the measurement increases over the period of the experiment and lingers afterwards, but diminishes over time as the vantage point ceases its activity.

• *Instrumentation bias.* As active measurement practices converge on a small set of widely used tools (e.g., ZMap, Masscan, Nmap, ZGrab) [23], [34], it becomes increasingly important to understand their differences and to calibrate measurements accordingly. Scans must balance throughput, packet loss, and measurement windows, among other factors. Although scanning tools have similar goals, differences in their probing mechanisms significantly affect both accuracy and coverage. Internet-wide scanners such as ZMap and Masscan prioritize efficiency over depth, relying primarily on stateless sweeps to maximize resource utilization, whereas tools like Nmap emphasize accuracy through stateful, vertical probing.

• *Probe bias.* Probing accuracy is one of the most relevant factors in active measurements. Whether we are testing for liveness or service exposure, parameters such as artificial delays between requests, connection timeouts, and intrusion level affect how accuracy degrades. As a result, even minor differences between otherwise similar probes can lead to substantial variation in measurement outcomes.

2.4 Identifying Security Weaknesses and Internet Exposure Through Large-Scale Active Measurements

This section describes applications of Internet-wide scanning methods to detect vulnerable devices facing the Internet. As shown in Figure 2.3, large-scale active measurements are a special case of Internet measurements where the scope is the full-range of the addressable and observable Internet from one or more vantage points.

2.4.1 Vulnerability Identification

The large majority of large-scale measurements aiming to uncover security weaknesses target selected protocols to study issues associated with a concrete group of devices or services. The vertical scope of these types of measurements is directly limited by the resources available. And, even for large scanning infrastructures, full coverage with high accuracy is still unfeasible, only reaching acceptable levels for a handful of ports and protocols [7]. Authors attempt to mitigate this issue by crowd-sourcing their measurements with the results from CTI services.

Targeting specific populations or groups of similar protocols is advantageous and limits the scope of the measurement without modifying its breadth to offer significantly more granulated results. Granulation and accuracy are different; granulation is depth as accuracy is correctness. Examples in the literature include the work of [11], one of the first Internet surveys to record the spread of vulnerable embedded devices, targeting exposed telnet services protected with default credentials (`root:root`) or fully open. The work of Saidi et al. [35] explores IoT deployments in the wild and security implications. Zhao et al. [36] uses only CTI data to uncover vulnerable IoT devices facing the Internet over a period of six months, identifying over a million devices with security issues ranging from lacking access control to N-day vulnerabilities. Other studies focus on ICS protocols, such as [8]. In addition, there are notorious examples of studies covering complex protocols individually. The work of Dahlmanns et al. [37] examines the exposure of OPC UA servers and their security measures in place. OPC UA is an alternative protocol that supersedes legacy OT standards; however, most of the security features offered by this standard are optional or difficult to configure. Some of the protocol's encryption methods used to establish secure connections have long deprecated. Unfortunately, OT environments consistently fail to adopt these changes. In a similar vein, Srinivasa et al. [38] studies IoT exposure to uncover misconfigurations and other security issues associated with poor security management and the role of society's relationship with cybersecurity. The authors employ

hybrid methods to first actively detect exposure, and then detect compromised systems through passive measurements.

The methodology and instrumentation used across the literature are very similar. The body of work relies on multi-staged scans, ensembling sweep scans over banner-grabbing phases using the ZMap ecosystem for active probing, and crowd-sourcing information to complete datasets with geolocation, AS data, and widely available details, such as OS and TLS handshake fingerprints.

Except for a few, most authors analyze device and service versioning information that reveals details on the device's usage and known vulnerabilities. For example, Dahlmanns et al. [39] studied the ubiquity of TLS adoption in exposed IoT and OT devices, showing that fewer than 7% of the deployments offer TLS-secured connections. Gasser et al. [40] analyzed BACnet exposure to demonstrate that these automation systems can be used as network amplifiers, accepting what seem innocuous unauthenticated UDP requests to retrieve device information. These issues speak to the harsh reality that, before we shift our focus to more advanced security vulnerabilities, we need to address the proliferation of basic security weaknesses and remediate contributing behaviors. However, this is not an *either-or* conclusion; the work on actively exploited issues drives the field (e.g., Mekdad et al. [41]), showing current attack patterns and prioritizing from the long list of security issues that need addressing.

2.4.2 Ethical and operational considerations

Internet-wide vulnerability assessments conducted through active scanning raise important ethical and operational considerations. Unlike controlled experiments, these measurements interact with large numbers of unknown and heterogeneous systems without informed consent (i.e., through public exposure), potentially affecting devices, networks, and services beyond the researchers' control. Responsible measurement practices require careful attention to probe design, impact mitigation, and transparency methods to allow participants to opt out.

Impact of Internet-wide Scanning. Internet-wide scanning campaigns generate substantial volumes of unsolicited traffic and often rely on probes that may be perceived as intrusive. As a result, authors are expected to explicitly describe the measures taken to mitigate the impact of their studies on target networks and devices. This is particularly important in large-scale measurements, where even benign probes can create unintended operational effects.

Risk to Critical and Sensitive Systems. Because Internet-wide scans interact with unknown and heterogeneous devices, including systems that may support critical infrastructure or life-dependent electronics, experiments must be designed with caution. Ethical considerations extend beyond individual hosts to the potential systemic effects of scanning activity on the Internet as a whole. Researchers should account for these risks in both experimental design and reporting, and clearly document the steps taken to minimize potential harm.

Probe Design and Interaction Constraints. To reduce ethical and operational risks, probes should be designed to prioritize protocol conformity, efficiency, and minimal interaction. Without explicit consent, invasive, malformed, or state-modifying probes are not acceptable.

Transparency and Opt-Out Mechanisms. Responsible Internet-wide measurements should provide clear mechanisms for transparency and communication with affected parties. This includes offering ways for network operators or device owners to identify scanning traffic and contact authors. Providing opt-out mechanisms and promptly honoring requests to exclude specific networks or hosts are widely recognized best practices in ethical Internet measurements.

Ethical Review. Finally, large-scale measurement studies should involve appropriate oversight and conformity with institutional or community ethical standards. While formal ethics committees are not always available for Internet measurement research, alternative forms of review are commonly used. This thesis includes multiple ethical statements documenting consultations with regional Cybersecurity Emergency Response Teams (CERTs) and institutional advisors, serving as a substitute for formal ethical review processes.

Several authors have attempted to establish guidelines and considerations for Internet measurements over the years [42], [43], [44], including requirements for ethical disclosures, thoughtful design choices, etc. However, the field has not reached the maturity level of others, where studies and experiments are evaluated by an external and often centralized committee that guarantees the ethical risks are acceptable.

2.4.3 Limitations and Open Challenges

The persistent nature of fundamental cyber-security problems highlights the need to critically examine the limitations of existing approaches. Prior to this thesis, the body of work already offered substantial guidance on addressing known gaps and proposed directions for advancing the field. For example, long-standing issues affecting ICS and, more broadly, OT networks have remained unresolved for over a decade. Despite the availability of straightforward mitigation measures, similar observations continue to reappear, suggesting that current approaches still fall short. This thesis contributes to the identification of root causes underlying persistent cyber-security weaknesses in IoT and OT networks. Beyond this, it advances the field toward more reproducible and comparable measurement practices and helps bridge the gap to more advanced large-scale Internet measurements. The proposed methods emphasize improved accuracy through probes that provide finer granularity, noise-aware device identification, and systemic vulnerability detection.

Accuracy at Scale. A central challenge identified in the literature is achieving accurate measurement results with high reliability and precision. Over time, the efforts of the community have produced new tools and methodologies to address the limitations of large-scale scans. In the past decade, Internet-wide scans transitioned from week-long single-port or single-protocol probing with the lowest accuracy, to highly reliable multi-port and protocol scans within a single day. Recent techniques promise improved accuracy, with probes capable of producing highly reliable results at smaller scales. However, the problem of achieving high accuracy-coverage ratios remains largely unsolved, a challenge that other fields of Internet measurements have mitigated through collaborative methods.

This thesis introduces new probes for more than ten widely used IoT and OT protocols, extending beyond traditional banner-grabbing by testing authentication, authorization, and encryption. Although these probes increase traffic volume, they provide higher reliability and accuracy, enabling the detection of systemic trends and the identification of false positives. As discussed in Section 2.3, deception and other forms of noise have become prevalent in large-scale measurements and are no longer negligible. While the population-level

behavior is more relevant than detailed observations of individual hosts, measurements must strive for more precise and noise-aware estimations.

Limited Reproducibility and Comparable Methods. Results and instrumentation calibration details are rarely shared among researchers. Authors spend significant efforts replicating the probes and experimental setups used in previous work. However, slight differences in the methodology produce large variances between measurements. As a result, comparing results and methods becomes unfeasible; discrepancies between results are confusing, and benchmarking is not possible in any axis. Erkek and Irmak [45] echo these issues in an analysis of commonly used CTI tools and their available data. This issue is also reported by Durumeric et al. [7]. From independent studies, Li et al. [46] reports observing more than 20K Internet-facing Modbus services and Wu et al. [47] nearly 28K, while Xu et al. [48] reports on hundreds of thousands, and Guo et al. [49] 76K. In addition, probe and tool efficiency have only been measured (and improved) in counted occasions [6], [30], [50], [51].

While these challenges remain open, our contributions to the literature include open-sourced probes and a new tool for the community to unify active measurements and conduct complex scans. This tool aims to simplify the compatibility between instrumentation and reporting on scan meta-information often discarded (e.g., throughput, packet loss, hit-rate, duration, coverage, etc.). Our goal is to improve transparency around measurements and facilitate reproducibility and comparisons between datasets.

Identifying Systemic Weaknesses in Internet-Facing Systems

Overview

This part addresses **RQ1** by examining persistent cyber-security weaknesses in exposed IoT and OT systems through a series of Internet-wide measurements. The chapters included in this part focus on identifying systemic issues arising from device neglect, obsolescence, and abandonment across a diverse set of widely deployed protocols. Together, these studies provide empirical evidence that long-standing weaknesses persist despite increased awareness and improved protocol designs, and they illustrate how large-scale measurements can be leveraged to reveal population-wide security trends.

The following chapters are presented with their original abstracts for reference and contextual discussions linking each contribution to the broader research questions of this thesis.

3 Towards identifying neglected, obsolete, and abandoned iot and ot devices

Context and Contributions

This chapter establishes the empirical foundation of **RQ1** by demonstrating the scale and prevalence of neglected and mismanaged IoT and OT deployments on the public Internet (**RQ1a and RQ1b**). It introduces a cross-protocol measurement approach that identifies indicators of abandonment and misconfiguration using widely adopted scanning tools. The findings motivate the need for longitudinal analysis and refined terminology, which are addressed in subsequent chapters. Lastly, this chapter shows that despite efforts, responsible disclosure campaigns often go unnoticed (**RQ1c**).

RQ	Contribution
RQ1a	Identification of persistent weaknesses
RQ1b	Characterization of exposed IoT and OT populations
RQ1c	Exposure mitigation through responsible disclosure campaigns

Related publication

R. Yaben et al., "Towards identifying neglected, obsolete, and abandoned iot and ot devices," eng, *Proceedings of the 8th Network Traffic Measurement and Analysis Conference (TMA Conference 2024)*, vol. 1, pp. 1–10, 2024. DOI: 10.23919/TMA62044.2024.10558996

Original Abstract

The rapid adoption of Internet of Things (IoT) and Operational Technology (OT) devices to control systems remotely has introduced significant cyber-security challenges. Attackers have compromised millions of such devices over the years, exploiting their lack of management and weak cyber-security. In this paper, we examine cyber-security issues of neglected, obsolete, and abandoned IoT and OT devices exposed to the Internet. The core of our work focuses on identifying these devices using common scanning tools to find indicators of vulnerabilities and misconfigurations. Moreover, we present an analysis of our Internet-wide scans during a period of two weeks targeting security issues in 8 IoT and OT protocols: MQTT, CoAP, XMPP, Modbus, OPC UA, RTPS, DNP3 and BACnet. We observed over 1 million addresses exposing one or more of these services, of which 675,896 appear vulnerable or misconfigured. Lastly, we examine the IP reputation of the vulnerable devices and show that 7,424 were reported at least once.

3.1 Introduction

The emergence of the IoT and OT has permeated most aspects of our lives. From smart home devices to medical instrumentation and critical infrastructure, all sectors of society are rapidly becoming reliant on these new technologies. While their benefits are undeniable, their rushed adoption introduced new risks and security challenges, inviting adversaries to take control of those lacking security. Recent large-scale IoT attacks such

as the Mirai botnet [1], powered by close to a million compromised devices, have evidenced the challenges society faces to secure their devices, posing a major threat to their environment and other systems. Researchers continue to work to mitigate this issue, with various studies focused on the landscape of IoT and OT devices exposed to the Internet [38], [52], [53], proposing mitigation strategies to reduce the number of exposed and vulnerable devices [54], and investigating society's cyber-security posture towards their devices [5]. However, there is a lack of research dedicated to identifying devices that appear forgotten in our networks, misconfigured (e.g., lack access control, encryption, or leak sensitive information), or deprecated (e.g., decommissioned or unpatched). That is, Internet-connected devices neglected of cyber-security, obsolete (yet in use), or abandoned altogether.

This paper focuses on the security issues associated with neglected, obsolete, and abandoned IoT and OT devices exposed to the Internet through the lens of Internet-wide scans targeting 8 protocols commonly found in general-purpose IoT and OT devices and ICSs: MQTT, CoAP, XMPP, Modbus, OPC UA, RTPS, DNP3 and BACnet. For this, we used the ZMap ecosystem to scan the IPv4 for two weeks in December 2023, supporting our dataset with further information from Shodan [55] and Censys [56] to fingerprint devices, and data from the NIST vulnerability database. Lastly, we include an IP reputation analysis of the vulnerable addresses using open blocklists and Greynoise [57]. Our contributions are listed as follows.

- We extend multiple ZGrab probes and develop two new ones (i.e., RTPS and OPC UA) to conduct a series of Internet-wide scans targeting 8 protocols commonly used in IoT and OT devices (one scan per protocol).
- We identify 1,019,887 systems exposed to the Internet, out of which 675,896 contain neglected, obsolete, or abandoned devices. The majority are general-purpose devices exposing CoAP, MQTT, and XMPP vulnerable services. Moreover, we show that most services used in ICS are insecure. We informed affected companies in our region (Denmark) and included here some insights on the responses we received.
- Using IP reputation services, we show that 7,424 devices are reported as suspicious or malicious, some of which appear infected with Mirai variants and other malware families.

The remainder of this paper is structured as follows. Section 3.2 begins with an overview of the relevant literature for identifying vulnerable IoT and OT devices over the Internet. In Section 3.3, we briefly introduce the scope of our work and our approach to scanning the Internet, as well as the ethical considerations and our self-imposed scanning limitations. Then, in Section 3.4 we analyze our scanning results to identify neglected, obsolete and abandoned devices. Lastly, Section 3.5 summarizes our findings, touching on the IP reputation of the potentially vulnerable devices we discovered, and the responses to our vulnerability disclosure. Section 3.6 concludes this paper.

3.2 Related Work

Numerous studies conduct Internet-wide scans to investigate vulnerabilities in IoT and OT devices [58], [59], [60]. The methods for scanning the Internet are well-established [10] and most authors use off-the-shelf common tools such as those from the ZMap ecosystem [50] or Masscan [61], alongside meta-scanners (e.g., Shodan and Censys), and IP reputation services (e.g., Virustotal [62] and GreyNoise). Authors extend or develop new probes for these tools to cover different use-cases; however, their scanning

choices largely depend on the scope of their work (e.g., vantage points, number of scans, and period) [9].

A significant part of the literature focuses on ICSs exposed to the Internet [37], [47], [63], given that many of those systems operate in critical environments and lack security features. In [4], the authors conducted multiple full IPv4 scans targeting nine ICSs-specific protocols with custom ZMap probes. They report finding over 60,000 exposed systems, some of which belong to critical infrastructure organizations, airports, and government facilities. Lastly, they supplement their work with an IP reputation analysis using a Network telescope to identify malicious traffic proceeding from these addresses. In another study, [5] introduced a 5-year longitudinal analysis using Shodan and Censys to fingerprint devices exposing either of 6 ICSs protocols. The authors offer a holistic perspective on this issue including human aspects in their study, such as owner security behaviors, and economic motivations driving cybercriminals. More recently, [39] studied the use of TLS in 10 ICS protocols, showing that less than 7% of nearly a million exposed devices secure their communications.

In addition, there has been a notable effort to identify vulnerable IoT and OT devices exposed to the Internet [35]. In [52], the authors scanned for specific IoT devices over the Internet to identify vulnerabilities and other issues associated with this technology. Moreover, [64] scanned the IPv6 space instead, targeting six common IoT protocols. They identified 36,400 IoT devices, highlighting security concerns such as non-trusted and expired TLS certificates. Lastly, the work of [38] is the closest to our study, focusing on misconfigured IoT devices exposing one of five widespread protocols. They also include a reputation analysis of the misconfigured devices they found using a network telescope and multiple honeypots, an analysis of the attack trends on each of the protocols they support, and a brief discussion on the attacker behavioral patterns they observed. The major difference with [38] is in the aim of our work, while [38] centers on current attack trends on IoT devices using honeypots and network telescopes, the cornerstone of our study is to identify vulnerable IoT and OT devices from their response behavior. Our study is inspired by these approaches to identifying vulnerable devices beyond matching Common Vulnerabilities and Exposures (CVEs), including other factors such as lack of authentication and encryption and disclosing internal resources.

In summary, most authors have focused on introducing new methods to fingerprint IoT and OT devices and identifying their vulnerabilities. The state of the literature includes many valuable lessons about the risks of exposing these technologies to the Internet and how to secure them. However, few authors draw on the security behaviors leading to such vulnerabilities, failing to represent the bigger picture: these devices are poorly maintained. To address this gap, we shift our attention from common vulnerabilities to how these devices are handled in practice, investigating the state of obsolete, neglected, and abandoned devices that remain connected to the Internet.

3.3 Methodology

This section covers our approach to scanning the Internet, including ethical considerations and technical limitations, as well as our decision pipeline for identifying vulnerable devices. The factors defining whether a vulnerable device shows signs of abandonment, obsolescence, or being neglected of cyber-security varies depending on the protocol and use case. Generally, we define neglected devices as those lacking security hygiene (e.g., reusing certificates) or appropriate maintenance, such as misconfigured (e.g., weak or no authentication, or using default values meant to be changed) or unpatched devices. Abandoned devices suffer from the long-lasting effect of being neglected, such as using

deprecated configurations and software versions, using expired certificates, or being reported as malicious. Lastly, obsolete devices are characterized as either lacking the security features required for Internet communications, or using decommissioned software or hardware. This includes legacy systems that remain active despite not receiving support.

3.3.1 Scanning the Internet

Drawing on the latest trends in the literature [9], [26], [50], [65], we divide our scans into two phases: first we carry a sweep scan using ZMap, followed by banner-grabbing scans using ZGrab. Sweep scans are remarkably fast, consisting of a single packet per port to identify responsive services; whereas banner-grabbing scans complete full connections to collect banner information and handshake details [10]. This method reduces the duration of the scans and the amount of traffic we generate toward each address.

We scanned the Internet for two weeks in December 2023 from a local vantage point, excluding certain addresses from those who had previously requested to opt-out of similar studies [66]. Moreover, we hosted a website at the same address containing details about our study (e.g., targeted protocols and ports), and opt-out and abuse contact information. Lastly, we included a signature in most probes to help system owners identify our traffic, indicating the address of our website and the name of our institution. The signature could be found in header fields such as the user agent, or in the payload for those protocols that accept content in the body of the request.

3.3.2 Ethical considerations and limitations

Conducting Internet-wide scans produces a substantial load of traffic on target networks [10], [67]. Therefore, we implement several technical measures to mitigate the impact of our scan and our level of intrusion. For example, we use the randomization features from ZMap to ensure a maximum distance between each probe targeting the same block of addresses [50], including a minimum of 15 seconds between probes to the same address.

Furthermore, our probes only establish anonymous communications with their targets, using empty credentials or a self-signed certificate (when authentication is required). In addition, we follow a similar approach to other authors [38], limiting our connections to 30 seconds and setting limits to the amount of data we gather (cf. Section 3.4 for the individual implementations).

Lastly, we conduct a notification campaign for the owners of vulnerable devices in our region (Denmark). We limited the notification/disclosure campaign to our country only as this required significant manual work. In this context, future work would benefit from automated notification of misconfigured devices. We discuss the general aspects of their feedback in Section 3.5.2.

3.3.3 Data processing

To focus on relevant data, we fine-tuned our scanner to exclude specific responses. First, our scanner drops echoed responses with identical information to our requests. Echo responses are common in low-interaction honeypots. While it could be interesting to apply our methodology to identify vulnerable honeypots as well (i.e., honeypots with unintended vulnerabilities), we will not study honeypots in this paper. Moreover, we exclude duplicate responses from the same address and service; we noticed this behavior while testing our methodology on 1% of the Internet, most likely caused by servers not receiving RST packets to close the connection, Internet churn, packet loss and drops, and other common issues associated to Internet scanning as documented in [6], [9], [50]. Adding to this, we are aware of the behavior of some controllers exposing RTPS services that will not stop transmitting data for long periods [68].

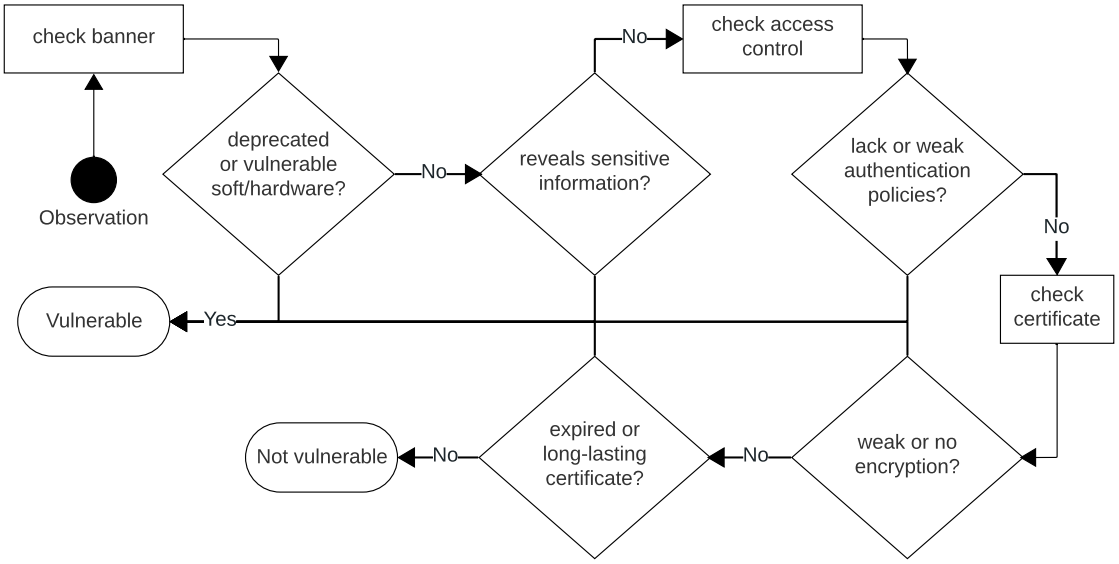


Figure 3.1: Decision pipeline to identify neglected, abandoned, and obsolete devices.

Regarding our post-processing pipeline, we enrich our results with data from Shodan and Censys, querying these services for the addresses in our dataset instead of merging their observations with ours. As other authors pointed out [5], these services do not provide sufficiently accurate snapshots of the IPv4. Therefore, we decided to use this data for minor parts of our analysis, such as geo-locating devices and filtering honeypots (e.g., addresses responding to all targeted protocols and self-disclosing honeypots). In addition, we use the NIST database for vulnerabilities [69] to find known vulnerabilities in the products we encounter. Lastly, we use open blocklists and GreyNoise to analyze the IP reputation of vulnerable addresses.

To process observations and come to our conclusions, we follow a decision pipeline as illustrated in Figure 3.1. This pipeline mainly focuses on three aspects: *banner information, authentication policies,* and *encryption*. We analyze each of these three aspects separately and combine our findings to determine whether a device can be considered vulnerable.

3.4 Results

This section provides a protocol-by-protocol analysis of the responses gathered during our Internet-wide scan, including brief descriptions of the protocols as well as our probes. First, we cover general-purpose IoT protocols, i.e., MQTT, CoAP, and XMPP, followed by OT protocols primarily used in SCADA systems, i.e., Modbus, OPC UA, RTPS, DNP3, and BACnet. We present our findings in terms of the vulnerabilities associated with each protocol to identify neglected, obsolete, and abandoned devices. An overall summary of our results is listed in Table 3.1.

3.4.1 MQTT

This is a publish-subscribe protocol commonly used in IoT environments. We extended the ZGrab2 probe to follow up on successful connections without authentication in place, first subscribing to the built-in system topic "`$SYS/#`", and then to the rest of the topics using the wildcard "#". Our probe maintains the connection for up to 90 seconds and collects names from at most 50 topics. When either condition is fulfilled, we immediately disconnect from the broker and discard any further traffic from the same broker.

Table 3.1: Summary of exposed and vulnerable services per protocol Probe: ○ default, ◐ modified, ● new

Transport	**Protocol**	**Port(s)**	**Total**	**Vulnerable**	Greynoise	Probe
TCP	MQTT	1883	491,794	424,961	2,986	◐
UDP	CoAP	5683	301,933	150,927	3,085	◐
TCP	XMPP	5222, 5269	186,949	62,092	729	◐
TCP	Modbus	502	28,787	28,787	318	◐
TCP	OPC UA	4840	1,797	1,210	30	●
UDP	RTPS	7400-7402	708	708	6	●
TCP	DNP3	20000	668	668	9	○
UDP	BACnet	57808	7,251	7,251	333	○
Total:			1,019,887	675,896	8,204	

Out of the 491,794 brokers we found, 424,961 (86.41%) accepted our probe without providing any authentication, allowing us to join sensible topics with the state of the device; only 62,655 brokers rejected our probe with non-authorized errors. Topic values provide further insights into, e.g., software, version, and activity of the broker. These values allowed us to distinguish 424,034 Mosquitto brokers, 40 HBMQTT/aMQTT and 739 other unidentified brokers.

When we analyzed the Mosquitto broker versions, we found 404,471 Mosquitto brokers running on vulnerable versions, with 11 brokers using $v1.0 - beta$. In addition, we cross-referenced the broker version with known security vulnerabilities to assess the risks of using deprecated or abandoned software. Figure 3.2 shows the mirrored distribution of the broker and version (upper side), with the vulnerabilities affecting those versions colored to represent severity (bottom). Beyond insufficient access control, we find that all of the exposed Mosquitto brokers had multiple severe software vulnerabilities, ranging from overflows - stopping the broker - to complete overtake. Regarding HBMQTT, this project had its last release in 2020, and aMQTT (a continuation of HBMQTT from different authors) in 2022. Compared to the wide range of Mosquitto versions in our dataset, we could only see HBMQTT instances using the latest version available. In addition, we could not find other vulnerabilities besides lacking access control; however, we assume this is due to its limited adoption.

Takeaway: Allowing anonymous clients to subscribe to internal topics is a non-negligible risk that may lead to further attacks (e.g., depleting resources, and privilege escalation). That said, none of the brokers revealed any non-internal topics, indicating that accessing other topics would require additional authentication. Further analyzing TLS certificates could help determine the broker's purpose and activity, providing a better understanding of the device's state. Overall, 424,961 brokers had insufficient access control, out of which 404,471 brokers used deprecated and vulnerable Mosquitto versions, which we consider a sign of abandonment and negligence toward the security of the device.

3.4.2 CoAP

CoAP enables constrained devices to communicate over the Internet using a structure similar to HTTP. Our probe sends an anonymous request to the home path of the server, which typically contains a banner with software and resource information. This probe tar-

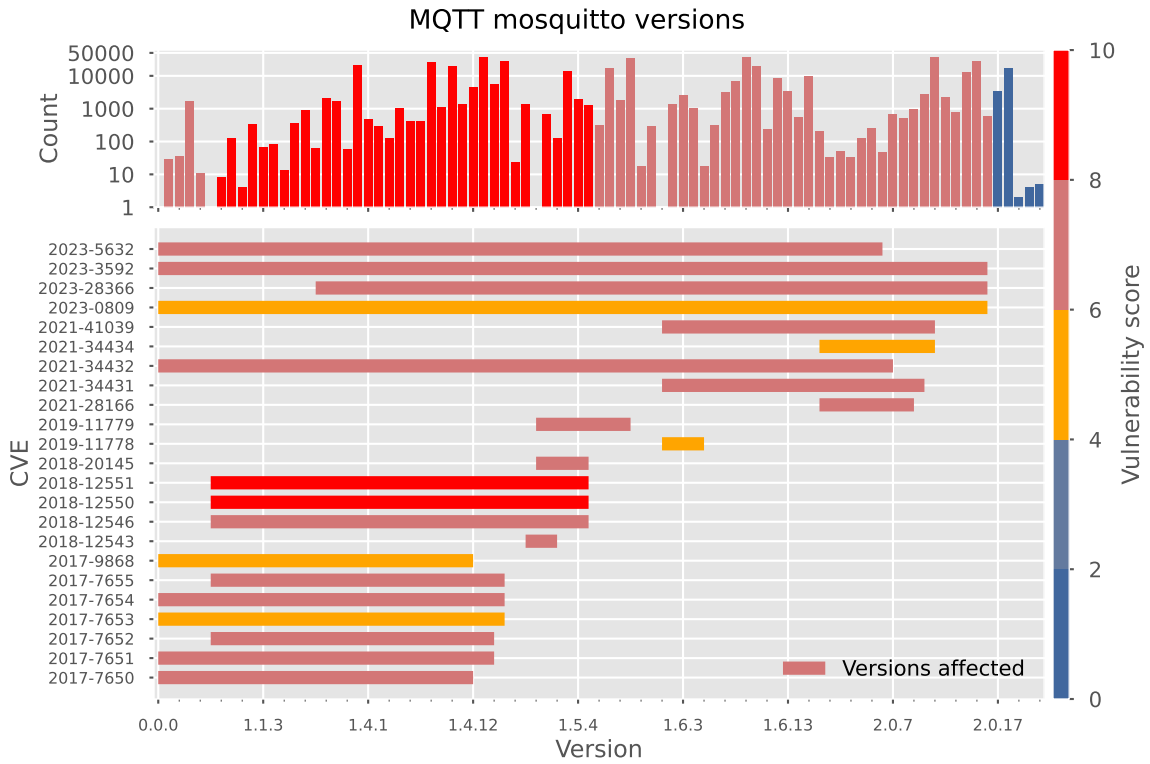


Figure 3.2: Distribution of MQTT Mosquitto versions found in our dataset (top), and their vulnerabilities (bottom) colored by severity score.

gets CoAP servers with basic security features disabled (e.g., TLS) to narrow our results to devices with clear indicators of being neglected or abandoned.

Our scan produced 301,933 CoAP results, with 151,042 disclosing their server implementations, while the rest responded with various errors. We noticed that only a few servers responded with authorization errors, suggesting that our probe could be improved to close the gap between successful responses and the total results. Regardless, we observed a principal group within the positive responses of 106,753 servers using `libcoap`, 86,688 of them running on the oldest version available dating from 2013, followed by 20,066 servers using a version from 2019. In addition, we find 44,095 Californium CoAP servers, where 39,259 use versions between $v2.0.0-M3$ (from 2017) and $v2.1.0$ (2020), plus 2,174 servers using older versions. The remainder of the servers returned values that we could not link to any known implementation. Lastly, 150,927 servers exposed their device type and other resources under the `/.well-known/core` path, from where we could identify 60,411 deprecated QLink and 89,205 NDM routers, and 1,311 Efento NB-IoT wireless sensors. Combining these findings we conclude that there is a significant number of obsolete and neglected routers exposed to the Internet waiting to be overtaken or abused in amplification attacks.

Takeaway: Allowing anonymous clients from the Internet to communicate with CoAP servers comes with severe security and privacy implications. These clients can access sensitive information regarding implementation details and further device characteristics. Therefore, we classify the 150,929 CoAP servers allowing anonymous connections and leaking device information as neglected or abandoned, with a large margin running on obsolete or deprecated versions.

3.4.3 XMPP

Previously known as *Jabber*, XMPP is the open Standard for messaging applications based on XML. Today, this protocol offers an alternative to MQTT and CoAP in constrained devices such as printers and sensors. Our probe initiates a stream communication channel with XMPP services acting either as a client or server depending on the targeted port. As a result, the probe prompts a banner response without the need for authentication.

We received 186,949 XMPP banners, with 127,718 responding servers, and 59,231 clients. In the case of servers, the XMPP banner indicates when authentication and encryption are required; however, we observed similar behavior from a few clients. Furthermore, 8,344 servers included their authentication challenge mechanisms in the banner. We show the top 10 most frequent challenge combinations in Figure 3.3 plus a runner-up in the 11th position with the most insecure combination of authentication methods. As depicted in the figure, the most common authentication methods are either plain-text challenges or deprecated ones (i.e., `DIGEST-MD5` and `CRAM-MD5`), with an interesting group in the fourth position including `SCRAM-SHA-1` as an option, and another in the sixth position with mainly insecure combinations, both disregarding TLS for the most part. In XMPP, servers supporting plain-text authentication are expected to encrypt the communication using TLS, and hashing the credentials [70]. However, a significant part of the observations do not require TLS. Moreover, we find 1,689 servers showing further signs of misconfiguration, such as using stream compression, which XMPP obsoleted recently due to a chosen-plaintext vulnerability.

From the total, only 14,748 enforced TLS as a requirement for communications. Because our probe cannot capture this information during the initial handshake, we use Shodan to query and fetch certificates from our list of addresses. Shodan's results are limited when compared to the extent of our dataset, yielding 2,768 certificates. Nevertheless, even such a subset of the certificates reveals the poor maintenance of XMPP servers. Figure 3.4 shows the validity of the unique TLS certificates, with the count of reused on top and expired certificates colored in red. We observed a large number of expired certificates, lasting longer than 10 years, and reused. Most of the reused certificates we found belong to contact and call center equipment, such as VoIP phones. For example, the two most frequent certificates in our dataset belong to 582 and 148 VoIP phones from the same manufacturer, with the latter certificate expired since 2016. The high load of devices from the same manufacturer suggests that most come preconfigured by default. In addition, continuing to use devices with expired certificates indicates a lack of security management. This duality highlights a common issue among IoT and OT devices, where manufacturers try to simplify the security configuration process but consumers fail to maintain it.

Takeaway: Pairing the lack of access control or encryption with insecure configurations and expired or long-lasting TLS certificates sums up to a total of 62,092 neglected and obsolete servers. Servers with expired or long-lasting certificates are no longer secure. In addition, servers with weak authentication options are prone to downgrade attacks, which indicates insufficient access control. Using vulnerable and default configurations puts servers at risk, even when other security measures are in place.

3.4.4 Modbus

Modbus is a master-slave protocol for industrial automation and control systems. This protocol lacks built-in security features, allowing adversaries to eavesdrop on connections in plain text, read (and potentially write) device information, flood them with traffic, and leverage compromised devices in further attacks [4], [71]. Our scanner uses the default ZGrab2 probe to send Read Device Identification requests, which triggers a response

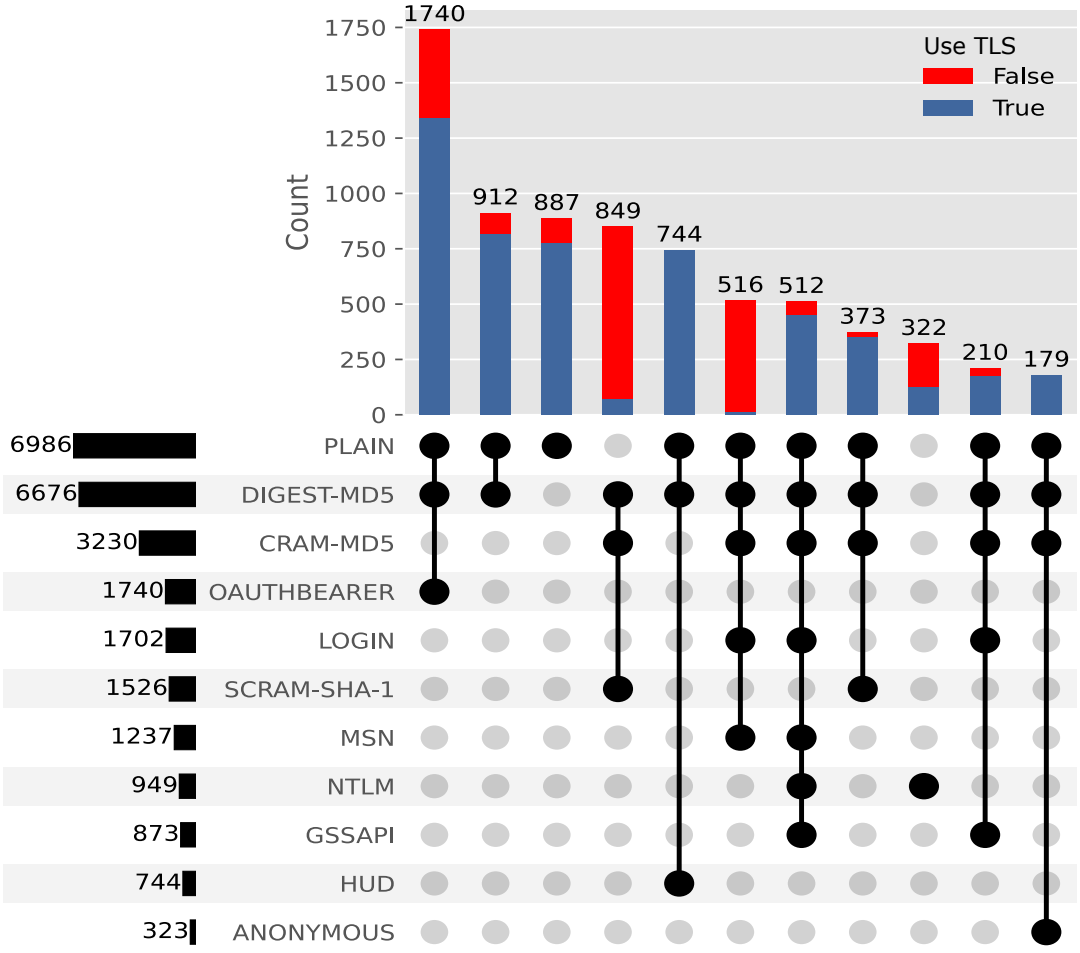


Figure 3.3: XMPP top 10 most frequent combinations of authentication mechanisms and count of observations using TLS.

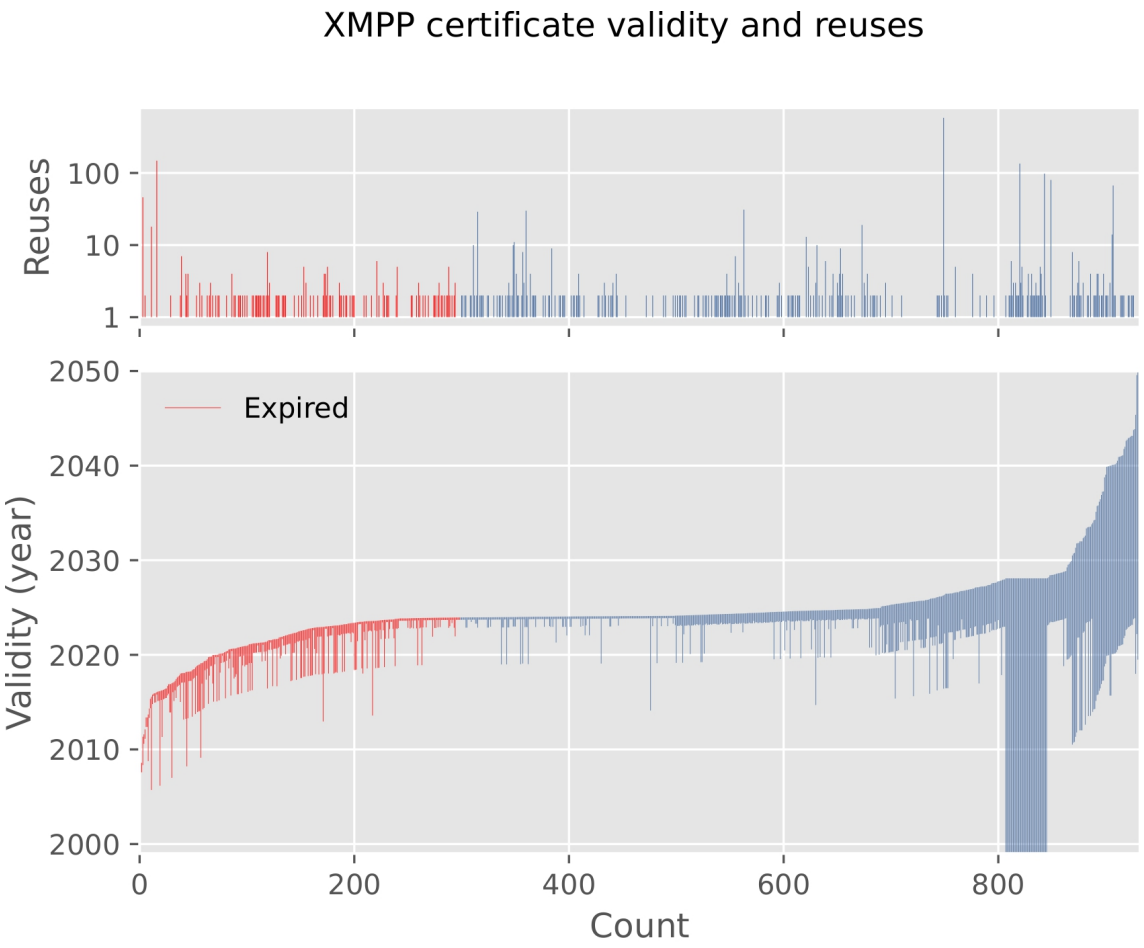


Figure 3.4: Validity of XMPP certificates and reuses. On the top, is the count of reuses for each certificate. On the bottom, the validity ranges for each certificate. Expired certificates are represented in red.

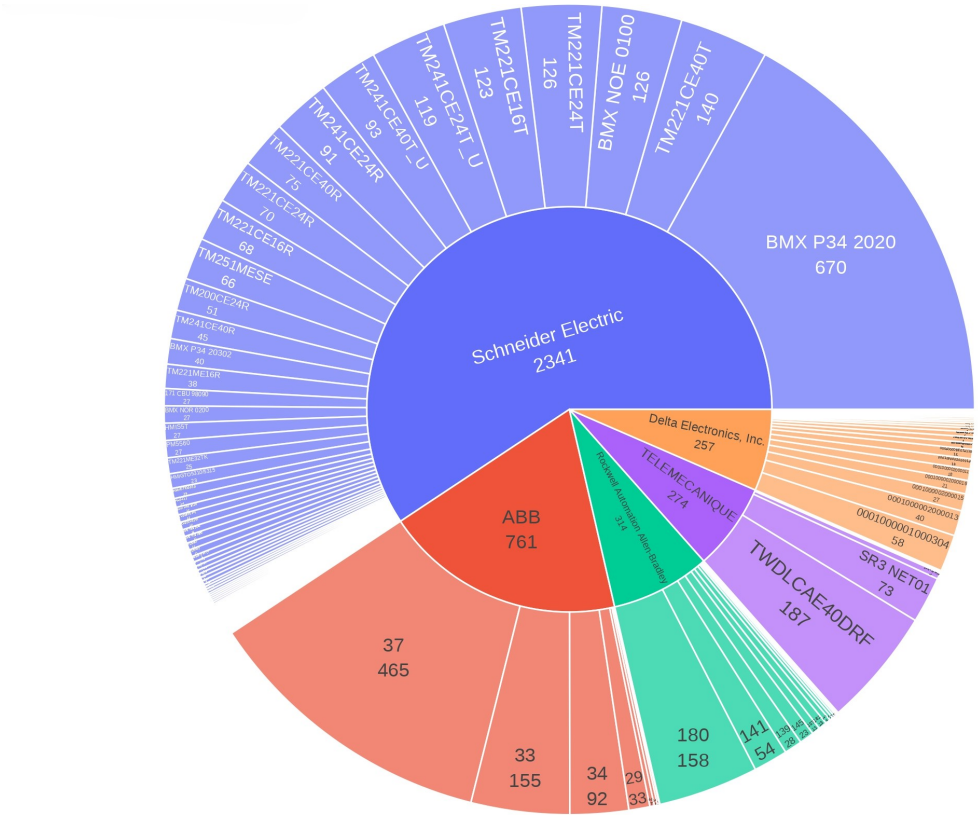


Figure 3.5: Top 5 vendor distribution of products exposing Modbus services on the Internet.

containing vendor and product names, unit functions, and other details. To reduce the load on the network, we limit our probe to a single packet with static identification values. More aggressive scanners can manipulate the probe to reduce the number of errors and invalid responses.

In total, we found 28,787 devices exposed to the Internet, of which 6,108 accepted our request and responded with their device information, nearly a 24% increase over the results from [4]. Our dataset contains 299 different products from over 80 vendors. Figure 3.5 shows the distribution of the four major vendors and products, the majority of which are generic Remote Terminal Units (RTUs) and Programable Logic Controllers (PLCs) from either Schneider Electric or ABB, with a total of 2,341 (8.13%) and 761 (2.64%) devices respectively. After further inspecting their product names and firmware versions, we find a large number of vulnerable generic controllers as well as sector-specific ones. For example, we found 659 `BMX P34 2020` controllers below the recommended version. Regarding sector-specific controllers, we primarily found solar monitoring devices (e.g., 181 Huawei SmartLoggers and 179 Solar-Log controllers), wind turbine monitoring devices, heat pump devices, and electric charger devices.

Takeaway: Environments exposing controllers to the Internet must implement further security measures to restrict communications with unknown devices, both inside and outside their network. Those devices communicating through Modbus lack basic security mechanisms, posing a risk to their own and other environments.

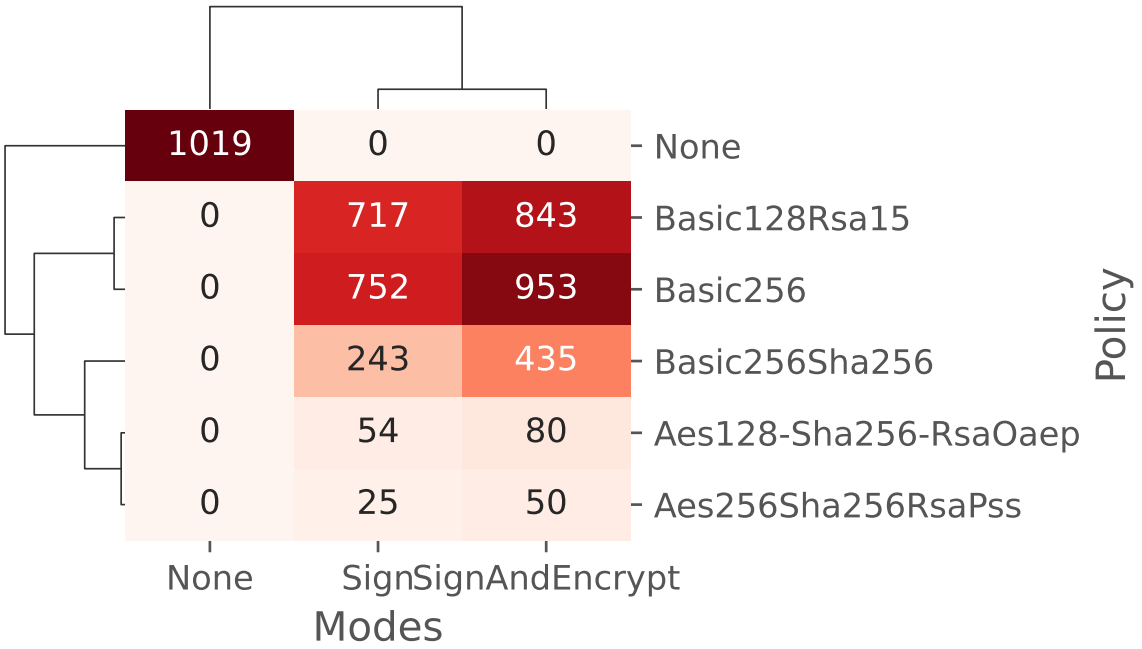


Figure 3.6: OPC UA combinations of security modes and encryption policies.

3.4.5 OPC UA

OPC UA is designed for abstracting PLC-specific protocols commonly found in ICSs. When properly configured, the protocol provides many standard security features, such as access control, and encryption. Before authenticating, clients can send discovery probes to retrieve the server security policies (used for encryption and key-derivation) and modes. We use this option to develop a simple probe that retrieves the server endpoint descriptions, selects the weakest policy and mode allowed, and then attempts to authenticate twice: first anonymously, and then using a self-signed certificate.

We identified a total of 1,797 exposed UA servers, a 38% increase over the results in [37]. Our scan discovered nearly 178 (9.9%) lacking basic authentication, of which 125 allowed anonymous authentication, and 53 allowed self-signed certificates. Allowing non-trusted sources to authenticate into UA servers is a serious violation of the minimum requirements for access control [37]. However, we cannot assess the severity of this flaw beyond this point since our probe closes connections immediately after the authentication without requesting any further information.

On the other hand, our results indicate that 59.6% of the total UA servers allow insecure combinations of security policies and modes (UA servers typically offer more than one policy for signing and/or encrypting messages). Figure 3.6 shows the correlation distribution between security policies and modes, with a staggering 59.6% of servers allowing insecure policies with no message signing or encryption, and a significant number of servers allowing deprecated security policies, such as `Basic256` (55.94%) and `Basic128Rsa15` (49.44%).

Lastly, we managed to collect certificates from 1,232 UA servers (approximately 68.56%). Apart from two, all other servers used self-signed certificates, with 604 (49.02%) servers reusing certificates. Our dataset contains 841 unique certificates from 70 different signers, most of which belong to manufacturers specialized in industrial controllers. Figure 3.7 shows the validity of the unique certificates we collected ranging from 2019 to 2050 (95% of the values), with 148 (nearly 17.59%) expired certificates colored in red, and the number

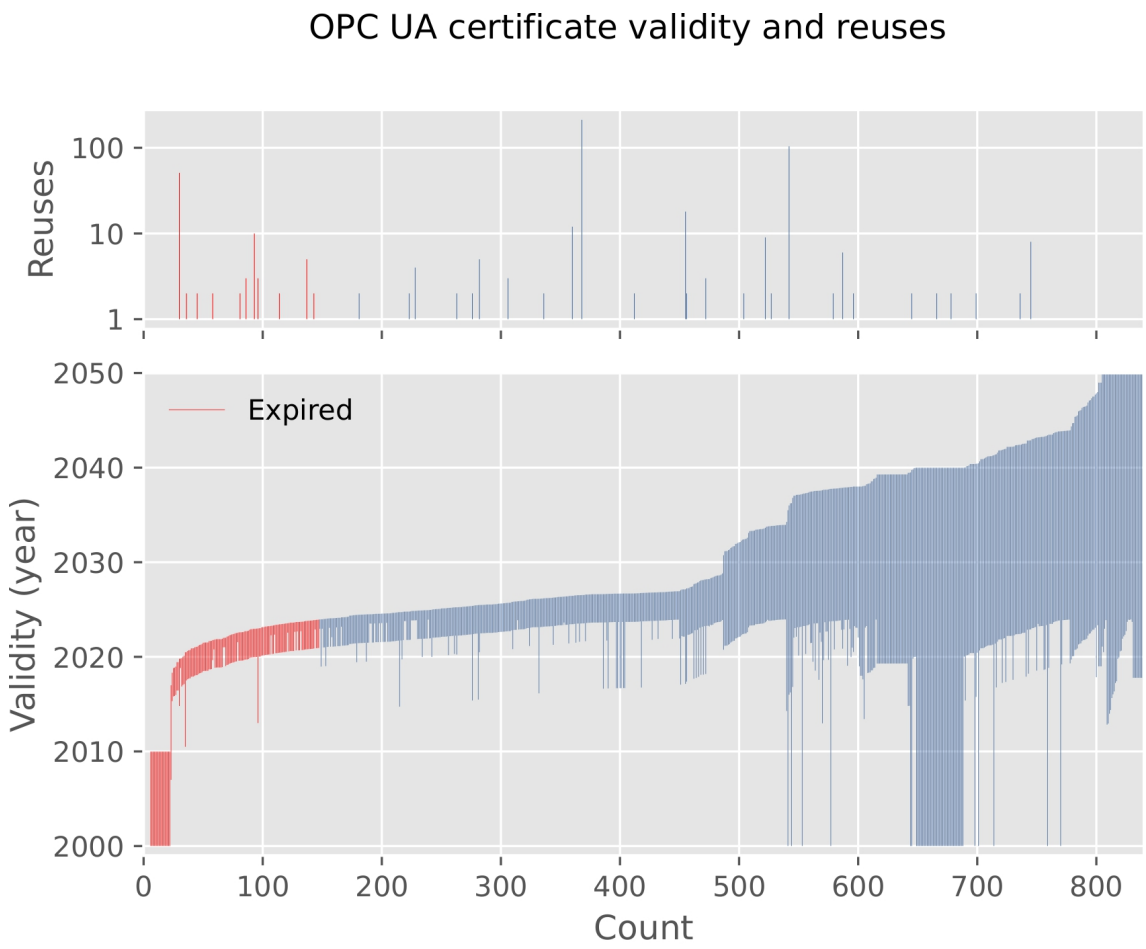


Figure 3.7: Validity of OPC UA certificates and reuses. On the top, is the count of reuses for each certificate. On the bottom, the validity ranges for each certificate. Expired certificates are represented in red.

of reuses for each certificate on top. The median duration of the certificates we observed is 5 years, similar to the default recommendations from most OPC UA implementations. However, 25% of the certificates violate this recommendation with validity durations between 20 to 50 years. Regarding the reused certificates, we underline two interesting cases: a certificate reused on *i.)* 211 different addresses in the same AS; and another on *ii.)* 104 addresses across 35 AS (valid for 10 years). These hand-picked examples show two different behaviors concerning many devices. In the first case, the consumer misconfigured the devices, while in the second the manufacturer hardcoded the server certificate on a range of automation devices.

Takeaway: UA servers with no authentication or weak combinations of security policies and modes lack access control. Moreover, servers with expired certificates or valid for the past 5 years are no longer considered secure or valid for cryptographic operations. In addition, reusing TLS certificates across multiple servers increases the attack surface, putting at risk all servers sharing the certificate when one of them is compromised. In total, we found 1,210 (67.33%) UA servers showing one or more of these characteristics, which we can safely classify as neglected or abandoned devices.

3.4.6 RTPS

RTPS is a publish-subscribe protocol used in real-time communications between distributed systems. RTPS is the wire protocol designed for DDS, allowing implementations from different vendors to interoperate seamlessly. This protocol is mainly used in industrial automation systems, smart grids, and other OT applications. Our probe uses the built-in discovery endpoints included in the protocol specifications to retrieve banner information available before authentication (e.g., vendor and version). Note that we choose not to join nodes [1] as participants.

We found 233 addresses exposing a total of 708 nodes, out of which 508 (71.75%) had unique values, to a mean value of 1.34 different nodes per address. `OpenSplice DDS` dominates the distribution of products and versions with a total of 408 (57.62%) using specification $v2.1$. Given the protocol specification versions are interoperable and imply only age, features, and open issues, we are not surprised that none of the nodes adopted the latest version ($v2.5$). Furthermore, we analyzed the combinations of protocol versions and products to identify potential issues. For example, `Connext DDS` introduced their support for $v2.2$ on their version 5.2 (released in 2015). We estimate that most nodes supporting $v2.1$ run on deprecated product versions, risking their integrity and participants. The most notorious vulnerabilities range from DoS to various overflows causing crashes. On a last note, and as previously reported in similar studies [68], we noticed that 167 nodes continued sending packets to our scanner for at least two hours, ignoring multiple flags included in our probe.

Takeaway: RTPS nodes exposed to the Internet that communicate with unauthenticated participants lack the basic governance required for these systems. For example, we found several devices to monitor and control railways and other critical systems. The severity of this issue is further aggravated in cases where non-trusted participants can read or change topics. These factors are known to be associated with precarious security policies. As a result, we perceive the 708 nodes as neglected, although the precise scale of this risk is unclear.

3.4.7 DNP3

This is a domain-specific protocol used in SCADA systems to relay messages between masters and slaves. Unlike other SCADA protocols, DNP3 SAv6 (an extension for this protocol) supports multiple security features, such as authentication and encryption [72]. Our probe targets devices that disable these security features, gathering responses from physical device addresses which allows us to create a link.

Our scan revealed 668 nodes exposed to the Internet, all of which included at least one linked device. In addition, we find several nodes to which we could establish 100 links, indicating that some of the nodes control large infrastructures. Proving that we can establish these many links is sufficient to estimate the size of the network and potential risks, although different probe configurations could establish links with the full range (65,520 links per node) to produce more accurate results. However, we could not identify the devices linked to the nodes, since our probe does not gather further information from the devices.

Takeaway: Since our probe targeted DNP3 nodes with most security features disabled, we conclude that the 668 nodes we found are either neglected of cyber-security, where administrators may choose to not use any security on their devices; or obsolete,

[1] Distributed systems use the term *node* referring to participant devices.

in the case of legacy nodes that do not support security features. It is trivial to see that nodes accepting writing requests from unauthorized users (e.g., command the device to stop) are vulnerable and a critical risk.

3.4.8 BACnet

BACnet is primarily used in building automation and sensor monitoring systems. This protocol uses a client-server structure, where clients can specify queries to read or write values. Some of the readable values include vendor description, software details, and device model. Since this protocol runs on UDP sockets and limits readings to one value per query, our probe generates significantly more traffic than others, requiring 9 different queries to fingerprint devices.

During our scan, we found a total of 7,251 BACnet servers exposed to the Internet using a variety of 488 products from 117 different vendors. Figure 3.8 shows the distribution of the 5 major vendors and products found during our scan. Notably, Tridium's Niagara 4 Station monitoring software makes up a substantial part of our dataset, accounting for 2,020 (27.85%) observations, alongside 417 Niagara AX stations (deprecated). From that count, 439 Niagara 4 stations are vulnerable to denial-of-service and cross-site scripting (XSS) attacks, and a few contain broken access control issues. Following closely, we identified various building automation controllers, such as 426 Delta Controls eBMGR and 406 JCI MS-NCE2506-0 controllers, representing approximately 5.8% of the observations each.

Takeaway: Without built-in security measures to protect BACnet communications, controllers and monitoring systems depend on their infrastructure to prevent exposure to the Internet [73]. Some manufacturers instruct the use of VPN for all BACnet communications. Therefore, we consider the 7,251 BACnet servers neglected, from which a large margin are obsolete or abandoned devices running on deprecated and vulnerable versions.

3.5 Discussion

3.5.1 IP reputation

We query Greynoise with the addresses of the devices we classify as vulnerable to find those seen scanning or attacking the Internet. Greynoise runs a large network of scattered sensors to capture and analyze suspicious traffic, behaving similarly to a network telescope. From the 675,896 addresses we classified as neglected, obsolete, or abandoned, Greynoise reported 7,424 of them and tagged 1,244 addresses as malicious. Most of these addresses were seen scanning for exposed SSH and telnet services or distributing malware. Table 3.1 includes a breakdown of the results per protocol; the counting is slightly higher due to some addresses exposing more than one vulnerable service. Note that these addresses may expose other vulnerable services besides the ones we target with our probes.

Diving deeper into the results, we distinguish various factors that may increase the probability of security breaches. In the case of XMPP, we see 665 servers with no encryption or authentication enabled, and the rest of the servers accept deprecated authentication methods (e.g., 54 servers using `DIGEST-MD5`). Then, most of the 30 suspicious OPC UA servers do not use any form of encryption or authentication, followed by insecure authentication combinations. Furthermore, MQTT brokers run mostly on deprecated versions with critical vulnerabilities. As for BACnet and Modbus devices, the distribution and products are evenly spread, showing that their infrastructure plays a crucial role in securing the

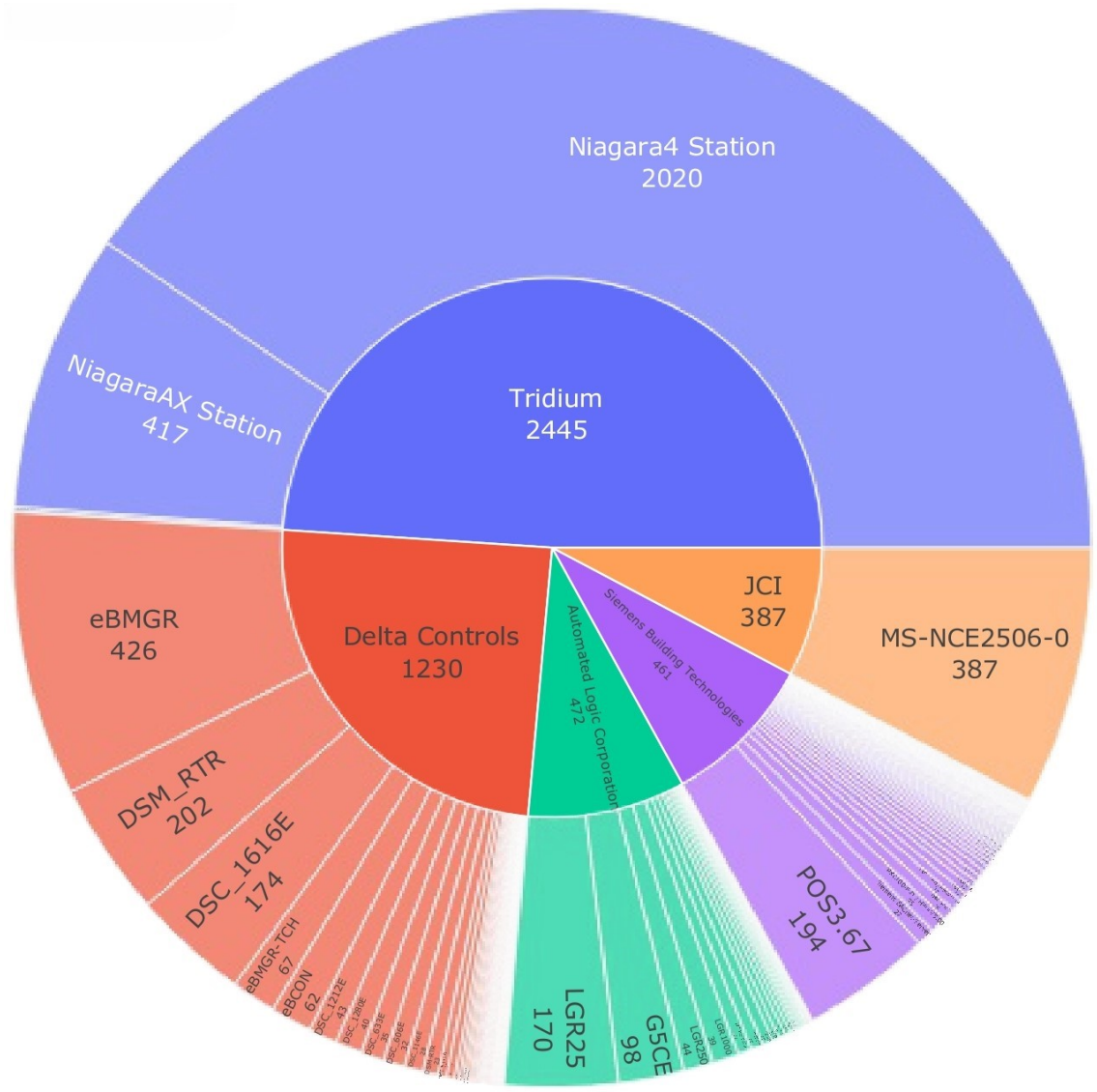


Figure 3.8: Top 5 vendor distribution of products exposing BACnet services on the Internet.

device. These findings align with the worst-case scenarios in our classification, indicating that most automated attacks use brute-force authentication methods and exploit known critical vulnerabilities. However, we do not find hard evidence linking certificate issues to compromised devices.

3.5.2 Vulnerability disclosure

We filtered our results to gather vulnerable addresses from ISPs in our region using `WHOIS` records. These records contain *abuse* email addresses to report suspicious activity originating from their IP ranges. A downside of relying on `WHOIS` records alone is that we are unable to directly contact the owner of the device [74]. Therefore, we enriched our results with Shodan information, which in some cases included further details such as the organization owning the device.

We were able to inform 30 organizations and ISPs through email following the recommendations in [74], including details such as the IP address, a timestamp, services affected, a description of our approach, and instructions to mitigate their risks. We received 5 responses so far, of which various organizations were unaware of their devices being exposed to the Internet (mainly ICS devices) and responded with very positive feedback. The rest of the responses were from ISPs, who had already contacted their customers regarding these obsolete and vulnerable systems. From these responses we learned that, at that time, most addresses were assigned to domestic households and mobile subscriptions, supporting previous findings regarding the precarious state of consumer and manufacturer cybersecurity postures [75], [76], [77]. Other authors raised their concerns regarding notification campaigns and the minimal impact on consumer behavior [40], [74], [78]. In general, the majority of notifications go unnoticed, are ignored, bounce back, or receive automated responses.

3.5.3 Summary

Most of the vulnerabilities we cover in this paper were associated with security management issues putting devices and networks at risk. We observed a general lack of proper access control, from severe cases of ICS devices used in building automation and railway stations that accept anonymous connections, to support center equipment pre-configured to accept insecure authentication methods. These security issues are worsened due to the absence of encryption, where most ICS protocols lack these capabilities altogether (e.g., Modbus and BACnet). We see that even though most protocols support encryption, it is often disabled or the device suffers from certificate management issues, with expired, long-lasting, or reused certificates. In addition, we discovered many certificates using weak encryption methods or short keys, which renders them useless. Some devices come with hardcoded certificates and default configurations which cannot be changed, while others may be unpatched, decommissioned, or obsolete. Overall, manufacturers and consumers approach cyber-security differently [79], [80]. However, it is a shared responsibility between them to maintain device security [76], [81].

Furthermore, we encountered some issues that prevented us from fully assessing the scope of the problem. As such, the numbers presented in this paper are likely to be conservative. For ethical reasons and to minimize intrusion, we designed our probes to close connections immediately upon receiving the banner, without testing the access level. In addition, some self-imposed limitations have impacted our results. In the case of MQTT, our probe only captured the names of 50 topics returned within the first 30 seconds. Extending the duration of the connection, removing the limitation to the number of topics, and capturing their values could produce very different results. For instance, we could not determine the device type or purpose from our results, although this information could

be inferred from other topic names. Similarly, our RTPS probe mimics the behavior of a single device and does not join the nodes to retrieve any topic information.

Moreover, our dataset showed significant differences compared to results from services like Shodan and Censys (e.g., small intersections, different values, and size of the datasets). For example, some of the results from Shodan were dated and did not represent the current state of an IP address. These services scan the Internet periodically, as opposed to creating a single snapshot of the Internet at a given time. Therefore, they are better suited for longitudinal studies.

In summary, we have shown that security maintenance issues are not unique to any sector of society in particular, but rather a common challenge. Many devices remain connected to the Internet for long periods despite being decommissioned, vulnerable, or already compromised; nevertheless, whether device owners accept the risks, ignore them or are unaware, remains an open question. While we received positive feedback during our vulnerability disclosure, it falls short to provide a conclusive answer. Further studies are necessary to address how society reacts to security advice and improve its security posture. Moreover, we have shown that these security issues are observable and targetable from the Internet using common tools with minor adjustments. The methodology presented in this paper relies on chaining patterns and filtering rules. However, further work is necessary to identify intricate vulnerabilities.

3.6 Conclusion

Throughout this paper, we presented an overview of the current landscape of IoT and OT devices exposing one or more of the targeted protocols. We identified 675,896 misconfigured, neglected, or abandoned devices exposed to the Internet. These devices lack security management, such as software updates, proper access control, or encryption mechanisms. A large margin uses deprecated or insecure authentication policies, such as allowing anonymous connections or accepting self-signed certificates. In addition, we find widespread deficiencies in certificate management, such as expired, long-lasting, and reused certificates. Furthermore, we examine the IP reputation of the potentially vulnerable devices and find that 7,424 of these addresses were reported previously by Greynoise, with 1,244 classified as malicious. Finally, we conducted an ethical disclosure of vulnerable devices discovered in our region. We shared insights on their responding behavior, showing that ISPs are the most active in notifying their customers. However, device owners rarely take action.

4 Digital ghost ships: abandoned, neglected, and obsolete IoT & OT devices exposed to the Internet

Context and Contributions

This chapter refines the findings of Chapter 3 by introducing the concept of Digital Ghost Ships (DGSs) as a unifying abstraction for persistent exposure caused by abandonment and neglect. By conducting longitudinal measurements, it directly addresses the temporal dimension of **RQ1**, showing that exposure is not only widespread but also persistent across years. The results demonstrate that mitigation efforts remain insufficient and that insecure deployments often linger despite increased visibility.

RQ	Contribution
RQ1a	Identification of persistent weaknesses
RQ1b	Characterization of exposed IoT and OT populations
RQ1c	Exposure mitigation through responsible disclosure campaigns

Related publication

R. Yaben and E. Vasilomanolakis, "Digital ghost ships: Abandoned, neglected, and obsolete iot & ot devices exposed to the internet," *Authorea Preprints*, vol. 1, pp. 1–12, 2025 [Preprint]

Original Abstract

The rapid adoption of Internet of Things (IoT) and Operational Technology (OT) devices to control systems remotely has introduced significant cyber-security challenges. Attackers have compromised millions of such devices over the years, exploiting their lack of management and weak cyber-security. This paper examines cyber-security issues of neglected, obsolete, and abandoned IoT and OT devices exposed to the Internet. To unify these issues under an umbrella term, we coined the term Digital Ghost Ships (DGSs). Our work focuses on identifying DGSs using common scanning tools to find indicators of security misconfigurations and misuse. Moreover, we compare two Internet-wide scans conducted two years apart, focusing on security issues in eight IoT and OT protocols: MQTT, CoAP, XMPP, Modbus, OPC UA, RTPS, DNP3, and BACnet. During our first scan ($S1$) we found 675,896 DGSs, and 75,007 during our second scan ($S2$). Lastly, we examine the IP reputation of the vulnerable devices and find that 7,424 ($S1$) and 792 ($S2$) DGSs were reported at least once.

4.1 Introduction

The emergence of the IoT and OT has permeated most aspects of our lives. From smart home devices to medical instrumentation and critical infrastructure, all sectors of society are rapidly becoming reliant on these new technologies. While their benefits are undeniable, their rushed adoption introduced new risks and security issues, inviting adversaries to take control of those lacking security. Recent large-scale IoT attacks such as the Mirai botnet [1], powered by close to a million compromised devices, have evidenced society's

challenges in securing devices, posing a major threat to their environment and other systems. These challenges urge the security community to develop new mitigation strategies. The state of the literature already includes several studies that focus on the landscape of IoT and OT devices exposed to the Internet [38], [52], [53], propose mitigation strategies to reduce the number of exposed and vulnerable devices [54], or investigate society's cyber-security posture towards their devices [5]. However, studies dedicated to identifying devices that are neglected in terms of cyber-security, obsolete – yet in use – or abandoned are scarce. Such devices lack basic security, such as authentication and encryption. They also leak sensitive information, skip major updates, use deprecated (insecure) features, or are decommissioned and no longer receive security updates.

We present this paper as an extension to our previous work in [12] to tackle this gap in the literature, comparing two Internet-wide scans conducted two years apart (December 2023 and January 2025 respectively) in a longitudinal analysis. Each dataset contains the results of an Internet scan using extended versions of the ZMap and ZGrab2 tools, supported with data from Shodan [55] for granulated classification, Greynoise [57] to identify malicious hosts, the NIST vulnerability database for known vulnerabilities, and RIPEstat [82] for routing information to notify device owners in our region (Denmark). Our main contribution consists of identifying security issues associated with neglected, obsolete, and abandoned IoT and OT devices exposed to the Internet through the lens of Internet-wide scans targeting eight protocols commonly found in these devices: MQTT, CoAP, XMPP, Modbus, OPC UA, RTPS, DNP3 and BACnet. Furthermore, we introduce the term DGSs to describe devices sharing such characteristics. Our year-to-year analysis suggests that the number of DGSs is generally decreasing, but most of the ones we discovered during our first scan reappeared during the second, often unchanged, and on occasions downgraded. We summarize our key findings as follows.

- We measure the IPv4 twice with a year gap between scans targeting eight protocols commonly used in IoT and OT devices. In addition, we analyze the responses from our scans following a systematic method to identify misconfigurations and indicators of misuse, such as broken access control, certificate management issues, and leaks of sensitive data.
- Our first scan ($S1$) revealed 618,765 DGSs, and our second scan ($S2$) 75,007 DGSs, showing an overall decrease of DGSs facing the Internet. Yet, most DGSs found during $S2$ were also observed during $S1$, often without observable changes between datasets.
- Using IP reputation services, we show that 7,424 hosts in $S1$ are reported as suspicious or malicious, some of which appear infected with Mirai variants and other malware families. We find 792 hosts in $S2$ with similar characteristics.
- We conducted vulnerability disclosure campaigns for each scan to notify owners of DGSs in Denmark and discussed some insights on the responses we received.

The remainder of this paper is structured as follows. Section 4.2 begins with an overview of the relevant literature for identifying vulnerable IoT and OT devices over the Internet. In Section 4.3, we briefly introduce the scope of our work and our approach to scanning the Internet, as well as the ethical considerations and our self-imposed scanning limitations. Then, in Section 4.4 we analyze our scanning results to identify DGSs. Lastly, Section 4.5 summarizes our findings, touching on the IP reputation of the potentially vulnerable devices we discovered, and the responses to our vulnerability disclosure. Section 4.6 concludes this paper.

4.2 Related Work

Numerous studies conduct Internet-wide scans to investigate vulnerabilities in IoT and OT devices [58], [59], [60]. The methods for scanning the Internet are well-established [10] and most authors use off-the-shelf common tools such as those from the ZMap ecosystem [50] or Masscan [61], alongside meta-scanners (e.g., Shodan and Censys), and IP reputation services (e.g., Virustotal and GreyNoise). Authors extend or develop new probes for these tools to cover different use-cases; however, their scanning choices largely depend on the scope of their work (e.g., vantage points, number of scans, and period) [9].

A significant part of the literature focuses on ICSs exposed to the Internet [37], [47], [63], given that many of those systems operate in critical environments and lack security features. In [4], the authors conducted multiple full IPv4 scans targeting nine ICSs-specific protocols with custom ZMap probes. They report finding over 60,000 exposed systems, some of which belong to critical infrastructure organizations, airports, and government facilities. Lastly, they supplement their work with an IP reputation analysis using a Network telescope to identify malicious traffic originating from these addresses. In another study, [5] introduced a 5-year longitudinal analysis using Shodan and Censys to fingerprint devices exposing either of 6 ICSs protocols. The authors offer a holistic perspective on this issue including human aspects in their study, such as owner security behaviors, and economic motivations driving cybercriminals. More recently, [39] studied the use of TLS in 10 ICS protocols, showing that less than 7% of nearly a million exposed devices secure their communications.

In addition, there has been a notable effort to identify vulnerable IoT and OT devices exposed to the Internet [35]. In [52], the authors scanned for specific IoT devices over the Internet to identify vulnerabilities and other issues associated with this technology. Moreover, [64] scanned the IPv6 space instead, targeting six common IoT protocols. They identified 36,400 IoT devices, highlighting security concerns such as non-trusted and expired TLS certificates. Lastly, the work of [38] is the closest to our study, focusing on misconfigured IoT devices exposing one of five widespread protocols. They also include a reputation analysis of the misconfigured devices they found using a network telescope and multiple honeypots, an analysis of the attack trends on each of the protocols they support, and a brief discussion on the attacker behavioral patterns they observed. The major difference with [38] is in the aim of our work; [38] centers on current attack trends on IoT devices using honeypots and network telescopes, however, the cornerstone of our study is to identify vulnerable IoT and OT devices from their response behavior. Our study is inspired by these approaches to identifying vulnerable devices beyond matching CVEs, including other factors such as lack of authentication and encryption, access control issues, and disclosing internal resources.

In summary, most authors have focused on introducing new methods to fingerprint IoT and OT devices and identifying their vulnerabilities. The state of the literature includes many valuable lessons about the risks of exposing these technologies to the Internet and how to secure them. However, few authors draw on the security behaviors leading to such vulnerabilities, failing to represent the bigger picture: these devices are poorly maintained. To address this gap, we shift our attention from common vulnerabilities to how these devices are handled in practice, investigating the state of obsolete, neglected, and abandoned devices that remain connected to the Internet.

4.3 Methodology

In this section, we present our approach to identifying Internet-exposed devices that exhibit signs of *abandonment*, *obsolescence*, or cyber-security *neglect*. We define *neglected* devices as those lacking basic access controls, displaying poor security hygiene, misconfigured, or missing critical security updates. These devices are characterized by their weak or lack of authentication and encryption mechanisms, certificate-related issues (such as reused and expired), and leaky or generally insecure configurations. Moreover, *abandoned* devices suffer from the long-lasting effect of being neglected. We distinguish those devices with indicators for their overall usage, such as deprecated software or configuration, and major certificate issues relative to the validity recommendations (e.g., long-lasting and legacy certificates). Lastly, *obsolete* devices lack essential security features for Internet communications, such as legacy or decommissioned devices that remain active even though they no longer receive official support. Therefore, our methodology mainly focuses on access control and security maintenance issues.

The remainder of this section covers our methods for scanning the Internet, ethical considerations, technical limitations, and our classification pipeline to identify DGSs.

4.3.1 Scanning the Internet

We divide our scans into two phases following the learnings from the literature [9], [26], [50], [65], [83], first scanning for L4 protocols using ZMap to identify responsive services (L4 UDP scans send protocol-specific probes, requiring separate scans for each L7 targetted protocol), followed by L7 scans using ZGrab2, which complete full connections to collect banner information and handshake details [10]. This method reduces the duration of the scans and the amount of traffic we generate toward each address.

To enable targeted scanning for DGSs, we extended both applications with new (MQTT, CoAP, OPC UA, and RTPS) and modified probes (XMPP and Modbus). Then, we scanned the Internet twice from a local vantage point, once in December 2023, and again in January 2025, excluding the addresses of those who had previously requested to opt-out of similar studies [66].

Lastly, we host a website on the vantage point with details of our study (e.g., targeted protocols and ports), and opt-out and abuse contact information. In addition, to help administrators identify our traffic, we retain the ZMap default IP identifier and include a signature on our probes with the address of our website and the name of our institution. This signature can be found in header fields or directly in the payload of protocols that accept content in the request's body, namely MQTT (client ID and certificate), OPC UA (client URI and certificate), and Modbus (request ID). However, the rest of our probes do not support sending additional data without breaking the protocol's standard.

4.3.2 Ethical considerations and limitations

Conducting Internet-wide scans produces a substantial load of traffic on target networks [10], [67]. Therefore, we implement several technical measures to mitigate the impact of our scan and our level of intrusion. For example, we use the randomization features from ZMap to ensure a maximum distance between each probe targeting the same block of addresses [50], including a minimum of 15 seconds between probes to the same address.

Furthermore, our probes only establish anonymous communications with their targets, using empty credentials or a self-signed certificate (when authentication is required). In addition, we follow a similar approach to other authors [38], limiting our connections to 30 seconds and setting limits to the amount of data we gather (cf. Section 4.4 for the individual implementations).

Lastly, we conduct a notification campaign for the owners of vulnerable devices in our region. We limited the notification/disclosure campaign to Denmark as this required thorough analysis and individual notifications. In this context, future work would benefit from automated notification of misconfigured devices. We discuss the general aspects of their feedback in Section 4.5.2.

We acknowledge that our scanning methodology has some limitations: i) scanning from a single vantage point limits our scanning visibility, and ii) reusing vantage points may have similar effects. Wan et al. estimated that these issues significantly impact the results of general Internet measurements, and our results should be interpreted considering these factors [9] .

4.3.3 Data processing and classification

To focus on relevant data, we fine-tuned our scanner to exclude specific responses. First, our scanner dropped echoed responses with identical information to our requests. Echo responses are common in low-interaction honeypots. While it could be interesting to apply our methodology to identify vulnerable honeypots as well (i.e., honeypots with unintended vulnerabilities), we will not study honeypots in this paper. Moreover, we exclude duplicate responses from the same address and service; we noticed this behavior while testing our methodology on 1% of the Internet, most likely caused by servers not receiving RST packets to close the connection, Internet churn, packet loss and drops, and other common issues associated to Internet scanning as documented in [6], [9], [50]. Adding to this, we are aware of the behavior of some controllers exposing RTPS services that will not stop transmitting data for long periods [68].

Before classifying our datasets, we flag responses following the targeted protocol and enrich our results with intersecting data from Shodan, querying the meta-scanner for the addresses in our dataset instead of merging their observations with ours. As other authors pointed out [5], meta-scanners do not provide sufficiently accurate snapshots of the IPv4. Therefore, we only use Shodan's data to complement our results and mitigate our probe's limitations.

Our classification method examines three main aspects of the communication with exposed services to determine whether we can consider them DGSs: i) access control mechanisms, ii) certificates, and iii) service details. We classify hosts as DGSs when they meet one or more criteria as summarized in Table 4.1. For access control, services qualify when they do not require authentication, allow anonymous access, or allow authentication via self-signed certificates. In most cases, our probes attempt to access other resources to verify we can read values from the service. We should note that the protocols we study in this paper are not intended for unrestricted or publicly accessible services. In addition, we assess the encryption mechanisms and authentication methods, classifying services as DGSs on their absence, or when these are weak or deprecated. Furthermore, we inspect the validity of the certificates, as well as their reuse across other hosts, and evaluate their encryption and signing algorithms. Invalid certificates include expired, long-lasting (according to the certificate and service specifications), and certificates set in the future or with a negative validity range. We also include hosts that reuse the same certificate or public key as others. Lastly, we analyze the service details included in the communication to determine the names and versions of the service and device, information leaks (e.g., configuration, access to resources, etc.), and usage indicators, such as timestamps and incremental counters. This process is protocol-specific and we discuss it as part of our results in Section 4.4.

Table 4.1: Classification criteria for DGSs

Class	Subclass	Criteria
Access control	Authentication	None, anonymous, self-signed certificate
	Access level	Read or write
	Encryption	None, weak or deprecated encryption
Certificates	Validity	Expired, long-lastling, invalid range
	Reuse	Reused
	Encryption and signing	Weak or deprecated algorithms, and short keys
Service details	System information	Vendor, product name and version, firmware version
	Software information	Service name and version
	Internal leaks	Internal state, configuration, resource access, network topology, sensitive information, etc.
	Usage	Timestamps, incremental counters

Table 4.2: Summary of exposed and vulnerable services per protocol. Probe: ○ default, ◐ modified, ● new.

		$S1$		$S2$		
Protocol	Port	Flagged	DGSs	Flagged	DGSs	Probe
MQTT	1883	424,961	424,961	27,382	27,382	●
CoAP	5683	228,536	58,083	40,602	38,116	●
XMPP	5222	150,472	28,575	9,260	2,896	◐
Modbus	502	6,186	5,894	874	759	◐
OPC UA	4840	1,708	1,706	1,812	1,174	◐
RTPS	7400	233	232	242	58	●
DNP3	20000	697	399	191	59	○
BACNet	47808	34,637	8,671	21,335	4,860	○

4.4 Results

This section provides a protocol-by-protocol analysis of our scanning results. First, we cover general-purpose IoT protocols (i.e., MQTT, CoAP, and XMPP), followed by OT protocols primarily used in SCADA systems (i.e., Modbus, OPC UA, RTPS, DNP3, and BACnet). Each protocol is analyzed systematically, with i) protocol descriptions, probe details, and DGSs classification method, ii) individual scan results and comparison of our findings, and iii) a takeaway of major risks and potential mitigations.

Our results are summarized Table 4.2 and organized by scan, protocol, flagged observations, DGSs, and the extent of our input on the development of the probe. The results presented in this paper were calculated using revised and more rigorous criteria compared to our previous work in [12], which led to lower numbers even though the $S1$ dataset is the same. In addition, we improved the precision of our probes targeting DGSs, resulting in fewer potential DGSs and a higher ratio of true-positives during $S2$.

4.4.1 MQTT

This is a publish-subscribe protocol commonly used in IoT environments. In MQTT, clients communicate through brokers that store messages in path-like topics. Our probe targets brokers that either lack authentication or accept self-signed certificates, subscribing to all

internal and public topics and collecting at most 50 topic names. During $S2$, we also collect one message from each topic to identify additional broker implementations and their usage. Furthermore, we use the following criteria to classify brokers. Neglected brokers lack encryption and grant read access to their topics, leaking sensitive information such as their version and usage. Brokers are considered abandoned when they are significantly outdated, or their certificate is no longer valid (e.g., reused, expired, long-lasting). Lastly, obsolete brokers include those with deprecated implementations.

Scan $S1$ included 424,961 responses with one or more topics from brokers without encryption or authentication, of which 424,013 were Mosquitto, and 40 HBMQTT brokers. We found 404,471 Mosquitto brokers running on vulnerable versions, with 11 brokers using $v1.0$-beta. In addition, we cross-referenced the broker version with known security vulnerabilities to assess the risks of using outdated software. Figure 4.1 shows the mirrored distribution from both datasets of the broker version, colored with the vulnerabilities affecting those versions representing their severity. This highlights a serious concern: at the time, most brokers were outdated and prone to severe vulnerabilities, from buffer overflows to total device takeovers. Therefore, we classified the 404,471 Mosquitto brokers with versions prior to $v2.0$ as abandoned, and the rest as neglected. In addition, we classify HBMQTT brokers as obsolete since the project was deprecated in 2020. Scan $S2$ included 2 million responses, of which 275,333 allowed unauthenticated connections, and 27,382 allowed subscribing to one or more topics and classified as DGSs. From their topic names and values, we identified 22,080 Mosquitto (see Figure 4.1), 1,015 ActiveMQ, 760 EMQX, 133 VerneMQ, and 35 Erlang MQTT (deprecated) brokers. Their topics show a variety of IoT products, such as home assistant hubs (e.g., 214 `zigbee2mqtt/bridge/devices`), alarm monitors (e.g., 1,190 `sys/001/alarm_cfg`), and vehicle monitors (e.g., 66 `teslamate/cars/1`).

While the number of brokers exposed to the Internet has barely changed, most brokers now include some form of authentication and authorization. According to dataset $S2$, Mosquitto continues to be the most popular broker, and though the distribution of versions remains consistent, the number of DGSs has decreased dramatically. However, the overlap between the datasets proves the persistence of DGSs, with 9,723 brokers common to both, and 4,805 remaining unchanged. The most common brokers appearing in this intersection are 1,174 Mosquitto brokers between versions $1.6.9-1.6.10$, and 1,349 between versions $1.4.13-1.4.15$, suggesting they have been unattended for several years. From those that changed, we find numerous instances of version downgrades across the board and replacements from Mosquitto to other brokers (e.g., ActiveMQ or EMQX). In addition, we find an interesting group of 105 Mosquitto brokers that received downgrades from various versions between $v1.6$ to higher than $v2.0$ that defaulted to version $v1.4.8$.

Takeaway: Allowing anonymous clients to interact with brokers is a non-negligible risk that may lead to further attacks. Furthermore, topics containing internal or sensitive information must be restricted with access control policies and require authentication. Lastly, we observed some brokers with unique identifiers or aliases as topic names, which, while adding some obfuscation, should not be used as a security replacement.

4.4.2 CoAP

CoAP enables constrained devices to communicate over the Internet using a RESTful architecture. Our probe sends an anonymous resource discovery request to identify exposed endpoints and gather server implementation details. This probe targets CoAP servers without authentication or encryption, narrowing the number of responses we receive. Furthermore, we classify CoAP servers as neglected when granted read access

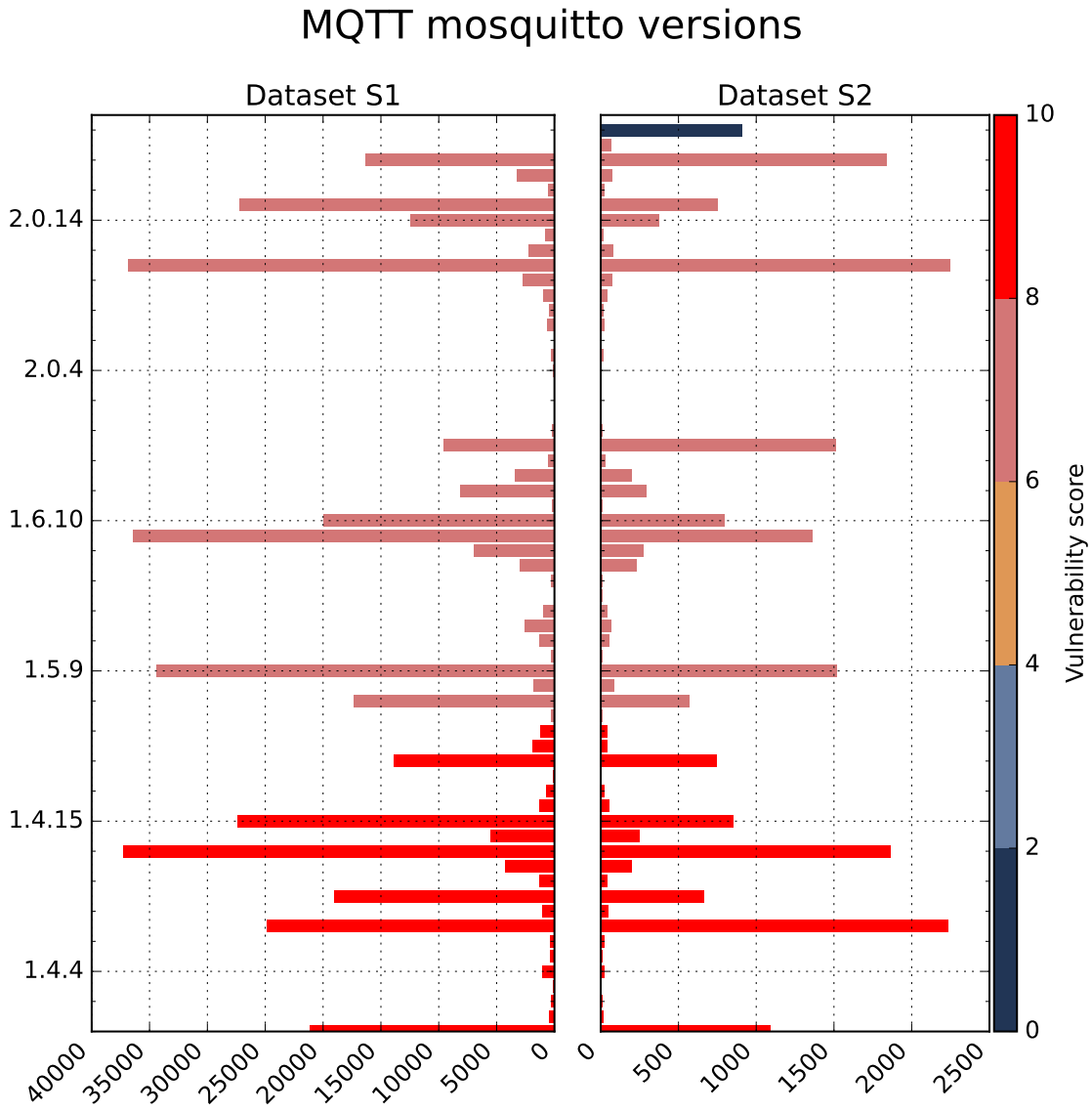


Figure 4.1: Distribution of MQTT Mosquitto versions colored by the severity of their vulnerabilities, with 0-2 in dark-blue indicating no vulnerabilities found.

to one or more resource endpoints, as those leak sensitive information. Lastly, we classify servers as obsolete or abandoned when their implementation version is significantly outdated or deprecated.

Scan $S1$ produced 228,536 CoAP results, with 58,083 disclosing their resource endpoints under the `/.well-known/core` path. Their resource descriptions show that 14,678 are vulnerable QLink routers and 1,067 Efento NB-IoT wireless sensors. In addition, a staggering 15,380 CoAP servers operated on insecure Eclipse Californium versions, while 41,792 relied on outdated libcoap implementations, leaving exposed resource endpoints highly vulnerable to attacks. Furthermore, dataset $S2$ included 38,116 CoAP servers exposing one or more endpoints. Their endpoints indicate that 27,024 were QLink-vulnerable routers, and 806 were Efento NB-IoT sensors. In contrast, $S2$ did not include any vulnerable Californium servers, and the number of vulnerable libcoap servers dropped to 24,322. The fact that 11,929 of these servers were also present in $S1$ suggests they are likely abandoned.

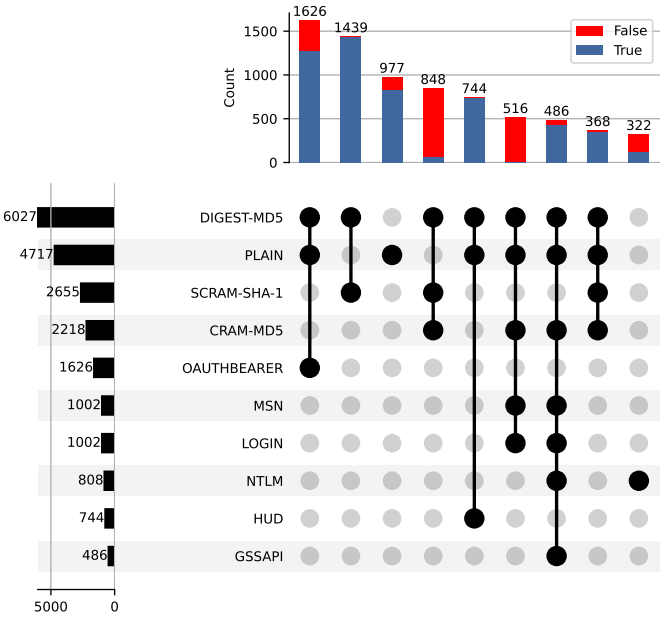
Takeaway: Allowing anonymous clients to communicate with CoAP servers has severe security and privacy implications. Clients can access sensitive information, such as implementation details and device characteristics. In addition, CoAP servers without security options enabled (NoSec mode) can be used on amplification attacks. Therefore, all resources, including the `/.well-known/core` discovery path, must be limited by access control policies and authentication, without disclosing the existence of other paths or returning unnecessary data to unauthorized users.

4.4.3 XMPP

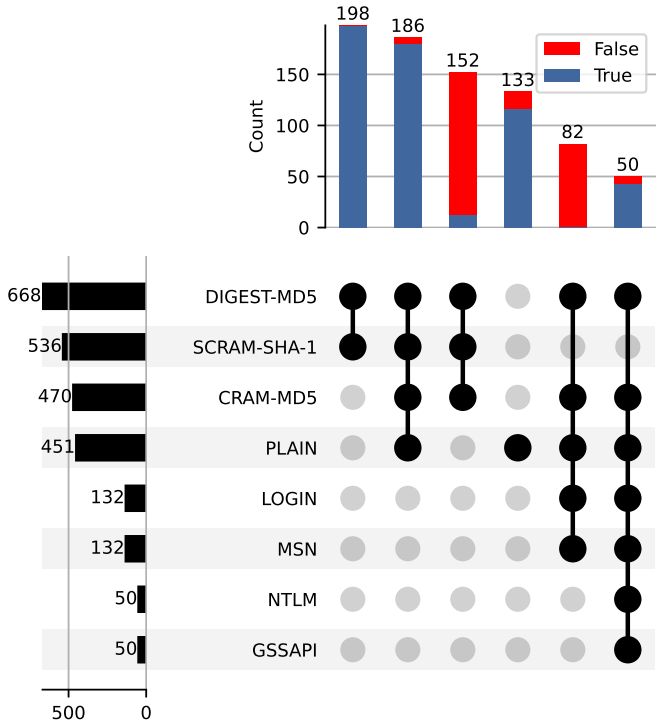
Previously known as *Jabber*, XMPP is the open Standard for messaging applications based on XML. Today, this protocol offers an alternative to MQTT and CoAP in constrained devices such as printers and sensors. Our probe negotiates a stream channel to assess the server's supported authentication and encryption capabilities. Servers offering no authentication or encryption are classified as neglected, while those relying on deprecated methods are considered obsolete. In addition, we use Shodan to gather server certificates, classifying servers as neglected when their certificate was reused or long-lasting, obsolete when using deprecated encryption or signing algorithms, and abandoned when certificates are expired.

Our $S1$ scan included 397,275 responses, of which 150,472 were XMPP. From this count, 9,564 included XMPP stream negotiation mechanisms, of which 9,175 included one or more being deprecated. In addition, we observed 1,689 servers using stream compression, an obsolete configuration option due to a chosen-plaintext vulnerability, and 6,340 servers using the obsoleted authentication protocol `iq-auth`. In contrast, our scan $S2$ draws a different picture; the number of responses is similar but the number of XMPP servers we observed is far lower (9,260 in total, 1,1133 disclosing authentication mechanisms), though the most common combinations of mechanisms remain the same: deprecated authentication methods and two particular groups without TLS support. For brevity, the $S2$ dataset includes 1,098 servers with deprecated authentication mechanisms and 700 supporting `iq-auth`. Figure 4.2 shows the most frequent combinations per dataset, often involving one or more deprecated options: `DIGEST-MD5`, `CRAM-MD5`, and `SCRAM-SHA-1`. It is worth mentioning that servers supporting plain-text authentication are expected to encrypt the communication using TLS and hashing the credentials [70].

Furthermore, we collected 14,735 unique certificates from Shodan for $S1$ and 952 for $S2$; Figure 4.3 shows the unique certificates and their validity ranges, with the number



(a) Dataset S1



(b) Dataset S2

Figure 4.2: XMPP top 10 most frequent combinations of authentication mechanisms and count of observations using TLS.

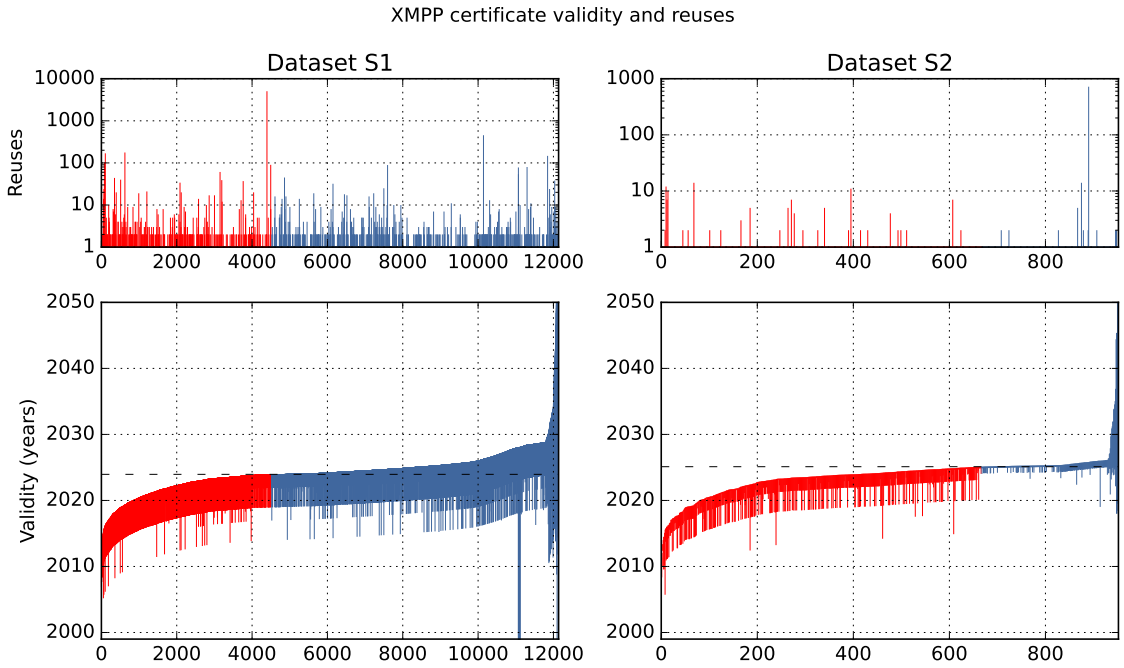


Figure 4.3: Validity of XMPP certificates and reuses. On top is the count of reuses for each certificate. On the bottom, the validity ranges for each certificate. Expired certificates are represented in red.

of reuses on top. Most certificates are long-lasting, expired, or reused, mainly linked to contact and call center equipment such as VoIP phones. Among $S1$ devices, the top two certificates were seen in 582 and 148 devices; both correspond to VoIP phones by the same vendor, though the certificate used by 148 devices had expired in 2016. In $S2$, the most frequent certificate appeared in 721 VoIP devices. The high load of devices from the same manufacturer suggests that most come preconfigured by default. This highlights a common issue among IoT and OT devices, where manufacturers oversimplify security and consumers struggle to maintain it.

The intersection between datasets includes 7,050 servers, i.e., 70% of the servers observed during $S2$. From this count only 461 servers remained unchanged, while the rest received multiple updates, most of which renewed certificates, modified authentication mechanisms, and added support for TLS. Once more, we observed a few downgrades as well, where some of these servers began supporting deprecated mechanisms or dropped support for TLS.

Takeaway: To comply with the XMPP specification, servers must use SASL or TLS to establish a secure communication channel and disregard deprecated authentication mechanisms putting the communication at risk. In addition, owners are responsible for maintaining server certificates, avoiding reuses, and renewing certificates as they expire, with a recommended validity period of just over a year.

4.4.4 Modbus

Modbus is a master-slave protocol for industrial automation and control systems. This protocol lacks built-in security features, allowing adversaries to eavesdrop on connections in plain text, read (and potentially write) device information, flood them with traffic, and leverage compromised devices in further attacks [4], [71]. Our scanner uses the de-

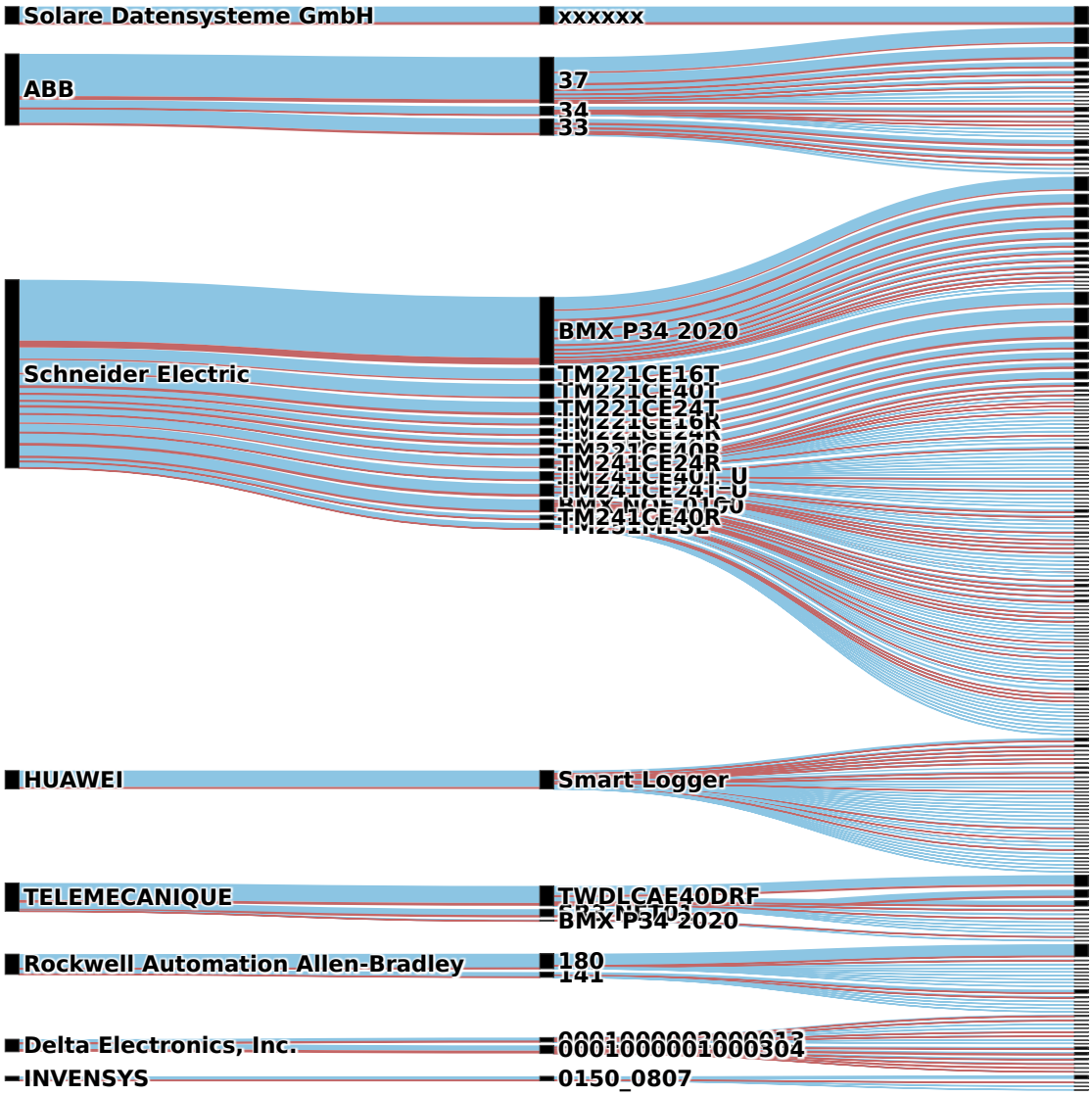


Figure 4.4: Top vendor distribution of products and their firmware versions exposing Modbus services on the Internet.

fault ZGrab2 probe to send Read Device Identification requests, which trigger a response containing vendor and product names, unit functions, and other details. Modbus masters from which we can read values are classified as obsolete.

The $S1$ dataset contains 6,186 responses from master servers, of which 5,894 included device information; this is nearly a 24% increase over the results from [4]. This dataset contains 288 different devices from 82 vendors. Figure 4.4 shows the distribution of the major vendors, products, and firmware versions, holding 75% of the responses in both datasets, with $S1$ colored in blue and $S2$ in red. Most of the devices we observed during this scan were generic RTUs and PLCs from either Schneider Electric or ABB, with a total of 2293 and 858 devices each. From their firmware version and product name, we conclude that a significant number of devices contain known vulnerabilities or are deprecated. For example, we found 659 `BMX P34 2020` controllers below the recommended version. Regarding sector-specific controllers, we primarily found solar monitoring devices (e.g., 181 Huawei SmartLoggers and 179 Solar-Log controllers), wind turbine monitoring de-

vices, heat pump devices, and electric charger devices. For $S2$, we received responses from 874 servers, with 759 including their vendor and product details. The case of Modbus is particularly interesting, as 421 of the servers found during $S1$ were also found during $S2$, with 364 remaining unchanged. This means that 79 devices received some maintenance over the past year, but not always for the best. We observed multiple cases in which devices were now disclosing more information than before (e.g., product name, and version) and two severe cases in which devices were downgraded (a `BMX P34 2020` and a `TSXETY4103`, both from Schneider Electric). In addition, we did not find any indicator of devices being replaced. However, regardless of their updates, all these devices remain exposed to the Internet and accept requests from unknown clients.

Takeaway: Environments that expose SCADA controllers to the Internet must implement further security measures to restrict communications with unknown devices, both inside and outside their network. Those devices communicating through Modbus lack basic security mechanisms, posing a risk to their own and other environments.

4.4.5 OPC UA

OPC UA is designed to abstract legacy protocols commonly found in ICSs. When properly configured, the protocol provides many standard security features, such as access control and encryption. OPC UA servers typically expose a discovery application disclosing accessible endpoints, which include their supported security policies, modes, and authentication mechanisms. Endpoints give access to registered nodes, which may implement individual access control policies to restrict clients from reading or writing data and executing functions. For our first scan, we used a simple probe that attempts to authenticate into each endpoint using an anonymous user and a self-signed certificate, without accessing nodes. The information we collected with this probe is limited to endpoint descriptors and whether we could authenticate. For our second scan, we used the probe provided in [37], which allows us to browse through nodes. Certain OPC UA nodes contain relevant system information, such as manufacturer and product details. It is important to mention that due to the complexity and rising relevance of the protocol – and the timing of this publication – this scan was analyzed separately and covered as an individual study in [14]. Servers without weak policies allowing us to authenticate into one or more endpoints are classified as neglected, while those supporting deprecated security policies are classified as abandoned.

Our $S1$ scan included responses from 1,708 OPC UA servers exposing endpoints, consistent with the results from [37], a similar study that identified between 1,761 and 2,069 servers over a series of scans in 2020, in which they found several issues in 95% of 1,114 of these servers exposing endpoints. From these, 1,118 forego all security, allowing clients to join anonymously and without encryption. Moreover, 1,010 of these servers advertise deprecated security policies, such as `Basic256` (56%) and `Basic128Rsa15` (49%), defeating the purpose of encrypting the communication. This allowed us to authenticate to 173 servers through their endpoints, 169 as anonymous clients, and 78 using a self-signed certificate. During $S2$ we observed 1,812 OPC UA servers, of which 1,203 exposed one or more endpoints, and 534 allowed us to authenticate and browse through their nodes. Of the servers with endpoints, 1,006 did not offer encryption to establish a secure connection, and 728 allowed anonymous sessions. Furthermore, 533 advertised deprecated security policies, which, compared to $S1$, is a decrease of nearly 50% servers with endpoints advertising these policies. Figure 4.5 shows the correlation distribution between security modes, and combinations of authentication and policies across endpoints in $S2$ (our $S1$ probe only captured information from a single endpoint, making the com-

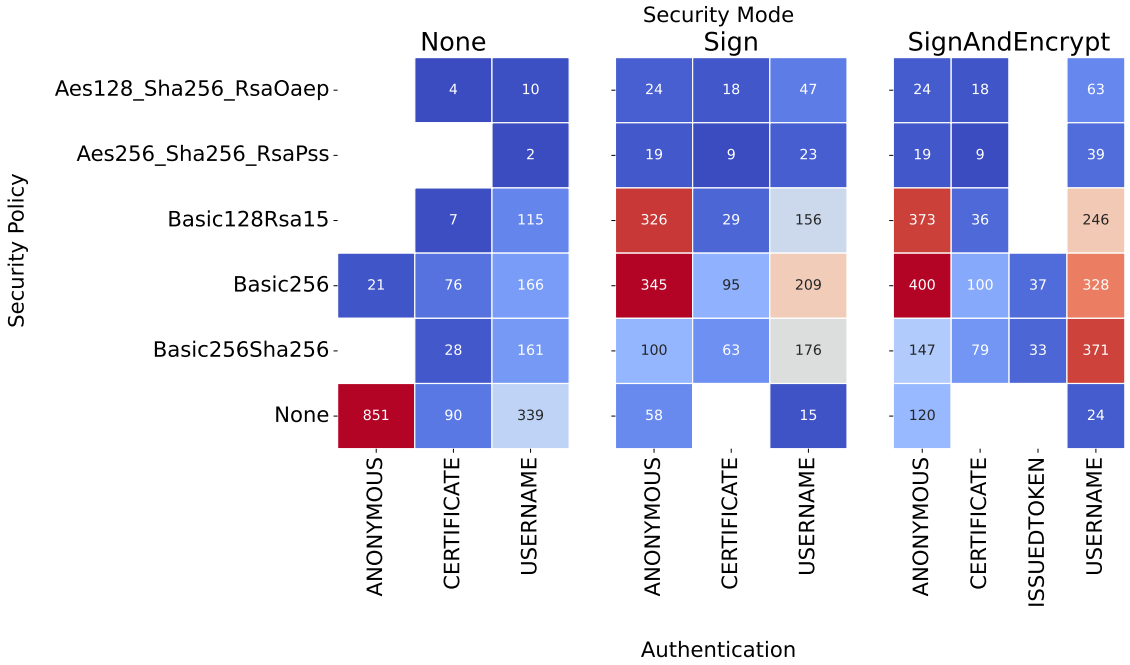


Figure 4.5: $S2$ dataset OPC UA distribution of security modes, and combinations of authentication methods with encryption policies.

parison problematic). As the figure shows, the majority of endpoints support insecure combinations, with most accepting unencrypted channels and anonymous clients. It is worth mentioning that the intersection between datasets contains 427 servers. However, the results are mixed: while we observed a few more OPC UA servers and many have stopped supporting deprecated policies, we authenticated into three times more endpoints using the same method. In addition, the number of OPC UA servers supporting insecure connections remains unchanged. OPC UA server administrators should realize that the security modes `None` and `Sign` are unsuitable for Internet communications, as these do not encrypt communications.

Regarding their certificates, our datasets include 841 ($S1$) and 717 ($S2$) unique certificates from 70 different signers, most of which belong to manufacturers specialized in industrial controllers. Figure 4.6 shows the validity of the unique certificates we collected across server endpoints, ranging from 2019 to 2050 (95% of the values), showing more than 100 expired certificates in both datasets (in red) and the number of reuses for each certificate on top. Note that $S1$ only collected a single certificate per server, while $S2$ includes the certificates from all advertised endpoints. This highlights the importance of collecting all certificates before concluding on the state of the server; as shown on the right side of the figure, most servers reuse the same certificate for multiple endpoints. In addition, our results indicate that approximately 50% of the certificates are being reused across multiple servers. Moreover, the median duration of the certificates we observed is 5 years, similar to the default recommendations from most OPC UA implementations. However, 25% of the certificates violate this recommendation with validity durations between 20 and 50 years.

Takeaway: Allowing non-trusted sources to authenticate into UA servers seriously violates the minimum requirements for access control [37]. Furthermore, maintainers must remove support for weak authentication and encryption methods. In addition, reusing

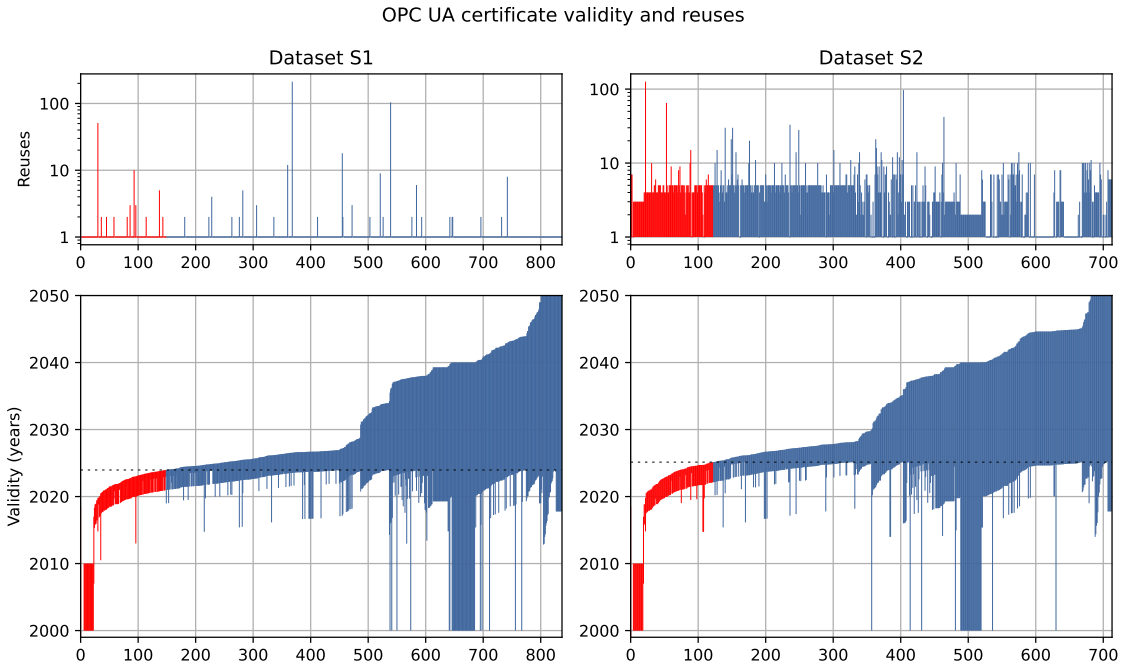


Figure 4.6: Validity of OPC UA certificates and reuses. On the top, is the count of reuses for each certificate. On the bottom, the validity ranges for each certificate. Expired certificates are represented in red.

TLS certificates across multiple servers increases the attack surface, putting at risk all servers sharing the certificate when one of them is compromised. Lastly, servers with expired certificates or valid for the past 5 years are no longer considered secure or valid for cryptographic operations.

4.4.6 RTPS

RTPS is a publish-subscribe protocol used in real-time communications between distributed systems. RTPS is the wire protocol designed for DDS, allowing implementations from different vendors to interoperate seamlessly. This protocol is mainly used in industrial automation systems, smart grids, and other OT applications. Our probe uses the built-in discovery request to enumerate endpoints included in the protocol specifications to retrieve banner information before authentication (e.g., vendor and version). Note that we choose not to join nodes [1] as participants. Services responding with vendor information and application data other than authorization errors are classified as neglected.

Our $S1$ scan yielded 232 responses with known vendor and product information. This dataset contains products from 8 different vendors, where `OpenSplice DDS` dominates the distribution with 118 servers using specification $v2.1$, followed by 63 FastRTPS servers using $v2.3$. Given the protocol specification versions are interoperable and imply only age, features, and open issues, we are not surprised that none of the nodes adopted the latest version ($v2.5$). Furthermore, we analyzed the combinations of protocol versions and products to identify potential issues. For example, `RTI Connext DDS` introduced support for RTPS DDS $v2.2$ in version 5.2 (released in 2015). We estimate that most nodes supporting $v2.1$ run on deprecated products, risking their integrity and participants. The most notorious vulnerabilities range from DoS to various overflows causing crashes. On a last

[1] Distributed systems use the term *node* referring to participant devices.

note, we noticed that 167 nodes continued sending packets to our scanner for at least two hours, ignoring multiple flags included in our probe. This is consistent with [68], which reported similar behaviors. Dataset $S2$ included 242 responses, however, this time around only 58 disclosed their implementation details, and the rest responded with various error messages. It is worth noting that all servers found during $S1$ also appeared during $S2$, which indicates that most of these devices have received updates to reduce the amount of information unknown clients can derive, or to deny access.

Takeaway: RTPS services exposed to the Internet that communicate with unauthenticated participants lack the basic governance required for these systems. For example, we found several devices to monitor and control railways and other critical systems. The severity of this issue is further aggravated in cases where non-trusted participants can read or change topics.

4.4.7 DNP3

This domain-specific protocol is used in SCADA systems to relay messages between masters and slaves (*outstation controllers*). Unlike other SCADA protocols, DNP3 SAv6 (an extension of this protocol) supports multiple security features, such as authentication and encryption [72]. Our probe targets outstations with disabled security features, requesting the status of the first 100 physical addresses to find linked devices. We expect vulnerable outstations to respond with the status of at least one linked device. Outstations responding to our probe with the status of one or more linked devices are classified as neglected.

Our first scan $S1$ contains responses from 697 outstations, of which 399 responded with frames for one or more links. Then, in $S2$ we received 191 responses, but only 58 seem to have linked devices. The intersection between datasets reveals that 62 of these hosts appeared in both, and all except 4 remained unchanged since we found them during $S1$. Although we collected a variety of responses, these can be categorized as a set of combinations, with 19 different combinations in $S1$ and 5 in $S2$. We define these combinations as the set of frames we received from a single target. DNP3 outstations respond to our probes with an array of frames, including the direction of the communication (from master to slave or vice-versa), and a function code answering our request for the status of a link. This means that from all responses we received during $S1$, for example, we only received 19 unique sets of frames. From these, we notice an atypical behavior, a combination that reappeared 250 ($S1$) and 126 ($S2$) times. This combination contained identical payloads to our probe, a behavior not described in the protocol standard; therefore, we labeled these records as echo responses and did not consider them for our classification. Furthermore, the most common combination contained the status of a single linked device, and while varying on the index of the link, most came from index 1. We observed this combination 325 times during $S1$ and 55 during $S2$. Another common combination included error messages for all of the requested links, which is an uncommon response to a health status request, a behavior only observed during $S1$. Of all outstations, few responded with link statuses for 2 to 12 linked devices, 9 during $S1$ and 1 during $S2$.

Takeaway: Outstations must implement access control policies and refuse to communicate with unauthorized users. Some outstations mitigate this threat by responding with error messages such as `UNCONFIRMED_USER_DATA`, or `UNKNOWN`, while others respond with `NOT_SUPPORTED` messages for each requested link. However, responding to unauthorized requests with comparable payloads consumes more resources than responding with a single error message. Furthermore, DNP3-SA should be used instead in outstations facing the Internet, supporting authentication and encryption capabilities.

4.4.8 BACnet

BACnet is primarily used in building automation and sensor monitoring systems. This protocol uses a client-server architecture, where clients can specify queries to read or write values. Some of the readable values include vendor description, software details, and device model. BACnet includes an addendum supporting encryption, authentication, and authorization mechanisms called BACnet/SC (BACnet Secure Connect); however, we are interested in finding devices without those capabilities or disabled. Therefore, our probe targets legacy BACnet nodes running on UDP sockets, which do not support the newer BACnet/SC implementation. Devices disclosing internal information are classified as obsolete since this BACnet implementation is unsuitable for Internet communications. In addition, devices with outdated firmware versions are classified as neglected or abandoned.

Scan $S1$ contains responses from 8,671 Internet-facing BACnet nodes from 138 vendors and 570 unique products. In contrast, during $S2$ we found 4,866 nodes distributed across 116 vendors and 441 products. Although this shows a diverse BACnet ecosystem, their distribution is largely concentrated around few products, with most devices in both datasets identified as Tridium Niagara4 Stations (2,021 in $S1$ and 1,228 in $S2$). This is important since both datasets have a major overlap of 3,228 nodes, of which 2,929 remained unchanged, and 728 were outdated Niagara4 Stations. However, the remaining devices in the intersection (with or without changes) suffer from similar issues, most of which use deprecated software, and few are decommissioned devices. Similar behaviors to other protocols were also observed, with major downgrades that put these devices at an even greater risk. However, unlike other OT protocols, we observed multiple device replacements and upgrades. Though the problem remains, BACnet does not offer security, and devices facing the Internet are open to attacks. This is particularly worrying as BACnet devices include description and location fields, often indicating the purpose and physical location of the device. Some examples include chlorine gas sensors (e.g., for indoor swimming pools), and centralized cooling systems.

Furthermore, Figure 4.7 shows the distribution of the major vendors and products found during our scans. Notably, Tridium's Niagara 4 Station monitoring software makes up a substantial part of our dataset, accounting for 2,021 $S1$ observations, alongside 417 Niagara AX stations (deprecated). From that count, 439 Niagara 4 stations are vulnerable to denial-of-service and cross-site scripting (XSS) attacks, and a few contain broken access control issues. Following closely, we identified various building automation controllers, such as 426 Delta Controls `eBMGR` and 406 `JCI MS-NCE2506-0` controllers.

Takeaway: Without built-in security measures to protect BACnet communications, controllers and monitoring systems depend on their infrastructure to prevent exposure to the Internet [73]. Some manufacturers instruct the use of a VPN for all standard BACnet communications. The BACnet/SC addendum should be considered otherwise.

4.4.9 Summary

While the number of exposed services remains similar from one scan to another for most protocols, we observed an overall steep decrease in the number of DGSs. However, OPC UA DGSs are slowly rising, which calls for further attention as one of the most interesting new technologies in the OT space. OPC UA offers the security properties that others lack, though, as seen in our results, there are many OPC UA servers improperly configured. In addition, a likely explanation to the overall decrease of DGSs can be attributed to our choice of reusing the same vantage point for our scans, which is an important element for the visibility of the scanner and how external networks perceive our probes [9], [67].

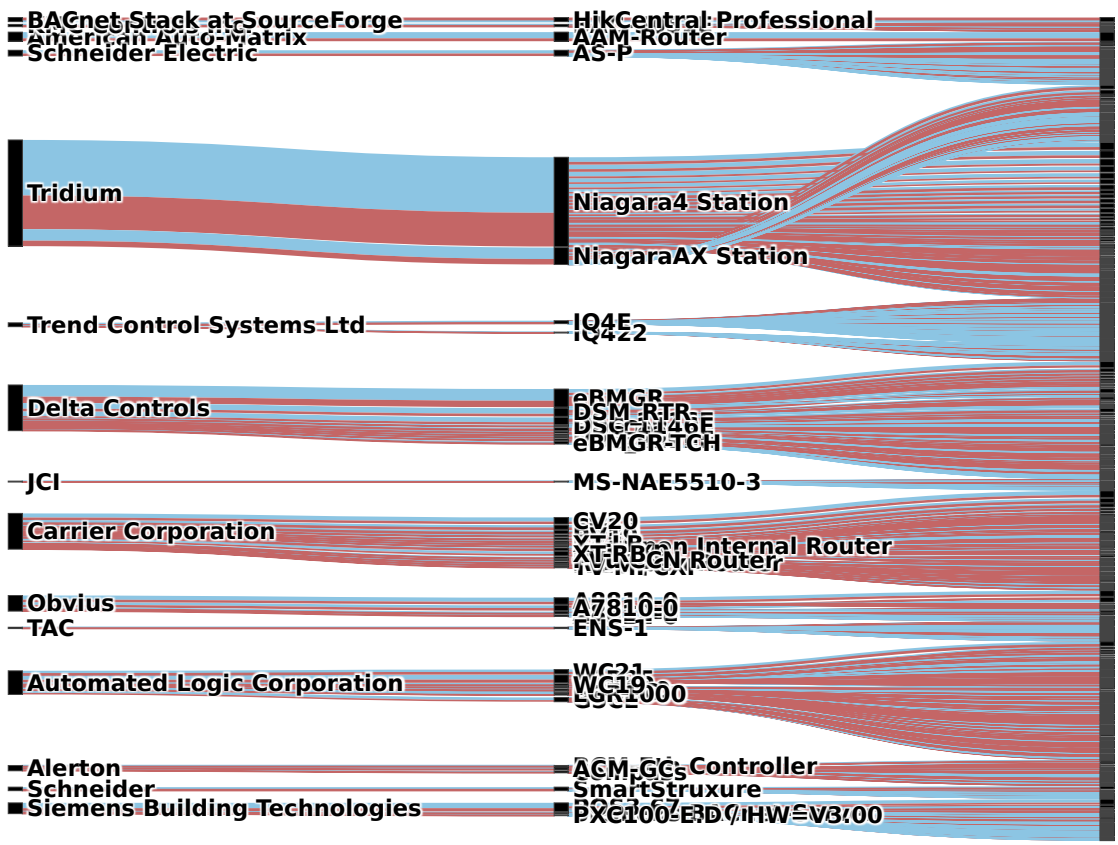


Figure 4.7: Top vendor distribution of products and their firmware versions exposing BACnet services on the Internet.

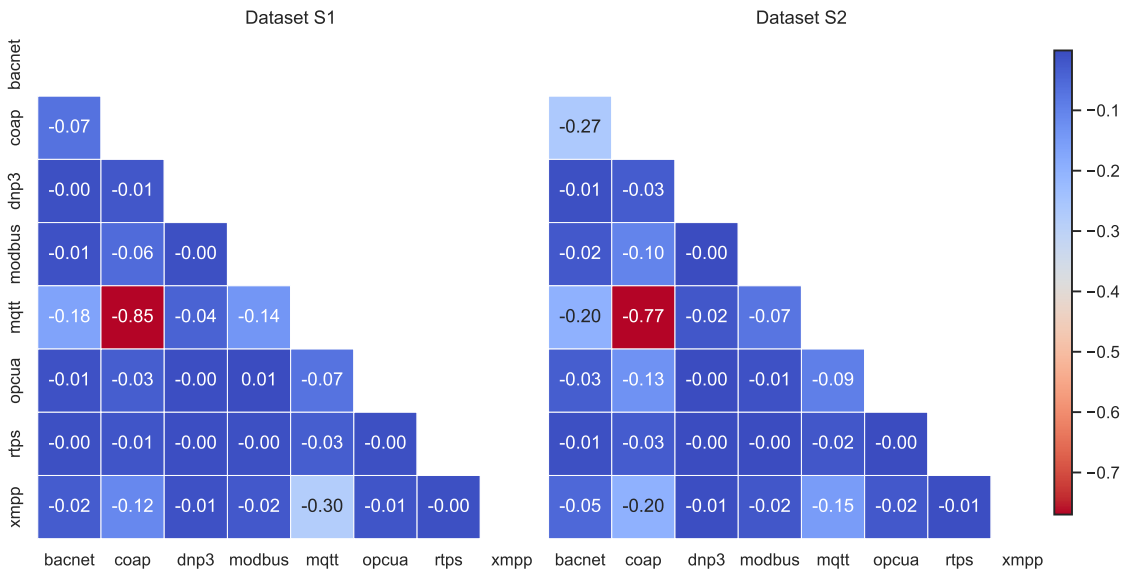


Figure 4.8: Correlation of protocols exposed in hosts.

This factor can be applied as a weak heuristic to detect abandoned devices and identify genuine DGSs, as devices that actively block unsolicited or suspicious traffic no longer qualify as DGSs. Regardless, our findings show that most DGSs appearing in $S2$ were also detected in $S1$, highlighting a considerable overlap between datasets. From these overlaps, we could see that most devices remained unchanged, while others received minor updates and in rare cases downgrades. This is an ongoing problem, as the most persistent DGSs remain unattended for years.

Lastly, the only strong correlation we found between protocols was a negative one with MQTT and CoAP of 80%, suggesting that it is highly unlikely to find hosts exposing both protocols simultaneously. Figure 4.8 shows the correlations between exposed protocols in hosts; these correlations are weak but negative, implying that the covered protocols are often exposed on their own (the correlation indicates the likelihood of finding another of these protocols in the same host).

4.5 Discussion

In this section we discuss our results at the host level, unifying the individual DGSs we found across the various protocols under the same IP. First, we give a brief overview of our results from the view of IP reputation services to identify DGSs previously seen conducting suspicious or malicious activities over the Internet. Then, we report on the results from our vulnerability disclosure campaign to notify owners of DGSs in Denmark. Lastly, we summarize our findings and lessons learned.

4.5.1 IP reputation

We query Greynoise with the addresses of the devices we classify as DGSs to find those seen scanning or attacking the Internet. Greynoise runs a large network of scattered sensors to capture and analyze suspicious traffic. From the $S1$ results we classify as DGSs, Greynoise reported 7,424 of them and tagged 1,244 addresses as malicious. Most of these addresses were seen scanning for exposed SSH and telnet services or distributing malware. As for $S2$, Greynoise had seen 792 addresses, classifying 210 as malicious. From this count, 138 were hosts exposing MQTT brokers, followed by 50 CoAP brokers, 14 BACnet outstations, 4 Modbus servers, 3 OPC UA, and 2 XMPP. Note that DGS hosts

may expose other vulnerable services besides the ones we target with our probes.

We distinguish several common high-risk factors that may have been used to breach the reported hosts. First, from the reported hosts exposing XMPP servers, most were found without encryption or authentication enabled or supporting deprecated authentication methods. In addition, several OPC UA servers did not contain any form of encryption or authentication, with some supporting insecure authentication methods. Furthermore, reported hosts exposing MQTT brokers used deprecated versions with critical vulnerabilities. The distribution of issues among vendors and products is evenly spread for BACnet and Modbus devices, showing that their infrastructure plays a crucial role in securing devices. These findings align with the worst-case scenarios in our classification, indicating that most automated attacks use brute-force authentication methods and exploit known critical vulnerabilities. However, we do not find hard evidence linking certificate issues to compromised devices.

4.5.2 Vulnerability disclosure

To determine which DGSs are located in Denmark, we first query RIPEstat for ASNs, their routing prefixes, and abuse contacts. RIPEstat offers information for the RIPE NCC, the Regional Internet Registry (RIR) for Europe among others. Then, we filter DGSs addresses within those prefixes and collect their `WHOIS` records, often containing abuse contacts and additional owner information.

For scan $S1$, we were able to inform 30 organizations and ISPs through email following the recommendations in [74], including details such as the IP address, a timestamp, services affected, a description of our approach, and instructions to mitigate their risks. We received 5 responses from various organizations unaware of their devices being exposed to the Internet (mainly OT devices) and responded with very positive feedback. The rest of the responses were from ISPs, who had already contacted their customers regarding exposed services. For $S2$, we contacted 16 organizations and ISPs with similar details. We received one single response from an ISP whose address was assigned to a residential customer and exposed a BACnet server.

From these responses we learned that most addresses were assigned to domestic households and mobile subscriptions, supporting previous findings regarding the precarious state of consumer and manufacturer cybersecurity postures [75], [76], [77]. Other authors raised their concerns regarding notification campaigns and the minimal impact on consumer behavior [40], [74], [78]. Generally, most notifications go unnoticed, ignored, bounce back, or receive automated responses.

4.5.3 Summary

Most of the vulnerabilities covered in this paper were associated with security management issues, putting devices and networks at risk. We observed a general lack of proper access control, from severe cases of OT devices used in building automation, open monitoring systems for oil pipelines, and railway stations that accept anonymous connections, to support center equipment pre-configured to accept insecure authentication methods. These security issues are worsened due to the absence of encryption, where most OT protocols lack these capabilities altogether (e.g., Modbus and BACnet). We see that even though most protocols support encryption, it is often disabled, or the device suffers from certificate management issues, with expired, long-lasting, or reused certificates. In addition, we discovered many certificates using weak encryption methods or short keys, which renders them useless. Some devices come with hardcoded certificates and default configurations which cannot be changed, while others may be unpatched, decommissioned, or obsolete. Overall, manufacturers and consumers approach cyber-security differently

[79], [80]. However, it is their shared responsibility to maintain the security of their devices [76], [81].

Furthermore, we encountered some issues that prevented us from fully assessing the scope of the problem. As such, the numbers presented in this paper are likely conservative. For ethical reasons and to minimize intrusion, we designed our probes to close connections immediately upon receiving the banner, without testing whether we could modify the state of the device. In addition, some self-imposed limitations have impacted our results, e.g., our probes close connections after 30 seconds. In the case of MQTT, our probe only captured 50 topics, which in most cases is sufficient to fingerprint the broker, but is often lacking to determine how these brokers are used in practice. Extending the duration of the connection and removing the limitation to the number of topics could improve the precision by which we can identify and analyze brokers. Similarly, our RTPS probe mimics the behavior of a single device and does not join the nodes to retrieve any topic information.

Moreover, our dataset showed significant differences from Shodan (e.g., small intersections, different values, and size of the datasets). For example, some of the results from Shodan were dated and did not represent the state of the host. Shodan scans the Internet periodically, as opposed to creating a single snapshot of the Internet at a given time. Therefore, it is better suited for longitudinal studies with multiple consecutive scans.

In summary, we have shown that security maintenance issues are not unique to any sector of society in particular, but rather a common challenge. Many devices remain connected to the Internet for long periods despite being decommissioned, vulnerable, or already compromised; nevertheless, whether device owners accept their risks, ignore them or are unaware, remains an open question. While we received positive feedback during our vulnerability disclosure, it falls short to provide a conclusive answer. Further studies are necessary to address how society reacts to security advice and improve its security posture. Moreover, we have shown that these security issues are observable and targetable from the Internet using common tools with minor adjustments. The methodology presented in this paper relies on chaining patterns and filtering rules. However, further work is necessary to identify intricate vulnerabilities.

4.6 Conclusion

Throughout this paper, we presented an overview of the current landscape of IoT and OT devices exposing one or more of the targeted protocols. We identified 618,765 DGSs exposed to the Internet during $S1$ and 75,007 during $S2$. These devices lack security management, such as software updates, proper access control, or encryption mechanisms. A large margin uses deprecated or insecure authentication policies, such as allowing anonymous connections or accepting self-signed certificates. In addition, we find widespread deficiencies in certificate management, such as expired, long-lasting, and reused certificates. Furthermore, we examine the IP reputation of the potentially vulnerable devices and find that Greynoise previously reported 7,424 during $S1$ and 792 in $S2$ addresses, with 1,244 and 210 classified as malicious. Finally, we conducted an ethical disclosure of vulnerable devices discovered in our region. We shared insights on their responding behavior, showing that ISPs are the most active in notifying their customers. However, device owners rarely take action.

5 Drifting away: a cyber-security study of Internet-exposed OPC UA servers

Context and Contributions

This chapter provides a focused case study addressing **RQ1** in the context of a protocol explicitly designed with security in mind. The findings show that protocol design alone is insufficient to prevent exposure when operational practices fail. By contrasting OPC UA deployments with legacy OT protocols, this chapter reinforces the thesis-wide observation that misconfiguration, neglect, and abandonment remain dominant drivers of Internet exposure.

RQ	Contribution
RQ1a	Protocol-specific weaknesses in OT environments
RQ1b	Effectiveness of security-by-design protocols in practice

Related publication

R. Yaben and E. Vasilomanolakis, "Drifting away: A cyber-security study of internet-exposed opc ua servers," in *2025 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2025, pp. 195–202. DOI: 10.1109/EuroSPW67616.2025.00029

Original Abstract

In recent years, OPC UA has risen in popularity as an abstraction technology for legacy protocols used in OT (Operational Technology) and SCADA systems, which often lack the security features required for secure remote communication with devices and sensors. However, deploying secure OPC UA servers is not trivial, and many servers end-up facing the Internet in a vulnerable state. To better understand their security challenges, we conduct an Internet-wide scan of OPC UA servers and evaluate the security properties they implement. Our analysis reveals that 62% of the 1,812 OPC UA servers facing the Internet on port 4840 suffer from various vulnerabilities associated with misconfigurations and abandonment, such as outdated software, broken access control, and certificate management issues. In addition, a comparison of our findings with previous work suggests that 25% of these servers have received either none or minor updates in the past years. This paper offers an overview of common and recurrent security challenges in OPC UA deployments, emphasizing the need for robust security measures to protect these and new servers from the same vulnerabilities.

5.1 Introduction

As one of the few technologies within the OT space following Secure by Design principles [84], OPC UA stands out as a protocol that allows interoperable integration between vendors and products, abstracting legacy protocols lacking security features under a unified standard [85]. OPC UA offers many security features required for today's communications between remote devices, such as encryption, authentication, and granulated access control [86].

However, previous studies show that there are thousands of insecure OPC UA servers exposed to the Internet [12], [37], with common issues such as supporting deprecated

or anonymous authentication methods, (re)using insecure certificates, and allowing untrusted clients to browse freely through the controllers linked to the server. Other authors pointed out that many implementations, manuals, and setup guidelines omit these security features in the first place [86], [87], [88], potentially leading owners to deploy insecure servers without realizing the risks. In addition, system owners do not always follow best security practices and leave their devices unattended for long periods, slowly drifting away from a secure state.

This paper explores today's security state of OPC UA servers exposed to the Internet through an Internet-wide scan on its two default ports: 4840 and 4843 (TLS) [89]. Our study strives to identify common and recurrent security pitfalls, emphasizing issues stemming from poor maintenance and misconfigurations. To this end, we examine certificates, access control mechanisms, and product versioning. Our analysis offers guidance for operators to detect weaknesses and lays the groundwork for refining OPC UA deployment manuals and security recommendations to prevent the proliferation of vulnerable servers. Our main contributions are as follows:

- We conducted an Internet-wide scan for OPC UA servers running on ports 4840 and 4843, finding 1,812 and 299 servers facing the Internet, of which 1,203 exposed one or more endpoints.
- Although the OPC Foundation provides clear security advice, we find 1,122 servers that neglect one or more of the described key points, with severe cases lacking security entirely. In addition, 25% of these servers continue reappearing since first observed in previous studies.
- Internal information from these servers (product and versions) reveals that most monitored devices were significantly outdated, where most severe cases included known vulnerabilities allowing attackers to bypass authentication and execute code in the machines. Among these servers, we found instances that monitor critical infrastructure. Our analysis reveals that 95% of the products used in these servers were not certified by the OPC Foundation. In addition, we identified 8 compromised servers that were seen attacking other networks.

The remainder of this paper is structured as follows. Section 5.2 gives an overview of OPC UA and its security properties. Section 5.3 includes a brief walk-through of the previous work about Internet measurements for OPC UA and other protocols used in OT. Section 5.4 describes our scanning methodology, and ethical considerations and limitations of this study. In Section 5.5 we cover the results of our analysis and principal findings. Then, Section 5.6 discusses the general aspects of our security concerns of OPC UA servers facing the Internet as well as potential future work. Lastly, we conclude this paper Section 5.7 with a brief summary of our work and the key takeaways.

5.2 Background

OPC UA is a rich but complex protocol, supporting multiple architectures (e.g., Pub/-Sub, and Client/Server) and communication methods for diverse products from various manufacturers with unique requirements. Clients can interact with OPC UA servers via SOAP/HTTP (now deprecated), HTTPS, WebSockets, or raw binary encoded messages. Our work focuses on the binary encoded protocol, as it is mandatory for all OPC UA servers.

Figure 5.1 shows how clients establish sessions with OPC UA servers and their endpoints. To find OPC UA endpoints, clients can send discovery requests to OPC UA servers, which

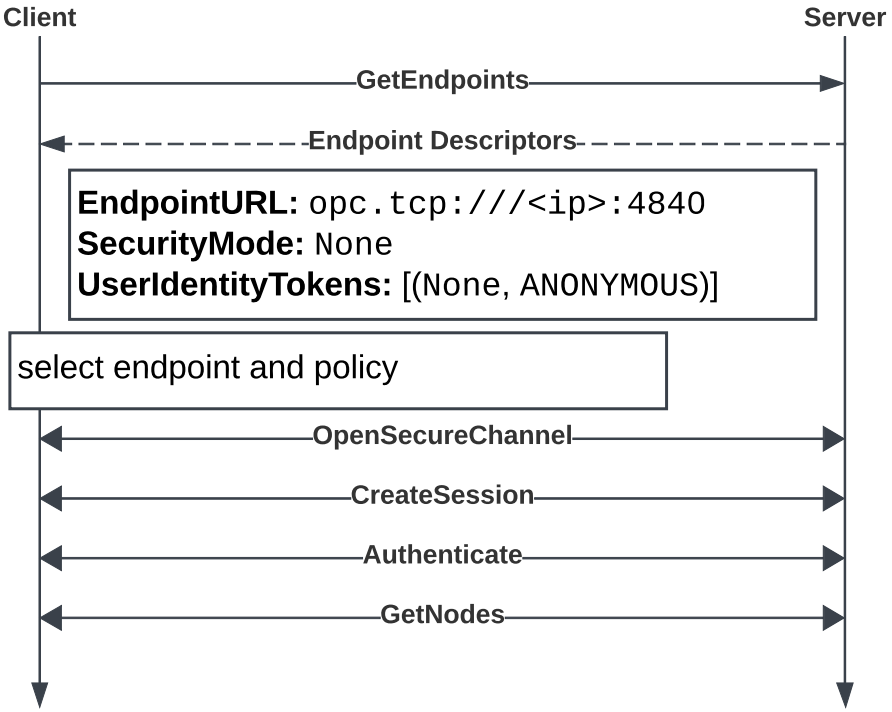


Figure 5.1: Communication steps with OPC UA servers to authenticate and retrieve nodes from endpoints.

will respond with endpoint descriptors (this step is not required and can be skipped when the target endpoint is already known). Each endpoint descriptor includes a `SecurityMode` field indicating the level of confidentiality supported. In addition, descriptors include the field `UserIdentityTokens` with combinations of authentication methods and security policies allowed (`UserTokenPolicy`) – clients must choose one if authentication is required. In cases where the security policy is not included in this field, clients must default to the global value from the descriptor. Security modes and encryption policies are used to establish a communication channel between the client and server; once established, clients authenticate using the advertised method. However, most combinations of authentication and security policies and modes are not safe for Internet communications and should be avoided.

Security modes appear in three flavors: `None`, `Sign`, and `SignAndEncrypt`. Advertising `None` as security mode carries severe security implications, allowing clients to join the endpoint anonymously with an unencrypted session; similarly, `Sign` does not offer confidentiality either and should be disabled in all Internet-facing servers. Furthermore, the OPC UA Foundation has deprecated two policies that rely on `SHA1` for signing: `Basic256` and `Basic128Rsa15`. Endpoints can choose not to offer encryption, which, together with deprecated policies, compromises the session's privacy. In addition, clients can authenticate using one or more of the following methods supported by the endpoint: with an anonymous user, a set of credentials, a token, or an $v3$ X.509 certificate. Anonymous login must be disabled in all cases, as it allows unknown clients to join endpoints without providing any proof of identity [2], [88]. Moreover, endpoints must refuse to authenticate clients with untrusted certificates. In practice, this leaves Internet-exposed OPC UA servers with a few viable options to authenticate clients, as they must discard dangerous authentication methods (1 or 2 out of 4), security modes without confidentiality (2 out of

Table 5.1: List of authentication methods, security policies, and security modes supported by OPC UA endpoints. Insecure options for Internet communications colored in red.

Authentication methods			
Method	Description		
ANONYMOUS	Anonymous access		
USERNAME	Credentials based authentication		
CERTIFICATE	Authentication via X.509 v3 certificates		
ISSUEDTOKEN	Authentication via pre-shared token		
Security modes			
Mode	Description		
None	No security (only used in anonymous profiles)		
Sign	Messages signed, offering integrity		
SignAndEncrypt	Messages signed and encrypted		
Security policies			
Policy	Signing	Encryption	Key Exchange
None	-	-	-
Basic128Rsa15	SHA-1	128-bit RSA+AES-128	RSA-1024
Basic256	SHA-1	AES-256	RSA-2048
Basic256Sha256	SHA-256	AES-256	RSA-2048
Aes128Sha256RsaOaep	SHA-256	AES-128	RSA-2048
Aes256Sha256RsaPss	SHA-256	AES-256	RSA-2048

3), and deprecated or insecure policies (3 out of 6). Table 5.1 summarizes the possible values to authenticate into OPC UA endpoints, with insecure options marked in red. It should be noted that other security policies exist (e.g., supporting elliptic curves), though our dataset does not contain any examples.

After authenticating into endpoints, clients can browse through their nodes. In the case of OPC UA, nodes represent data points and executable functions. It should be noted that endpoints may implement access control measures at the role, user, or node level, allowing only authorized users to read, write, or execute nodes (other users may only see node names and their access level).

5.3 Related Work

The research community's efforts have brought numerous studies covering security issues in OT [2], [90], [91], [92], [93]. The state of the literature is rich and diverse, with many different methods to identify vulnerabilities, propose mitigation strategies, and raise awareness. Notably, Gao et al. [94] spoke about the security issues with SCADA systems exposed to the Internet while proposing multiple mitigation strategies. Then, Mirian et al. [8] analyzed the state of vulnerable Internet-facing ICSs through multiple Internet-wide scans with probes for 10 protocols, in addition to deploying honeypots to identify those conducting similar scans; however, OPC UA is not covered in their work. Moreover, Nawrocki et al. [95] used passive scanning methods to identify ICS traffic through an IPX, focusing on developing new mitigation strategies against malicious activities.

Related work in OPC UA is limited and often part of larger studies. This is expected from emerging technologies with limited adoption and specific use cases. Nevertheless, OPC UA is a technology primarily used in critical systems with higher security requirements, thus needing further attention from the community. Here we highlight four studies tackling security concerns in OPC UA implementations and Internet-exposed services.

The work of Dahlmanns et al. [39] analyzes the use of TLS in industrial IoT systems, covering OPC UA and 10 other protocols. Their results indicate that none of the 2,193 OPC UA servers they found support TLS. In addition, they found no evidence of OPC UA servers exposed on the default port for TLS communications (4843). However, these results are likely an overgeneralization and must be interpreted carefully. There are numerous reasons to refuse connections from unknown clients. Erba et al. [88] takes a different direction in assessing OPC UA security implementations. They evaluate vendor implementations of the protocol in 22 products and 16 libraries, including their manuals and example setups. Their analysis of the implemented security features reveals several issues in all libraries and 15 vendor products.

To date, Dahlmanns et al. [37] is the closest work to our study, which focuses on security issues found in Internet-facing OPC UA servers. They analyzed seven months' worth of weekly Internet-wide scans to uncover insecure OPC UA server implementations. Their analysis covers the distribution of manufacturers and products, security policies and authentication methods, and access control and certificate-related issues. Their results suggest that 92% of the 1,114 OPC UA servers facing the Internet suffered from severe security issues. In addition, the authors analyze the number of servers reappearing throughout their study and carry out a responsible notification campaign. However, they received limited feedback, aligning with concerns expressed in similar studies. Recently, we conducted a similar study covering multiple protocols used in IoT and OT devices, including OPC UA [12]. Our goal was to uncover security issues associated with neglected, obsolete, and abandoned devices (diverging from common vulnerability reports) to identify human errors, misuse, lack of maintenance, and poor security hygiene. We reported a significant increase in OPC UA servers compared to [37], identifying 1,797 exposed servers, of which 1,210 were vulnerable, and 30 showed malicious behavior. However, we used a less intrusive probe; this probe was sufficient to survey the Internet for exposed OPC UA servers, but it limited our evaluation of the issue. Overall, these studies (i.e., [12], [37]) miss valuable insights (e.g., product and version distributions, location, and usage) to determine where and how OPC UA fails.

The rapid developments in OPC UA and their proposed security measures call for new Internet measurements to identify security pitfalls in today's deployments. Previous studies have shown that even certified OPC UA products suffer from multiple security issues [88], [96], such as hardcoded certificates, deprecated authentication mechanisms, and insecure examples in their documentation. Others highlighted the challenges of deploying and maintaining secure OPC UA servers, and the limited feedback received after notifying their owners [12], [37], [39]. In this study, we offer a granulated analysis of OPC UA deployments in the wild, covering security issues across endpoints and nodes beyond what has been examined in the related work. In addition, we give further details on products, versions, and locations to connect particular issues.

5.4 Scanning Methodology and Ethical Considerations

In February 2025, we deployed a scanning campaign to identify OPC UA servers facing the Internet. The campaign targeted two ports commonly used for discovery services: 4840 and 4843 (TLS). Our scans are divided into two phases. First, we use ZMap [50] to identify hosts accepting TCP communications at either of the targeted ports. ZMap is a stateless L4 scanner capable of sweeping the entire IPv4 in a matter of hours using gigabit network interfaces. Then, we scan the responding hosts using the OPC UA probe provided by Dahlmanns et al. [37] (with minor modifications) for Zgrab2 [97], an L7 scanning tool from the ZMap family designed to capture banner information. This probe crawls

OPC UA servers, mapping endpoint applications known to the discovery server and their resources. In addition, the probe attempts to access endpoints with security features disabled using an anonymous user and a self-signed certificate. However, this probe cannot identify OPC UA servers communicating over UDP.

Regarding our experimental setup, we used a single vantage point hosting a website along the scanner with information about our research, scanning methodology, and contact information to opt out of our studies [98]. In addition, we use a blocklist to remove several IP ranges, including local and private networks, reserved spaces, networks that belong to government institutions around the globe, various network telescopes, and those who previously requested to be removed from ours or similar studies, accounting for roughly 25% of the IPv4 address space. Lastly, each connection is limited to 30 seconds per host and includes identifiers to help administrators distinguish traffic from our scanner. Note that this vantage point was recently used in other Internet surveys, which may have impacted our results.

To identify security issues, we focus on three principal aspects of the communication with exposed OPC UA servers: access control, certificates, and device meta-data. We evaluate access control issues based on the depth to which our probe accesses internal resources: first, i) access to one or more endpoints, ii) authentication using anonymous credentials or self-signed certificates, and iii) browsing through the registered nodes and retrieving their values. As pointed out in previous research [12], [37] as well as by the OPC Foundation security guidelines [99], endpoints must implement access control policies and limit communications with untrusted clients. Furthermore, we analyze server certificates for weak cryptography, including short key lengths, re-usage issues, and expired or long-lasting validity periods (while the common recommendation is 1 year, OPC UA defaults to 5 years as of [99], [100], [101]). In addition, we analyze the device meta-data exposed during the communication, such as manufacturers, products, and versions, usage indicators, and implementation details.

5.5 Results

Despite the numerous revisions and security improvements OPC UA has received recently, and the efforts of the OPC Foundation to provide security guidelines [99], the landscape of insecure OPC UA servers facing the Internet has only continued to grow [12], [37]. The OPC Foundation has contributed to CISA's Security by Demand and Secure by Design initiatives [102], which aim to mitigate many security challenges owners face while deploying OT products. However, such initiatives are not categorical requirements, but recommendations and suggestions that often fail to get through. In this section, we analyze the observed OPC UA servers and their security implementations to identify common pitfalls and compare them with the results from previous studies (see Section A for detailed explanations on the evaluation criteria).

5.5.1 OPC UA servers

Our dataset contains responses from 1,812 OPC UA servers facing the Internet on port 4840, and 299 in port 4843. Based on this count, our probe retrieved endpoint information from 1,203 servers on port 4840. The rest responded with various errors, or the connection timed out before accessing any endpoint. On the one hand, these results align with the findings from [37] and [12] with marginal differences. On the other hand, none of the servers found in port 4843 allowed us to communicate past the discovery request. This contradicts the findings in [39], whose results showed no evidence of TLS use on exposed OPC UA servers. This may be due to a limitation in their probe, preventing the authors from determining when legitimate OPC UA servers refuse to communicate. We overcome

Table 5.2: Summary of OPC UA servers facing the Internet, with a breakdown of security issues found on each server. Totals show servers with one or more issues within the same category.

OPC UA Servers	
Exposed servers	2,111
With endpoints	1,203
Authenticated to one or more endpoints	534
Access Control Issues	
Anonymous access	728
Accepts self-signed certificates	53
Endpoints with deprecated policies	533
Total servers with one or more issues:	**902**
Certificate Issues	
Reuses certificates	674
Certificates were expired	156
Certificates were long-lasting (>5 years)	94
Invalid certificates	1
Weak hashing algorithms	182
Short keys	160
Total servers with one or more issues:	**802**

such issues by capturing all traffic during the scan, instead of relying on our probes alone. Table 5.2 summarizes the number of OPC UA servers and their vulnerabilities (1,122 vulnerable servers in total). Furthermore, by comparing our results with the public dataset from Dahlmanns et al. [103] and our previous study's dataset from [12], we could partially verify that nearly 25% of these servers were also observed in the past, having received none or minimal updates since.

5.5.2 Endpoint security

As previously discussed in Section 5.2, each endpoint includes the security mode, and combinations of security policies and authentication methods they support (cf. Section 5.2). Figure 5.2 shows the combinations we observed across all accessible endpoints on each server. The figure shows that most endpoints support other insecure combinations of modes and policies. Although it is recommended to use Basic256Sha256 as the minimum security policy [99], most systems were configured with the weakest option: anonymous access and no security at all. Overall, advertising multiple authentication methods allows clients to select the weaker ones to access the endpoint. When paired with the absence of TLS, methods that offer no security, or deprecated security policies, such as `Basic128Rsa15` and `Basic256`, servers become susceptible to privacy and access control issues, from eavesdropping to an array of attacks bypassing authentication. Our dataset contains multiple examples, with servers advertising certificate-based authentication methods using deprecated policies. Allowing anonymous authentication mechanisms or deprecated policies should be a security concern for all device owners, even in the presence of other security measures. These concerns apply to 902 of the 1,203 servers exposing one or more endpoints, roughly 50% of the OPC UA servers facing the Internet. Supporting these authentication methods defeats the security measures that OPC UA implements. This suggests that their owners choose OPC UA for convenience rather than security purposes, which may be associated with the absence of TLS. It is worth mentioning that, while a significant number of servers advertise just one endpoint (223),

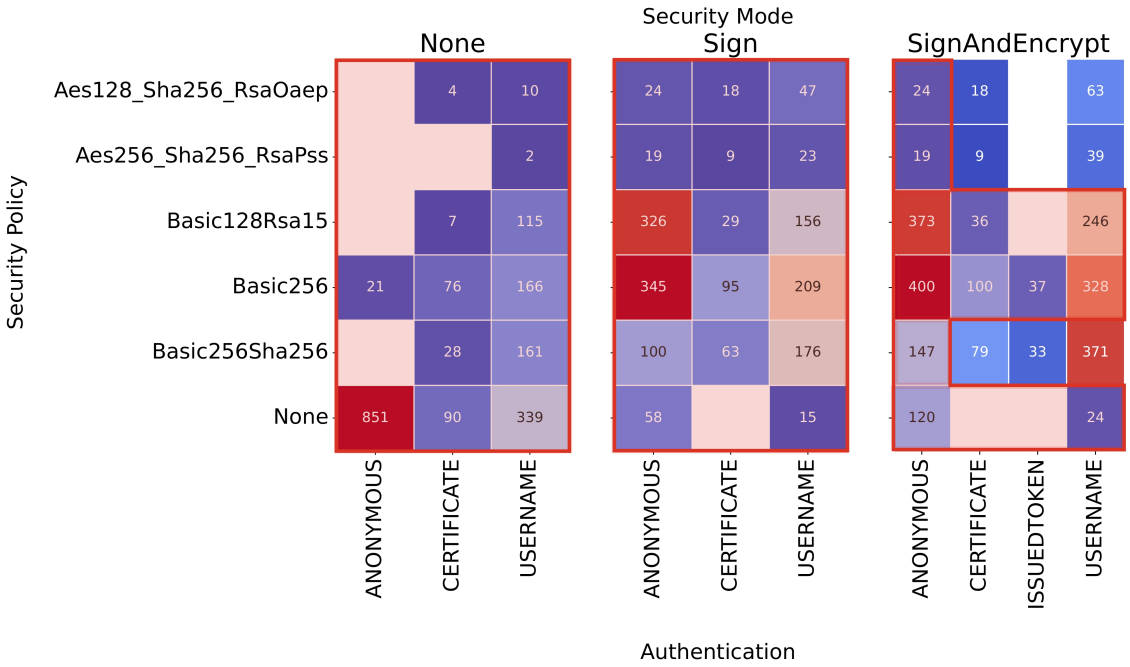


Figure 5.2: Combinations of security modes with authentication methods and security policies used across endpoints. Insecure combinations masked in red.

the majority advertise between 2 and up to 30 different endpoints (see Figure 5.4). For reference, our dataset contains 4,569 different endpoints.

5.5.3 Manufacturers, products, and versions

As depicted in Table 5.2, for 534 servers, we could authenticate to one or more endpoints and crawl their nodes to identify the device manufacturer, product name, and firmware versions. It is important to remember that the level of access to most nodes is limited to reading their descriptors (e.g., node names, data types, and access), and neither read nor write their values. In fact, our dataset does not contain any nodes writable by our user, as opposed to the findings in [37]. However, we still consider it a significant risk to allow unknown clients to gain knowledge of endpoints and nodes. In total, 66% (357) of the endpoints from which we could read nodes disclosed their system information. Figure 5.3 shows the distribution of the most common products, led by Siemens (210), B&R Industrial Automation GmbH (146), and Schneider Electric (105) devices. These devices included multiple control and alarm systems overseeing various tanks, as well as building automation systems. At the other end of the spectrum, we also found systems related to critical infrastructure, such as an oil pipeline (their owners have already been notified). To the best of our knowledge, none of the manufacturers had ceased operations, nor products we found were deprecated, which is a common issue with OT devices. With few exceptions, OPC UA servers and those with the same Fully Qualified Domain Name (FQDN) use the same type of product for all endpoints. On the other hand, more than 95% of these servers use non-certified products. Moreover, we highlight multiple old versions that these devices are running on: with build dates ranging between 2011 to 2025, with quantiles at 2016 (25%) and 2021 (75%), and median at 2019. Several products contain known vulnerabilities, enabling attackers to bypass authentication entirely, deplete resources, or cause DoS (Denial of Service). For example, some servers were running KEPServerEX in a deprecated version, allowing attackers to crash the server and

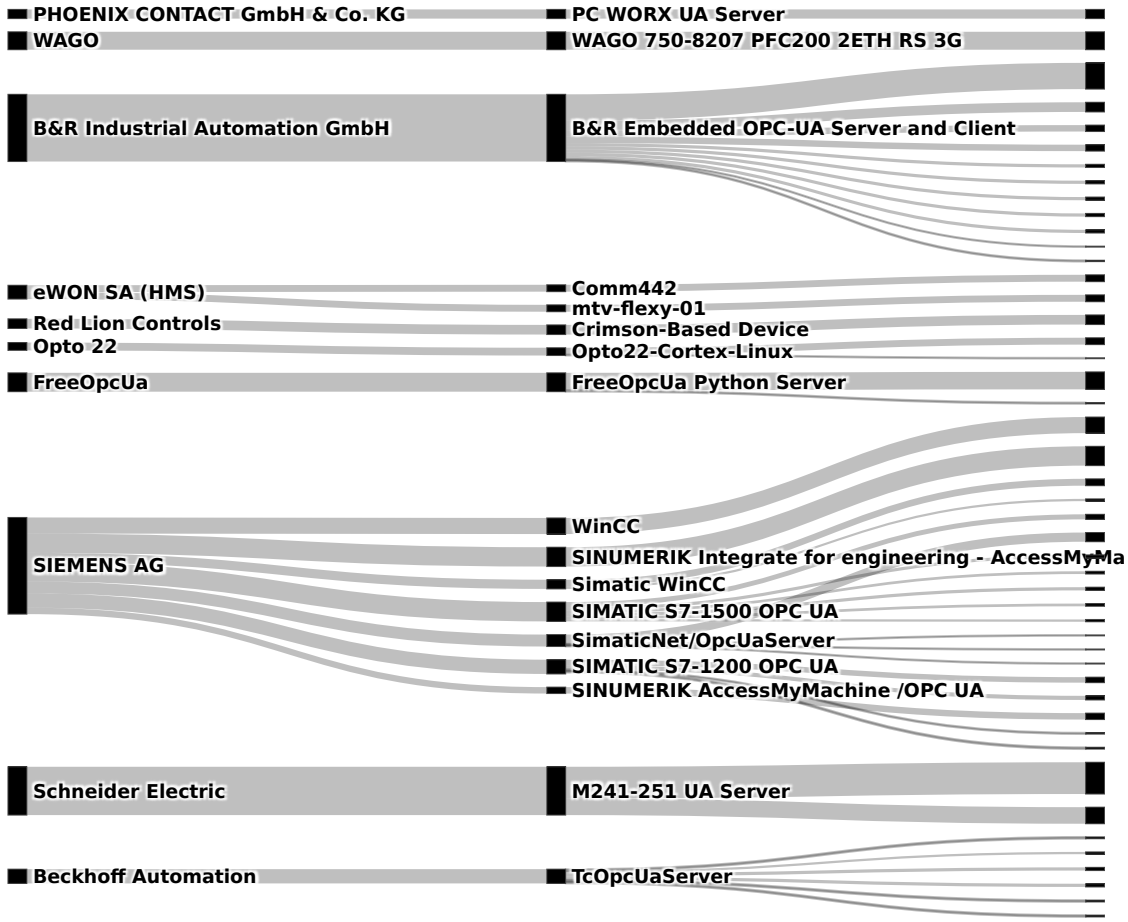


Figure 5.3: Distribution of the most common manufacturers and products, and their firmware versions.

remotely execute code [1].

In many cases, risk factors accumulate, leading to a heightened state of vulnerability. OPC UA servers with several issues, such as broken access control and legacy builds, show signs of abandonment, a predictive aspect of devices potentially compromised [12]. To explore this further, we gather additional information on their origin and ownership using the RIPE Atlas service [82], while leveraging AbuseIPDB [104] and Greynoise [57] to identify hosts engaged in unsolicited traffic. From these records we could see that the majority of servers were located in China (318), the United States (242), Germany (182), and South Korea (93), with the major providers being Alibaba (170), Akami Cloud (169), Deutsche Telekom (99) and Korea Telecom (40); this suggests that most OPC UA servers exposed to the Internet are routed through cloud services, which may be an important factor to consider when suggesting implementations and best security practices while deploying OPC UA servers.

Since January 2025, AbuseIPDB and Greynoise have flagged 50 hosts running OPC UA servers for suspicious activity. Of these, 8 were classified as malicious, engaging in brute-force attacks on Telnet and SMB — behavior commonly associated with Mirai-infected devices. Unsurprisingly, their OPC UA servers lacked any security measures.

[1] https://www.cve.org/CVERecord?id=CVE-2020-27265

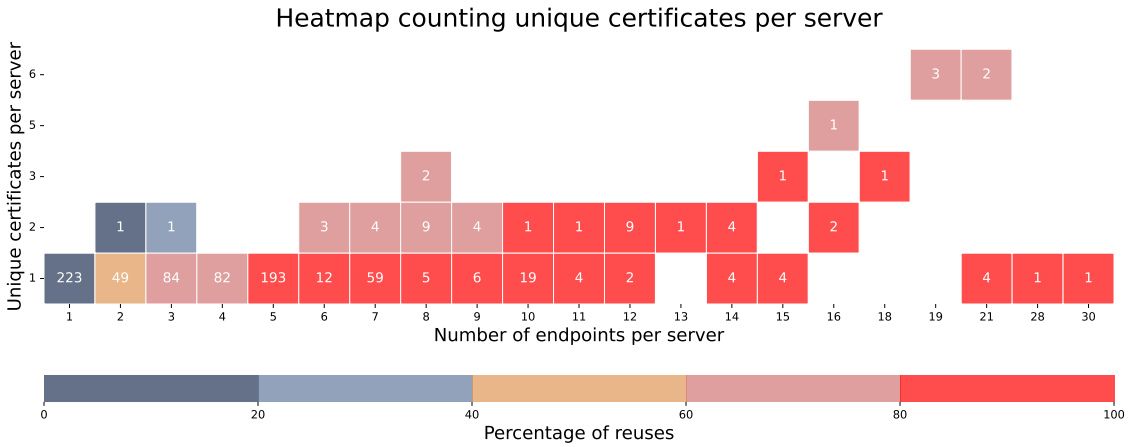


Figure 5.4: Certificate reuses across endpoints per server. Counts represent servers, and colors the ratio of certificate resuses within the server.

5.5.4 Certificate management

Out of the 1,203 servers exposing endpoints, we could retrieve endpoint certificates from 802. As for the rest of the servers, these only allowed credential-based authentication methods (i.e., anonymous, username, or tokens) without encryption. A major recurring issue we observed was the reuse of certificates across endpoints within the same server, affecting 72% (578) of the servers. Figure 5.4 shows the distribution of endpoints per server with certificates, where the colors represent the ratio of reuses within the same server. This figure depicts a daring landscape, with severe cases of servers exposing more than 20 different endpoints with the same certificate. While reusing a certificate across multiple machines may be convenient, it carries significant risk—if one certificate is compromised, the entire infrastructure is at stake.

Besides reuses, 156 servers had endpoints with expired certificates, and 94 with long-lasting certificates that may never expire (> 50 years, the previous default value of auto-generated certificates). Expired and long-lasting certificates are not categorically vulnerable, but a hint at the level of maintenance these servers receive. In the device's security lifecycle, renewing certificates helps owners verify the device's health and maintain its reliability for the duration of the certificate or until it is revoked. As we observed these issues in tandem with broken access control and reuses across servers within the same FQDN, these organizations likely struggle to configure OPC UA servers and implement security hygiene strategies. Another possible explanation is that these devices were configured by default with hardcoded certificates. However, our dataset contains multiple instances of other servers using the same devices but different configurations (e.g., versions and certificates). Figure 5.5 summarizes the certificates we found across endpoints and servers, showing the validity period for each unique certificate we found (bottom), and the number of reuses (top), with expired certificates colored in red.

Moreover, 182 servers were found with endpoints using certificates with deprecated signing algorithms relying on `SHA1`. In all cases, encryption is handled by `RSA`, while signing is done by either `SHA1` (897), `SHA256` (2,542), or `SHA512` (5). This is a reduction of almost 50% over the results from [37], suggesting there is some level of improvement. Additionally, 739 certificates with `SHA1` use 1024-bit keys, and the same problem appears in 69 certificates with `SHA256`. The standard recommendation suggests 2048-bit keys since 2015, when 1024-bit keys were officially phased out. Despite other issues, 2048-bit was the most common key length among certificates, with some instances reaching over 3072

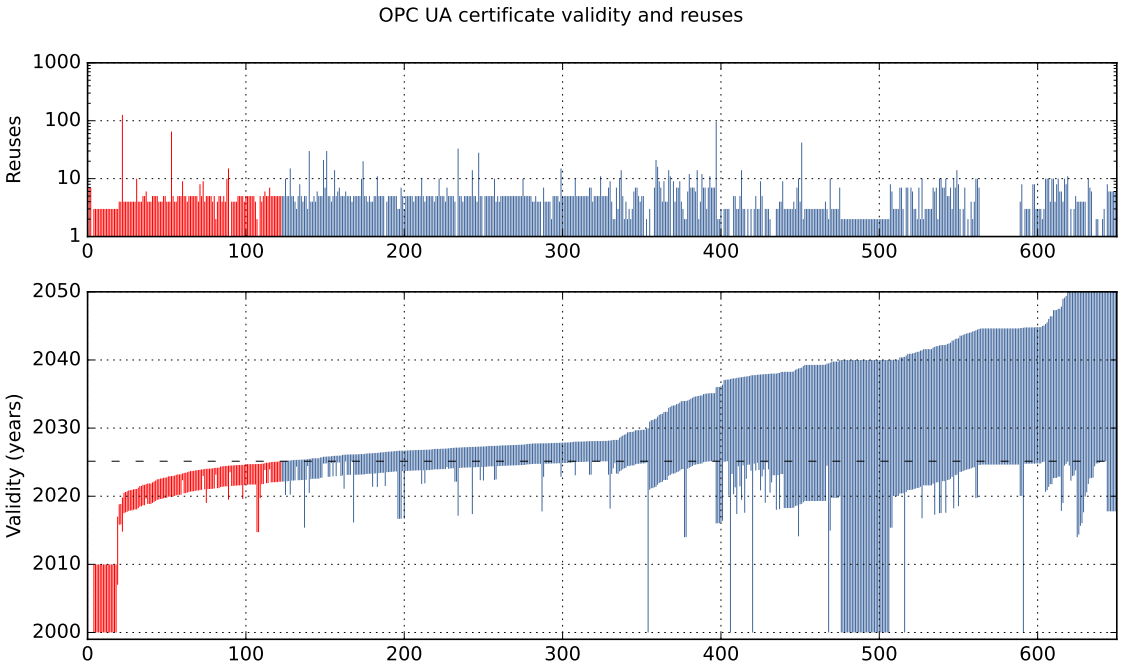


Figure 5.5: Certificate reuses (top) and validity periods (bottom). Expired certificates are colored in red.

and 4096-bit keys.

5.6 Discussion

The majority of security concerns we identified throughout this paper were related to access control issues, security maintenance, and certificate management, aligning with the findings of related work [12], [37]. Our findings show that OPC UA servers communicating over TLS appear secure for the most part (i.e., refusing to communicate), but those lacking it face a multitude of vulnerabilities.

On the issue of access control, allowing unauthorized users to access internal resources is a non-negligible risk, even when these are protected with policies preventing reads, writes, and executing functions. From names and locations, attackers can easily estimate the value of their target and get an understanding of the infrastructure running internally. This was evident in servers that allowed unrestricted browsing of endpoints and nodes, enabling us to determine build versions, products, and in many cases, the industries these servers were monitoring. However, those implementing certificate-based authentication methods are not exempt from issues either. As shown, the second method we used for authentication was self-signed certificates. OPC UA endpoints must disallow clients to authenticate with untrusted certificates, and remove deprecated authentication methods based on `SHA1`. Furthermore, a large fraction of the devices we identified were outdated and vulnerable, and many had certificate-related issues. While updates and renewals may offer a temporary fix, long-term risk mitigation requires additional measures. These vulnerabilities often stem from poor security practices, highlighting the need for better training and awareness.

Regardless of these issues, we remain optimistic about the OPC UA landscape. The OPC UA Foundation already provides clear security recommendations, such as regular updates, certificate management, and secure deployment practices. In addition, the re-

search community continues to point out the remaining challenges observed in Internet-facing servers, manuals, guidelines, and products. To expand on these efforts, we recommend further guidelines for secure deployments in cloud services and provisioning strategies for remote controllers.

In terms of future work, while past attempts to reach device owners have seen limited success, we believe this remains an essential matter, independent of feedback. Therefore, we will continue this practice and notify those whose servers appear drifting away. Lastly, we did not find any measurements covering the state of OPC Classic servers exposed to the Internet, which may be an interesting direction for future studies.

5.7 Conclusion

As one of the few protocols within OT offering security and privacy out of the box, OPC UA soars among the competition to create safe environments accessible from remote locations. Moreover, OPC UA stands out for its approach to abstract legacy protocols instead of replacing them, allowing system owners to transition to this new technology as it becomes available in more products. However, deploying OPC UA servers is not trivial, as system owners frequently struggle to configure their servers and maintain them secure. Failing to follow the security guidelines offered by the OPC UA Foundation may lead to severe security risks [99], [105], an issue that is only worsened without proper security maintenance.

This paper seeks to identify common and recurring security challenges within OPC UA servers facing the Internet. From the results of our Internet-wide scan, we identified 1,812 OPC UA servers exposed on the default port (4840). Our analysis shows that 1,203 of these servers advertise up to 30 different endpoints, of which 1,122 suffered from one or more issues related to access control or certificate management. In addition, we show that we could authenticate and browse through endpoint nodes in 534 servers using anonymous credentials or self-signed certificates. From these nodes, we could identify most products, versions, and implementation details, and we show that a significant fraction of them were outdated and had known vulnerabilities, with 8 of these hosts seen attacking other networks. We also explained our reasons for considering these issues as vulnerabilities and proposed immediate fixes (e.g., patches and removing insecure authentication methods) and long-term mitigation (e.g., security training and awareness). In addition, our analysis suggests that these servers are mostly located in cloud providers, thus, further guidelines for cloud deployments seem necessary. Lastly, we compared our results with those from previous authors, showing that more than 25% of the servers continue to reappear across datasets and through the years.

Appendix

A Evaluation criteria

For completion and to assist in reproducing this study, this section includes further details in our identification and classification process, as well as a breakdown of the security issues covered in this paper.

It is important to mention that Internet exposure is not a vulnerability in itself, but it increases the attack surface of the working system and its risks. However, determining whether exposed services are insecure is a delicate task that requires careful planning and a well-defined ethical process. Mishandling probes and scans can lead to unintended Denial of Service (DoS), privacy breaches, and other severe issues. Therefore, researchers conducting Internet surveys of these characteristics must state beforehand the goal of the study, settle on reasonable levels of intrusion and Internet noise, and consider the level of detail included in their publications. Internet measurements such as this one are especially sensible, since they cover widespread issues that can only be fixed with human intervention and considerable individual efforts.

Previous measurements covering OPC UA highlighted the issue of thousands of servers exposed to the Internet without support for TLS [12], [37], [39]. However, they do not offer further insights regarding the devices themselves, such as their type, model, location, or firmware version. These details can help us in better understanding their risks and form an impression on whether these issues are localized (e.g., to one manufacturer, geographical location, community, etc.) or common to all. Part of this information is often available within many endpoints to help operators manage their assets. However, unauthorized access to this information may also lead to further attacks. Therefore, our probe must test for access control issues and attempt to access unprotected internal information. Our previous probe used in [12] severed connections immediately after testing for authentication issues, without browsing nodes or testing access control levels. On the other hand, the probe from Dahlmanns et al. [37] required minimal modifications to handle this corner-case. With our changes, the probe handles servers requiring TLS, captures certificates, iterates endpoints advertised by discovery servers, attempts to authenticate using an anonymous guest user and a self-signed certificate, and browses nodes at particular indexes (instead of exhausting all nodes, which may be in the thousands of requests). This probe allows us to expand on the literature and continue the work we presented in [12], covering misconfigurations and issues associated with human behavior.

The security vulnerabilities we discuss in this paper are tightly linked to (not following) the security recommendations from the OPC UA foundation on secure server deployments and maintenance. In summary, these recommendations guide operators to meet their security goals, covering certificate management, authentication, and access control – the same we cover here. Table 5.3 includes a summary of the criteria we use to classify OPC UA servers as vulnerable or not. The remainder of this section covers each criterion individually.

Certificate management. To detect certificate management issues, we first collect all certificates from all endpoints and servers, then cross-reference them to test for reuse. This process reveals reuse within and across servers globally, potentially uncovering unintended connections and severe issues like hardcoded certificates. However, we do not

Table 5.3: Evaluation criteria to identify vulnerable OPC UA servers exposed to the Internet

Category	Label	Description
Certificate management	Expired	Certificate validity period expired before date of scan
	Negative	Expiration date is before starting date
	Long-lasting	Certificate validity period longer than 5 years
	Weak hash	Uses deprecated and insecure hashing algorithms (MD5, SHA1 or DSA)
	Weak encryption	Uses deprecated and insecure encryption algorithms
	Short key	Uses a public key below 2048 bits length
	Reused	The certificate appears in other systems
Authentication	Anonymous access	Probe can authenticate using empty credentials
	Self-signed certificate	Probe can authenticate using a self-signed certificate
	Weak security policy	Endpoint accepts deprecated security policies (Basic256 or Basic128Rsa15)
	Weak security mode	Endpoint does not offer privacy and integrity (modes None and Sign)
Access control	Read nodes	Probe can browse through nodes
	Write nodes	Nodes explicitly state the user has write access
	Execute nodes	Nodes explicitly state the user can execute it as a function
	Leak internal information	Nodes leak internal or sensitive information (e.g., state, implementation details, measurements)

identify already compromised certificates. We also evaluate certificate validity periods against the default five years and match them to the scan date, helping identify security negligence such as expired or overly long certificates. Finally, we assess cryptographic properties to detect weak algorithms, short key lengths, and reuse.

Authentication. OPC UA servers exposed to the Internet must implement some form of authentication. Therefore, we attempt to authenticate into endpoints supporting none or weak security policies and authentication methods. Our probe uses either an empty username and password combination to log in as a guest anonymous user, or a self-signed certificate. Both of these options must be disabled in such servers and refuse to communicate with unauthorized clients. During this process, we also evaluate the security modes used to establish the secure channel, i.e., whether the endpoint offers integrity and confidentiality once authenticated, and which algorithm is used to create such a session.

Access control. Once authenticated, our goal is to determine the level of privileges allowed. For this, we crawl specific node indexes and collect their descriptors and values (e.g., state and device information). This process allows us to fingerprint endpoints and profile their environment, which may be beneficial for correlating issues with particular devices or sectors. Nodes typically indicate their type and whether they are writable. In addition, observable nodes explicitly state whether the current user has the right to write, read, or execute them. Our probe should not be allowed to browse nodes, collect names, descriptions, or values.

Advancing Internet Measurement Methodology and Mitigation Strategies

Overview

This part aims to address **RQ2** by introducing a new identification and classification engine to deploy advanced scanning campaigns and conduct complex Internet measurements. The chapters in this part dive into the limitations dragging Internet measurements. In particular, the studies included in this part aim to mitigate limitations large-scale measurements targeting OT networks. By focusing on alternative paths, these studies complement the literature on how to conduct active probing and nuances particular to monitoring OT networks. These studies highlight the need for further developments in Internet measurements to include state-of-the-art scanning techniques and evaluation methods, aiming to increase accuracy and coverage.

Similarly to Part I, the following chapters are presented with their original abstracts for reference and contextual discussions linking each contribution to the broader research questions of this thesis.

6 Rolling the DICE: A Device Identification and Classification Engine to detect vulnerable devices facing the Internet

Context and Contributions

This chapter demonstrates how Internet-facing IoT and OT devices can be systematically identified, classified, and analyzed. It introduces a modular Device Identification and Classification Engine (DICE) to help automate Internet-wide scans, label devices, and provide mechanisms for notifying owners about security issues (**RQ1c**). By applying DICE to 8 widely used protocols, the study highlights security pitfalls beyond common vulnerabilities, including signs of abandonment, obsolescence, and mismanagement. Furthermore, the modular approach provides a practical foundation for improving measurement reliability, reproducibility, and responsible disclosure practices (**RQ2a, RQ2b**).

RQ	Contribution
RQ2a	Identification of methodological limitations and noise affecting Internet-wide scans
RQ2b	Improvement of measurement reliability and reproducibility through modular scanning and classification

Related publication

R. Yaben and E. Vasilomanolakis, “Rolling the dice: A device identification and classification engine to detect vulnerable devices facing the internet,” in *2025 9th Network Traffic Measurement and Analysis Conference (TMA)*, 2025, pp. 1–4. DOI: 10.23919/TMA66427.2025.11097013

Original Abstract

In recent years, the Internet has experienced a significant surge in connected devices, with an ever-growing number of sensors and monitoring systems—spanning industries and domestic networks—now exposed to the Internet and reliant on our ability to keep them secure (e.g., in healthcare, home automation, and manufacturing). However, securing Internet-facing devices is no trivial task. Applying patches, firewall rules, and strong credentials are only small steps during their security life-cycle. Since these steps work in tandem, failing even a few can significantly increase the risk of compromise. The cybersecurity community continues to build on its efforts to mitigate this issue from many fronts, all while investigating society’s new challenges with technology and their security implications. To aid in this task, we present DICE, a modular Device Identification and Classification Engine to detect vulnerabilities on Internet-facing devices. DICE assists in most phases of the identification process, from automating Internet-wide scans to labeling results. In addition, DICE can help notify the affected device owners – an ongoing issue across the literature – by creating detailed reports and mitigation strategies. As proof of concept, we share preliminary implementations of various modules to identify recurrent issues in 8 protocols widely used in IoT and OT devices. These modules aim to discover security pitfalls beyond common vulnerabilities, such as signs of abandonment, obsolescence, and security negligence.

6.1 Introduction

As the number of Internet-facing devices continues to grow, so do concerns about their security. Ensuring these devices remain secure while exposed to the Internet is an increasingly complex challenge with potentially severe consequences. Although the field of cybersecurity is maturing and societal demand is rising, its complexity is evolving just as rapidly. We are reminded of this reality daily, as cybersecurity incidents soar and have a larger impact [1]. Therefore, the community's involvement is paramount in understanding the issue comprehensively and mitigating its threats. In this regard, notable efforts have been made to propose tools and methods to measure and analyze the Internet's population and behavior, such as Nmap, masscan, and the ZMap ecosystem; Scopus alone contains more than 600 publications on the topic of Internet measurements in the last 25 years and has grown every year since ZMap's release. As Durumeric et al. [23] mentioned in their review of ZMap's usage over the years, Internet scanning tools have been integral to studying Internet behavior and uncovering widespread security issues. According to their analysis, these tools are also found in many vulnerability scanning solutions (e.g., Palo Alto's Cortex Xpanse, Rapid7 InsightVM, and Nuclei) and Internet scanning services (e.g., Shodan, Censys, and ShadowRunner). However, despite the numerous publications and security tools available, one of the most common limitations – and promises to solve in future work – is the lack of reproducible and comparable results [20], [106]. Most studies lack transparency on their methodologies (e.g., publicly available probes, labeling, and classification systems), forcing authors to spend significant efforts re-implementing the state of the art instead of focusing on the issue at hand.

This paper introduces our current work in progress: DICE, a Device Identification and Classification Engine. DICE is primarily designed as a modular vulnerability identification engine for Internet-wide surveys, capable of profiling devices and orchestrating targeted scans. Our goal is to create a foundation for modern Internet surveys and address the aforementioned limitations. We are developing DICE as a common platform for the community to strategize, execute, analyze, and compare Internet measurements. As a proof of concept, we also share an early implementation of DICE modules to identify security issues in IoT and OT devices exposed to the Internet. We hope to inspire the community to establish new and refined requirements for Internet measurements and create a common language for this space, rolling DICE modules, and measure their effects. Our contributions are as follows:

- We introduce DICE, a modular Device Identification and Classification Engine for Internet measurements. We describe how DICE addresses common limitations in the literature, and explore further use cases and research directions.
- We demonstrate an example implementation of DICE signatures to detect indicators of security misuse, such as misconfigurations and abandonment. These signatures are applied to scanning results from a previous measurement study targeting eight widely used protocols in IoT and OT.

6.2 Related Work

Internet surveys are modern methods of studying the Internet and its population based on remote-host interrogation techniques. Tools such as ZMap and Masscan made scanning the Internet possible within hours and with fewer resources than their spiritual predecessor, Nmap. This method has proven to be extremely useful, with many studies uncovering widespread security issues such as Heartbleed [107], security concerns in IoT and OT [36], [38], [52], [108], [109], identifying or fingerprinting vulnerable devices [65], [110], issues

in honeypots [111], monitoring events such as tracking botnets [1] and the impact of war on critical Infrastructure [112]. Their limitations are also well studied, with multiple publications on the implications of vantage point location [9], Internet churn, scanning velocity, and blocking behavior [67]. The literature even has multiple examples of strategies and guidelines to conduct Internet measurements [106]. However, the lack of reproducible and comparable methods threatens to slow down the good pace achieved in recent years, even when this issue is commonly voiced across the literature and is treated as one of the most fundamental impediments in Internet measurement studies and surveys [20], [23], [106], [113]. In response to these challenges, we propose DICE, an engine that lays the groundwork towards standardizing processes and methods for conducting and comparing Internet measurements.

6.3 DICE: The Engine

This section introduces DICE's design principles and core concepts of its structure. In addition, we include use cases to explain how DICE can help mitigate the field's limitations.

6.3.1 Design Principles

DICE combines many design principles from previous attempts at creating similar solutions. Our design goals are meant to develop an engine that can assist at most stages of an Internet measurement as needed, whether it is classifying devices or orchestrating complex measurements.

Connected. Scanning, identifying, and classifying are separate tasks that DICE tackles simultaneously. DICE distributes each task in a component, which should remain separate for independent use (e.g., for re-classifying, verifying, comparing, or extending results). However, measurements with higher complexity and internal dependencies require DICE components to stay connected – known as the rule of composition.

Scanner agnostic. At its core, DICE is designed to support widely adopted scanners while preserving user flexibility to choose, integrate, or develop their own scanning tools. Its guiding philosophy is to advance the field by sharing resources and providing access to the tools and materials used in measurements. Imposing restrictions on scanner choices would undermine its usefulness; therefore, DICE remains loosely coupled with scanners and deliberately unopinionated.

Support a wide range of measurements. We adopt the concept of *measurement modules* described by Paxson et al. [114], and heavily used in designing most modern scanning tools (e.g., Nmap, Masscan, ZMap, and ZGrab). In DICE, we separate the engine from the measurement, which becomes a chain of probes, identification rules, and classifiers to survey the Internet.

Self-reflective. As an engine for orchestrating and comparing Internet measurements, DICE must provide testable metrics that reflect the measurement state, performance, and results summaries. Such metrics are rarely shared in existing studies, making evaluating and comparing measurements difficult. Sharing these metrics enables use cases such as observability analysis, performance tuning, and classification modeling. Future versions of DICE could leverage these insights to diagnose measurement issues and recommend improvements.

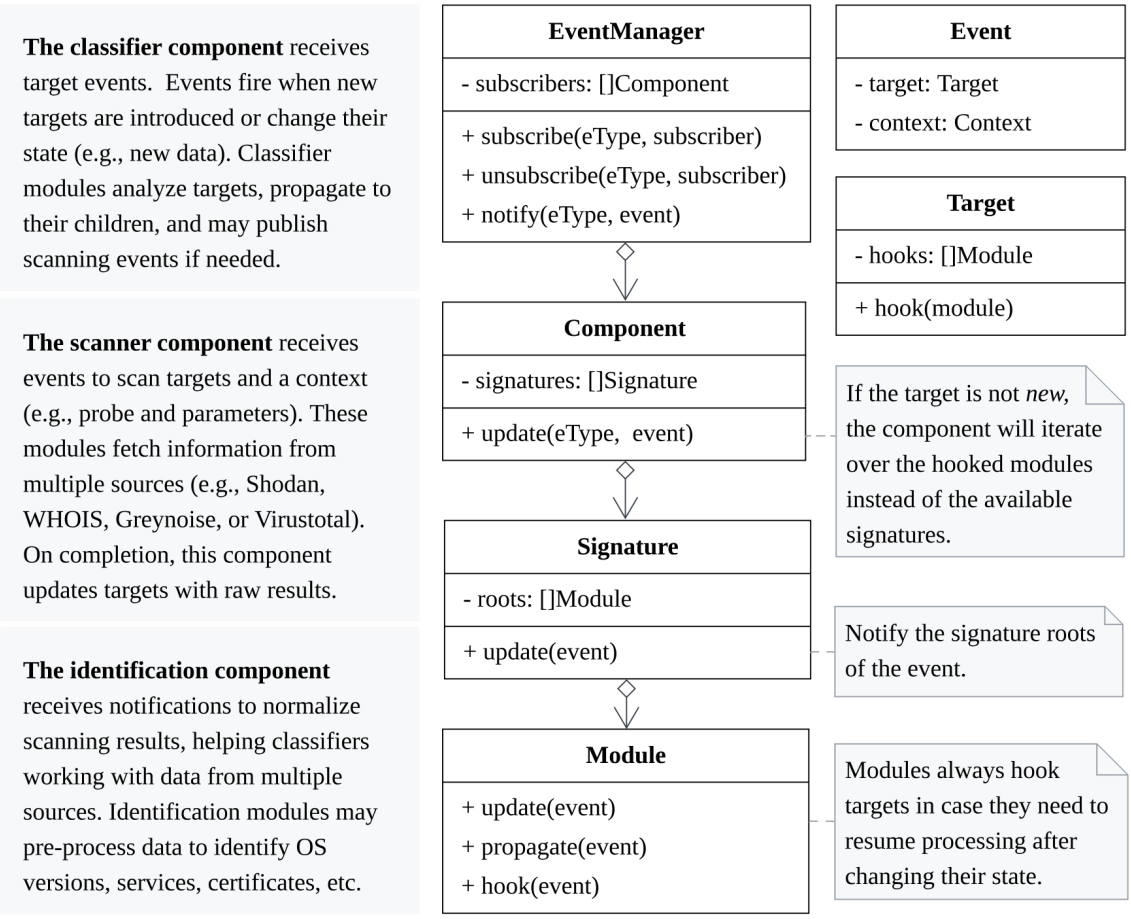


Figure 6.1: DICE's implementation of observer pattern and descriptions of the three main components: *scanner*, *classifier*, and *identifier*.

Help navigating results. DICE functions should remain within the scope of the engine, i.e., orchestrating measurements, and identifying and classifying devices. However, its usefulness would be significantly limited without built-in methods for navigating and interpreting results. This additional functionality enables result comparison and report generation, streamlining the process of conducting responsible disclosures.

6.3.2 Structure

This section provides a brief overview of DICE's core concepts, as depicted in Figure 6.1: *components*, *signatures*, and *modules*.

Components. DICE splits the phases of a measurement into *components*: scanning, identification, and classification. These components are *modular* to support a wide range of measurements, designed to work simultaneously as needed, and supervised by the engine. Module behavior depends on the parent component: (1) scanning modules take targets and dispatch scanning results, (2) identification modules normalize these results, and (3) classification modules assign labels and may propose further scans. Targets are DICE's inputs, references to an IP address in a measurement, and hold all related data to that address. DICE outputs targets as profiles of the collected addresses, their discovered services, fingerprints, and labels assigned. DICE enables this functionality by adopting an event-driven architecture with the observer pattern, propagating targets across component modules to manage state transitions and assign labels.

Signatures. Conceptually, components in DICE are made of Directed Acyclic Graphs (DAGs) in which the vertices correspond to modules. DAGs are powerful tools helping DICE to add dependencies between modules and avoid loops. First, DAGs are directed, i.e., graphs are made of vertices connected by edges with a direction (DAGs can have only one edge between two vertices). Then, DAGs are acyclic, i.e., the graph does not contain loops. Other rule-based systems with similar characteristics name their rulesets *signatures* (e.g., Suricata and Snort). We also adopt this naming convention for simplicity and refer to DAGs of modules in DICE as signatures. Signatures maintain their mathematical properties, allowing DICE to combine and embed other signatures to form a larger one. This approach is particularly useful for collecting signatures (e.g., identifying IoT issues) and describing measurements as configuration files, including other parameters such as the scanner choice. This design helps ease sharing and comparing measurements.

Modules. Simply put, modules are processing units that assign labels or add further information to targets, including scanning results. DICE communicates with modules over RPC, simplifying the development of new modules and supporting most programming languages. Modules vary in complexity, from simple filtering rules to complex classification models. DICE feeds targets to all roots of the loaded classification signatures (roots are vertices without an inbound edge). Classification modules may check for prerequisites before processing targets and suggest new scanning signatures. Targets that match a module's evaluation are assigned a label and propagated to the module's children.

6.3.3 Use cases and research directions

DICE's modular structure and dependency features allow us to define and share whole processing pipelines and measurement metadata (e.g., monitoring metrics, aggregations, and configurations). We anticipate DICE will be used beyond the covered in the literature and help in other areas of interest, e.g.: (1) those derived from longitudinal studies, such as outages and cyber-security developments, (2) artificial intelligence on DICE classified datasets and new modules using these models, (3) IP and port prediction techniques to improve efficiency and scanning coverage, (4) mitigating Internet churn for time-sensitive measurements, etc. We look forward to seeing what other research questions DICE may support.

Figure 6.2 shows a simplified example of DICE pipelines to scan the Internet for vulnerable devices exposing MQTT and OPC UA services. In this example, DICE loads the signatures to scan, identify, and classify hosts exposing these services, and any additional signatures included as internal dependencies, such as certificate classifiers or other services. Signatures describe their layout, referencing modules and how they are linked (left side). In this case, the MQTT `access` module should only trigger on targets with `auth` labels, such as successful anonymous authentications or through self-signed certificates. Similarly, certificate modules trigger when a service includes this information during the communication – dependencies are represented as circles in the graph. Lastly, DICE outputs session metadata, scanning results, identification fingerprints, and classification labels to a new database. These results can be used to conduct responsible disclosure campaigns and for further analysis.

To inspire the community, we share an early implementation of signatures to identify vulnerable IoT and OT devices following a classification criterion similar to the ones presented in [12] (see [115]). These signatures include modules to identify access control issues, allowing untrusted clients to establish anonymous connections and access internal information. Moreover, we evaluate server certificates for validity issues, reuses, and insecure configurations. The included identification modules fingerprint hosts at different

```
$ dice scan -s opcua,mqtt
```

```
// opcua.dice
cls opcua-auth
cls opcua-access (cls: opcua-auth)
cls opcua-nodes  (cls: opcua-auth)
sig certificates (cls: opcua-auth)

// mqtt.dice
cls mqtt-auth
cls mqtt-access  (cls: mqtt-auth)
cls mqtt-topics  (cls: mqtt-auth)
cls mqtt-cve     (cls: mqtt-auth)
sig certificates (cls: mqtt-auth)

// certificates.dice
cls x509-validity
cls x509-encryption
cls x509-reused
```

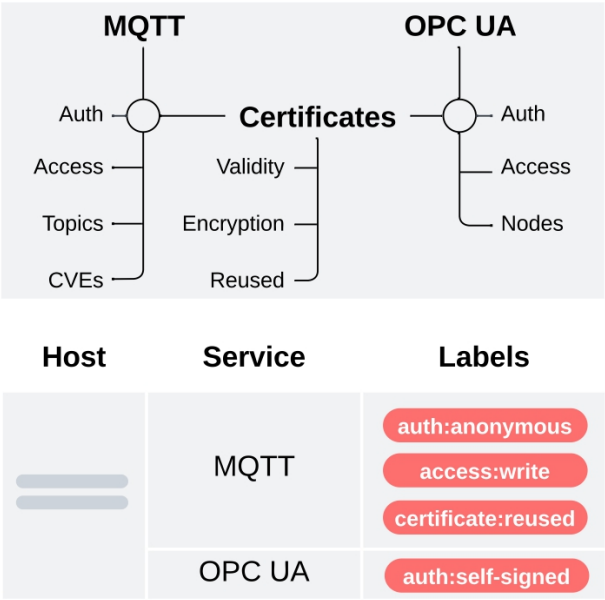


Figure 6.2: Example of DICE scanning pipeline targeting Internet-facing devices exposing vulnerable OPC UA or MQTT services.

granulation levels based on the information they leak, and classify those leaking sensitive information (e.g., server state) or suffering from poor maintenance (e.g., deprecated versions). Most of these modules build upon previously assigned labels (e.g., authorization rules can only be tested on targets with authentication labels), exploiting the signature dependency features in DICE to create meaningful profiles of Internet-exposed devices. Finally, we use a dataset from early 2025 with 8 protocols as an input, and show a summary of the results.

7 Measuring What Matters: Revisiting Internet Exposure of OT Networks

Context and Contributions

This chapter addresses **RQ2** by examining the exposure of OT networks on the public Internet. It highlights that many Internet-wide scans are affected by noise such as honeypots, network telescopes, and tarpits, which can inflate counts of vulnerable devices (**RQ2a**). By applying a noise-aware methodology across Modbus, Fox, EtherNet/IP, and IEC 60870-5-104 services, the study identifies systematic sources of dataset pollution, including condensation, displacement, volatility, and hostility. Even after filtering these artifacts, misconfigurations, obsolescence, and broader security management issues remain widespread (**RQ2b**), emphasizing that the underlying security landscape of OT devices is largely unchanged. These findings provide guidance for improving measurement accuracy, reliability, and reproducibility in future Internet-wide studies.

RQ	Contribution
RQ2a	Identification of limitations affecting current Internet measurement methodologies targeting IoT and OT networks
RQ2b	Improvement of measurement accuracy, reliability, and reproducibility by accounting for noise and artifacts

Related publication

R. Yaben et al., "Measuring what matters: Revisiting internet exposure of ot networks," Journal preprint, available at SSRN, 2025. DOI: 10.2139/ssrn.5974783. [Online]. Available: https://ssrn.com/abstract=5974783[Preprint]

Original Abstract

Many Internet-wide measurement studies report thousands of vulnerable 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.

7.1 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 [93], [116], [117]. Other studies highlight widespread security management issues (for example, unrevoked certificates, use of certificates past their end-of-life dates, and deprecated firmware) [12], and a large part of the community focuses on discovering vulnerabilities [118], [119] and characterizing attack trends [7]. The urgency of this topic is justified, as OT systems are both essential and increasingly attractive targets. Incidents such as Stuxnet [120], [121], BlackEnergy [122], Industroyer [123], Havex [41], and LockerGoga [124] 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 [32]. According to Srinivasa et al. [33] and Mladenov et al. [32], 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/IP and IEC 60870-5-104 (IEC 104) as two additional widely used protocols that are underrepresented or mischaracterized in the literature. In 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 proto-

cols 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 [125].

The rest of this paper is structured as follows. Section 7.2 describes our data collection methodology and the ethical considerations we took into account during our experimentation. Section 7.3 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 7.4 analyzes the results of our four case studies: Modbus, Fox, IEC 104, and Ethernet/IP. Section 7.5 discusses the impact of noise on OT Internet measurements, along with limitations of our study and directions for future work. Section 7.6 presents related work on active Internet measurements of OT networks and the research gaps that motivate this study. Section 7.7 concludes the paper.

7.2 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.

7.2.1 Data collection

Our data collection method consists of an active Internet-wide scan using a traditional two-step approach combining ZMap [50] and ZGrab2 [97], supplemented with crowdsourced data from RIPE Atlas [126], and CTI and IP reputation services GreyNoise [57] and AbuseIPDB [104]. Using exclusively CTI services (e.g., Shodan [55] or Censys [7]) 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 AS data.

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

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 [7], [9].

Scanners. We use ZMap and ZGrab2 to conduct Internet-wide L4 and L7 sweeps [50], [97], 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 7.4 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.

7.2.2 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 [6]. 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.

7.3 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 [38]. 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. [127] and Singla et al. [128] 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 [25], [32]. 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 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 7.1, 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 7.2 provides a summary of these policies.

7.3.1 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. [32] 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. [27] 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 [7], [129], often showing these concentrations are more common but not unique to research networks [111]. Despite being considered a known oddity in the literature, the concept has rarely been openly correlated to deception systems or other forms of noise [111] – with few exceptions, such as Knight et al. [130], which leverages artificial diversity in dense populations as a form of bio-inspired deception. In fact, Durumeric et al. [7] 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. [38] 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. [38], [131] As Män-

Table 7.1: 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 7.2: 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

nel et al. [131] 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 [95], [132], [133], [134], [135]. 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 (7.1) where P is the prefix and $\sum H_P$ is the number of responding hosts during our ZMap scan on that prefix.

$$\text{Density}(P) = \frac{\sum H_P}{2^{32-\ell(P)}} \tag{7.1}$$

Where:

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 7.1 illustrates this relationship, and how, despite fitting hosts exposing different protocols, the

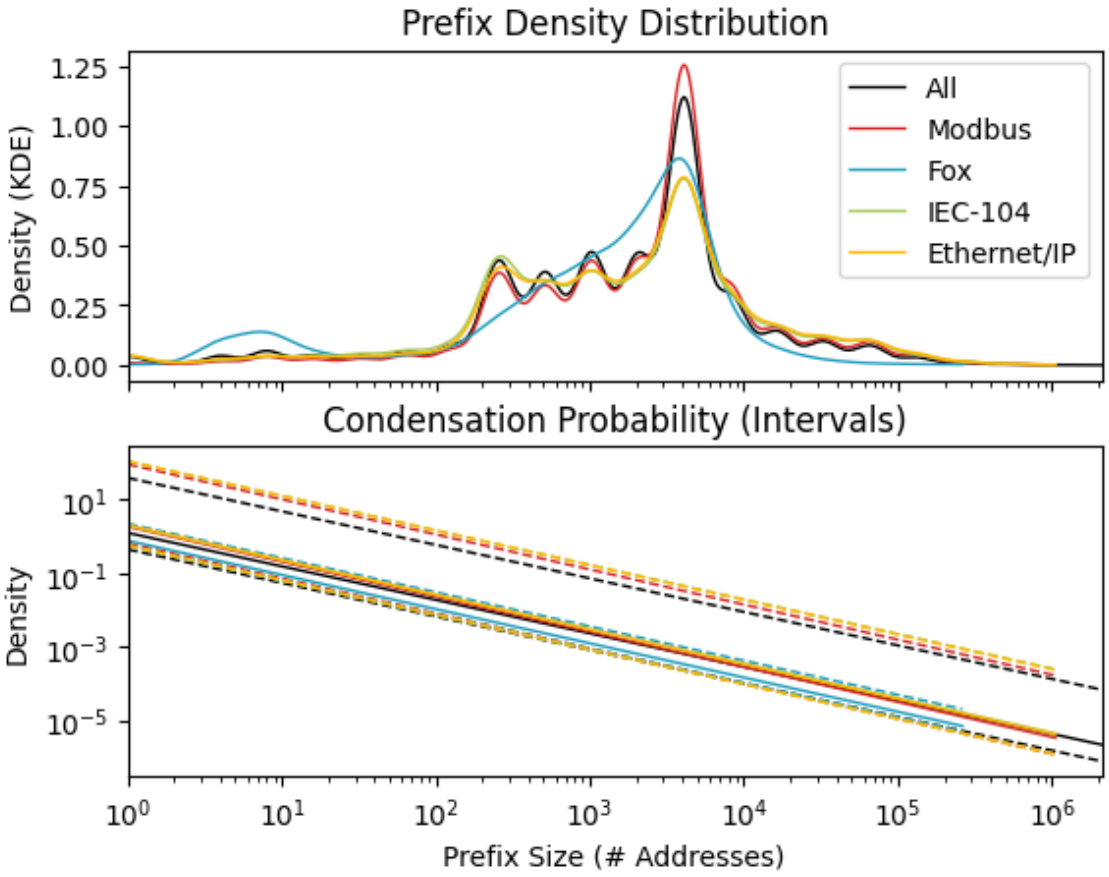


Figure 7.1: Prefix density distribution (top), and condensation probability ranges for prefixes of each size (bottom). Probability ranges are given between the dashed lines representing the confidence interval between percentiles at the extremes (95% and 5%). Condensed labels are assigned to prefixes with hosts crossing the $p > 95\%$ threshold for their size.

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 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 7.2 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 7.8 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

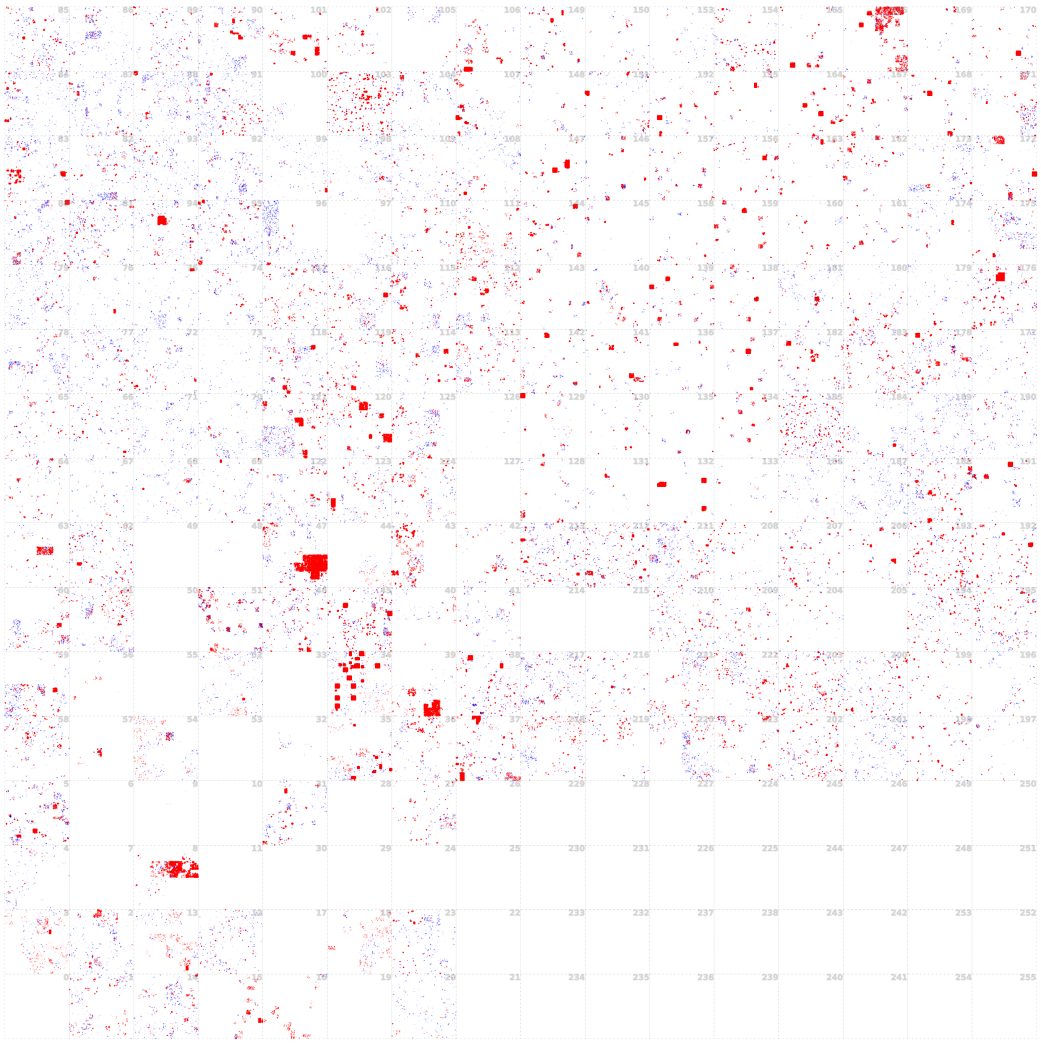


Figure 7.2: Dense prefix classification over hosts responding to OT probes. The figure shows large clusters of host addresses classified and colored in red, while scattered hosts across prefixes are represented in blue. Four large clusters are particularly dense in octets 8, 34, 39, 47, and 166.

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.

7.3.2 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 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. [32] 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. [33] 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 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 [25], honeypot fingerprints [136], 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 [32], [33]. 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

Table 7.3: 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

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 7.3, 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 [32], [136]).

Aletheia. We implement the signatures described by Cordeiro and Vasilomanolakis [25] 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, 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 [32], [33], [136], [137] for ICS environments with Conpot [138] 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 [139] 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 7.4.4). While simplistic, these classifiers could identify 240 odd hosts.

7.3.3 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 [140], 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 [140], [141]. 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.

7.3.4 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* [142]. 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 [143]. Some of these tarpits try to confuse stateless network scanners by sending up to a hundred TCP handshake completion packets [142], which could explain some of the behaviors experienced by the community. Other far more aggressive tarpit variants may block functions in their clients [144], 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.

7.4 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.

7.4.1 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 7.4 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 deduplicates 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 con-

Table 7.4: 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

nection 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 7.5; 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 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.

7.4.2 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

Table 7.5: 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	Class	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

where we can test for tarpitted connections abusing this vector. Figure 7.3 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 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 dataset contains 92 unique vendors and 279 products with 659 different firmware revisions. Figure 7.4 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 [13] reveals minimal differences, mainly associated with a wider spread

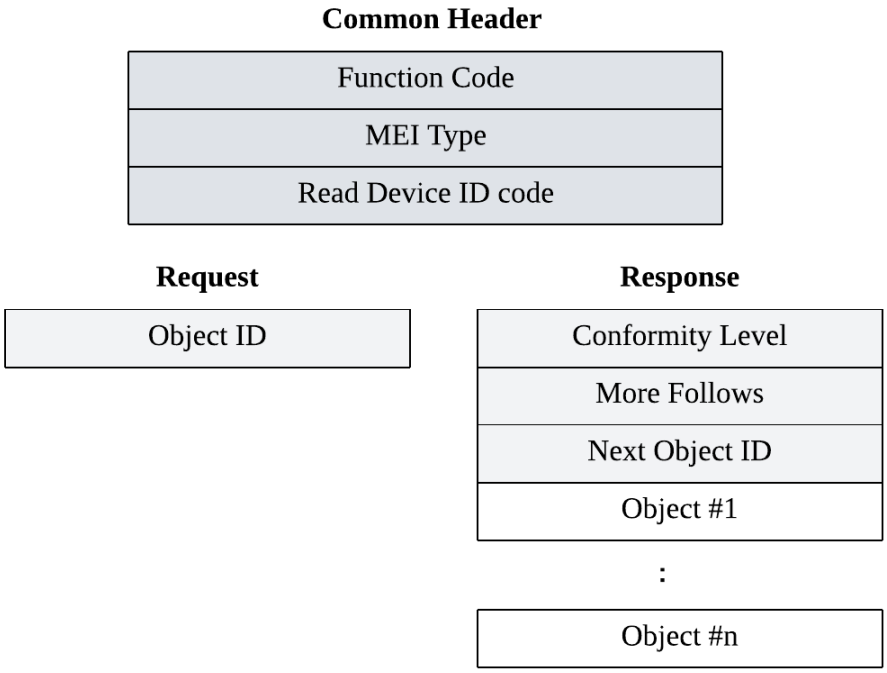


Figure 7.3: Modbus frame structures of reading requests and MEI responses with common headers. MEI responses may contain one or more objects, and may be split into multiple frames.

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

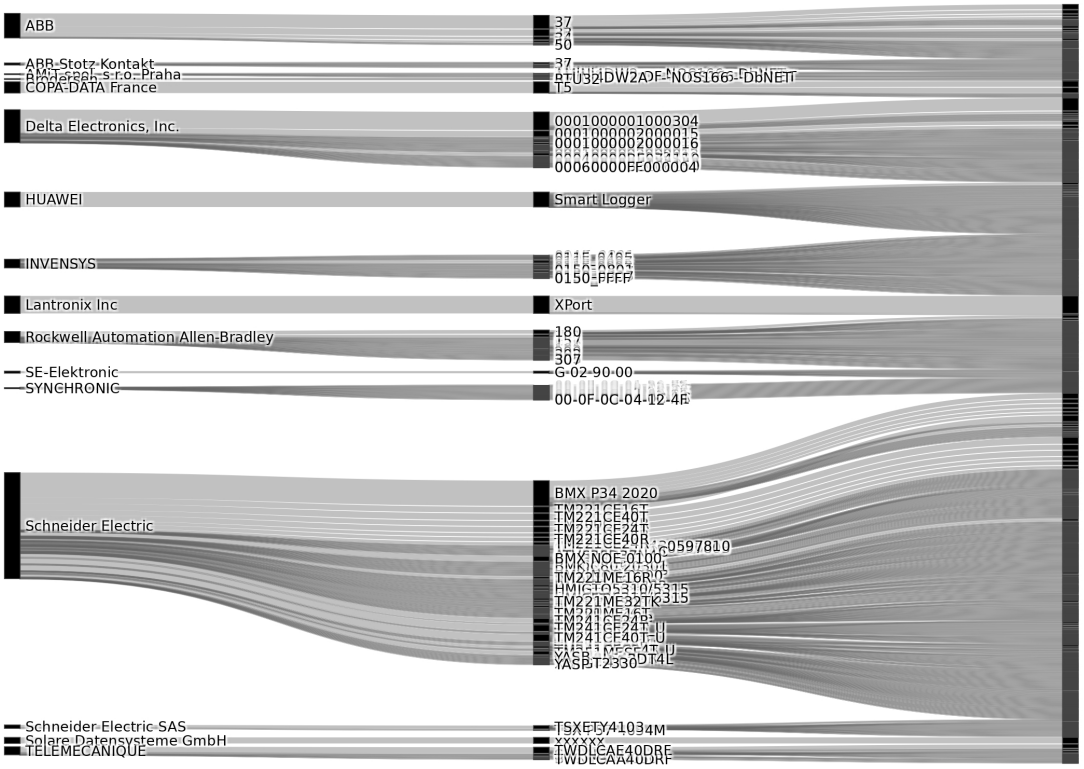


Figure 7.4: Modbus sankey diagram of vendors, products, and distribution of revisions occupying 75% of the dataset.

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 7.5, 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 miss-

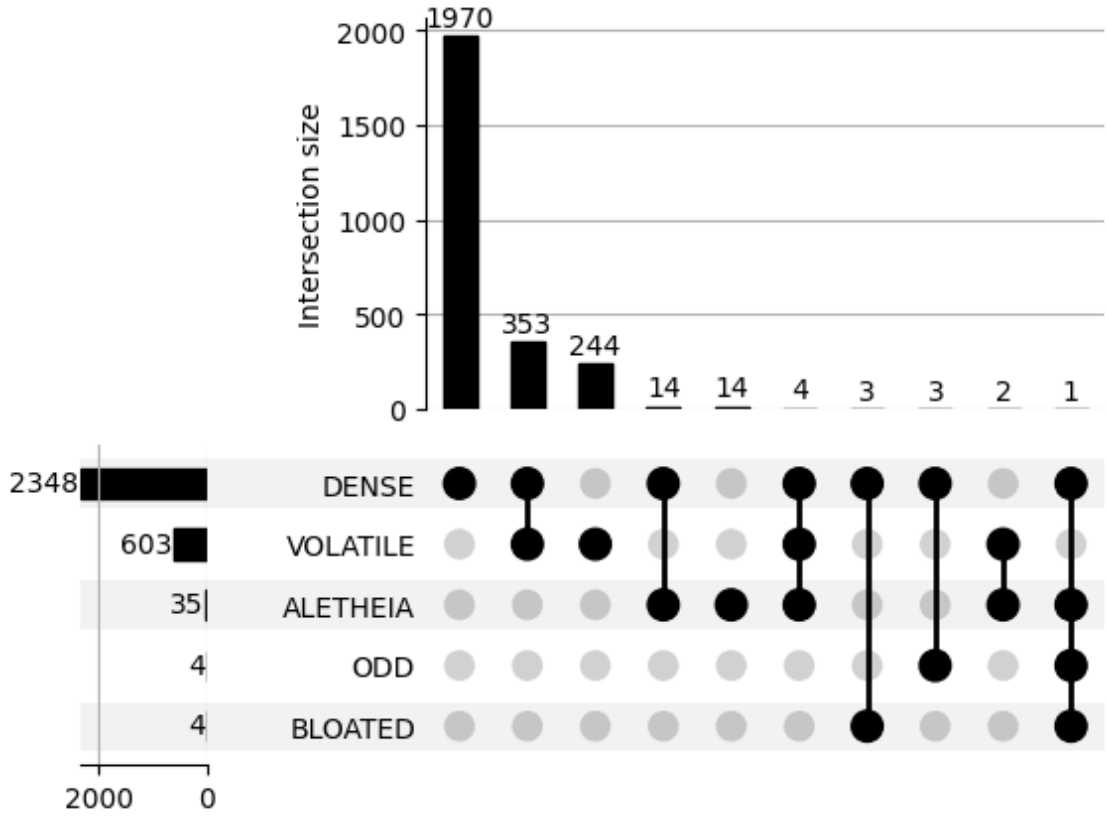


Figure 7.5: Modbus correlation diagram of noise labels.

ing 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 using the aggressive policy.

7.4.3 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.

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

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

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 7.6 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.

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 7.7 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.

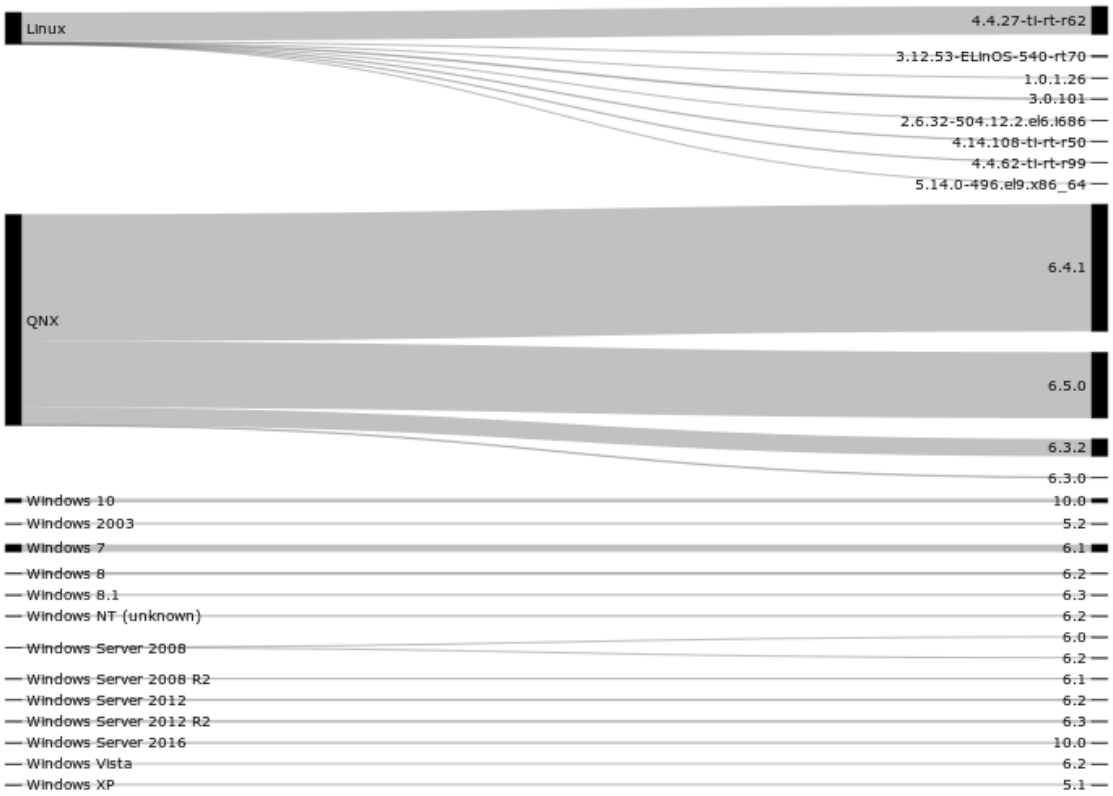


Figure 7.6: Fox distribution of operating systems and their versions.

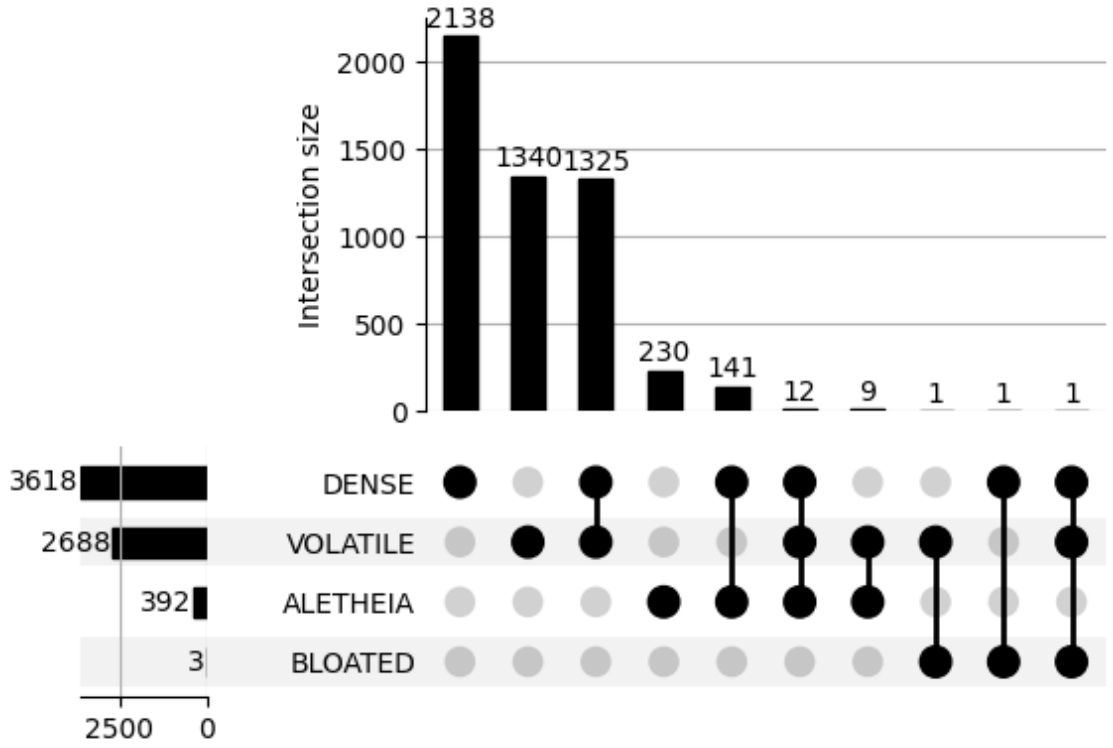


Figure 7.7: Fox correlation diagram of noise labels.

7.4.4 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 7.8 shows the probe interrogation process to identify IEC 104 services, while Figure 7.9 describes the content structure of the APDU.

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

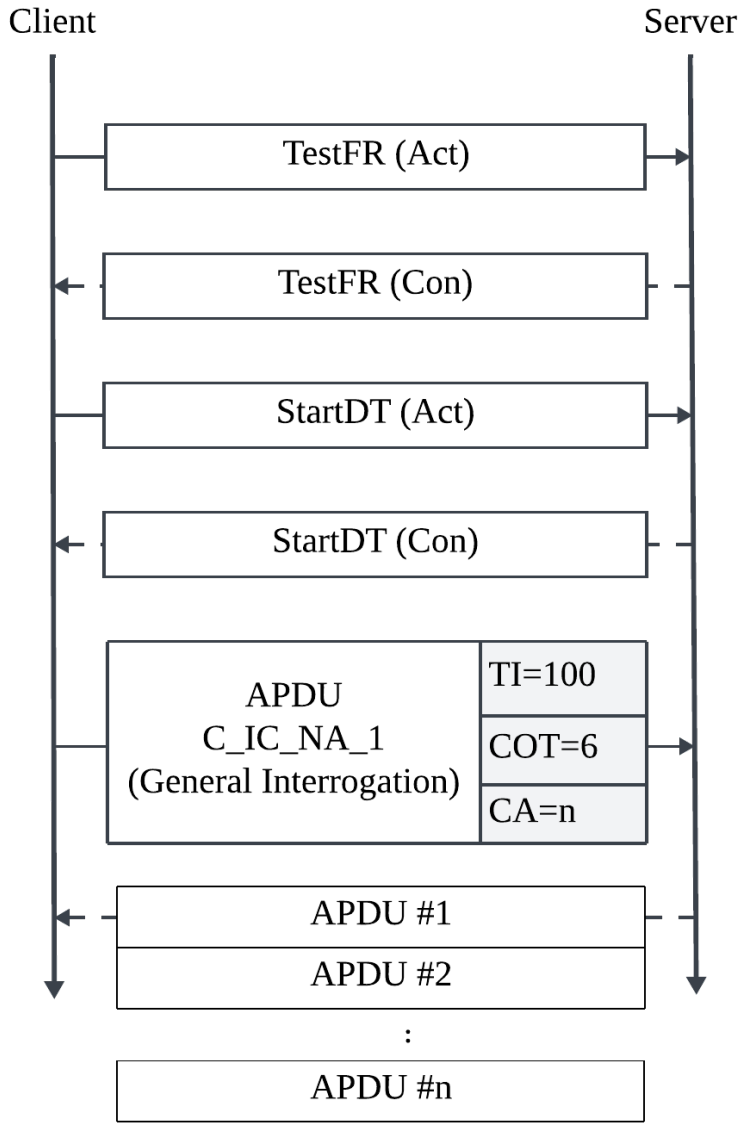


Figure 7.8: IEC 104 probe communication flow between the vantage point and remote hosts, sending `TestFR` and `StartDT` requests followed by a General Interrogation command.

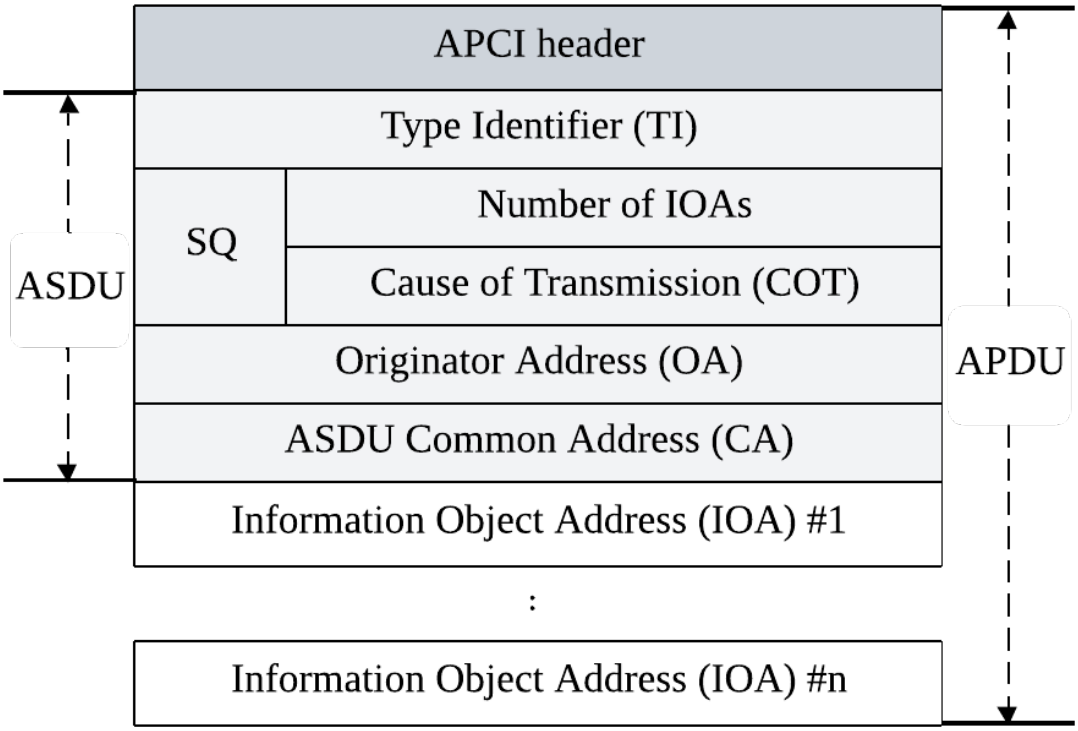


Figure 7.9: IEC 104 APDU structure of major fields with expanded ASDU section. A single frame may include multiple APDUs, each with an arbitrary number of IOAs, within the limits of the allowed frame size.

Figure 7.10 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.

Excepting the *dense* tag, the majority of tagged IEC 104 services include more than one tag. Figure 7.11 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% – 5̃0% 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

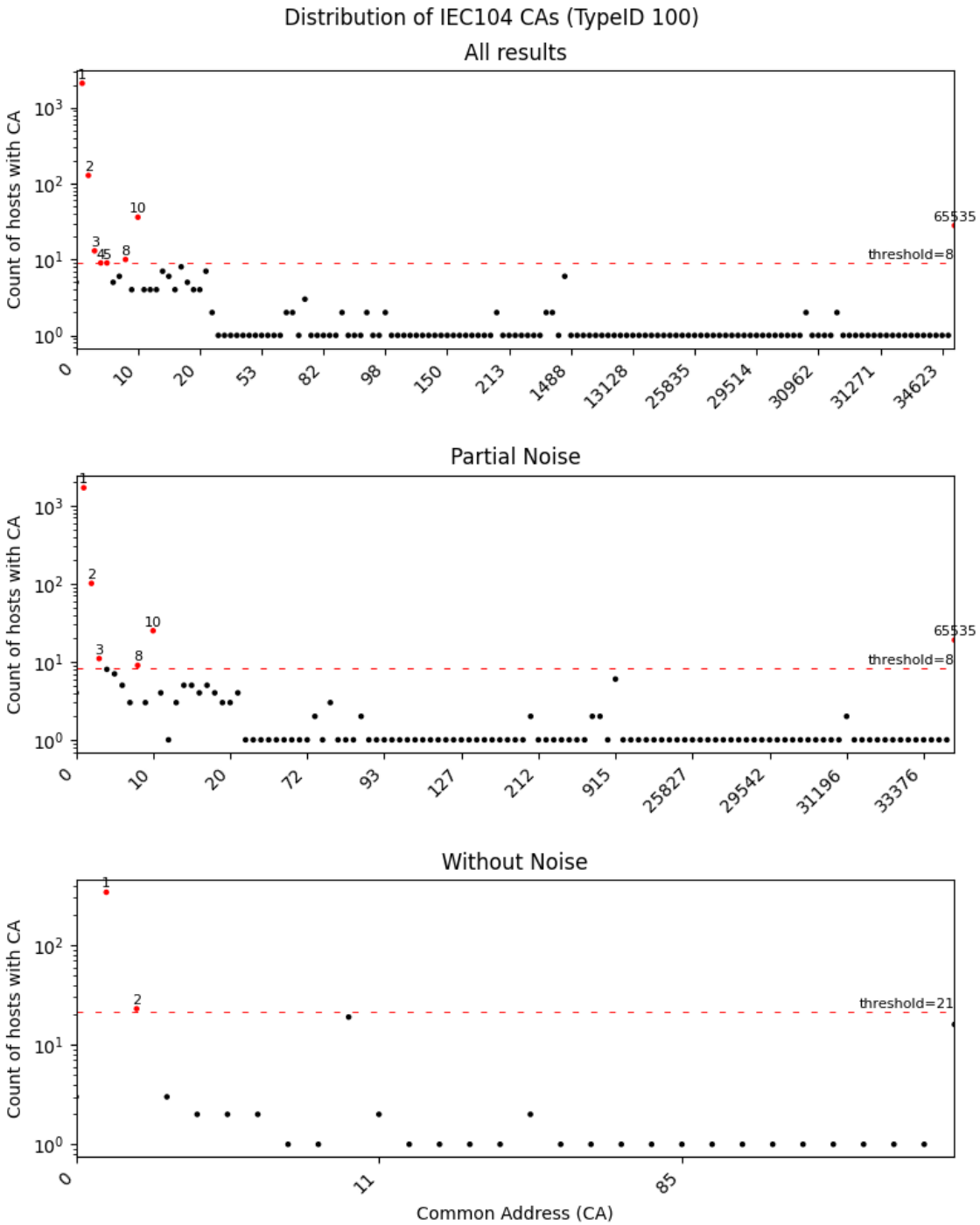


Figure 7.10: Distribution of IEC-104 CAs in: i) all results, ii) partial noise removed (w/o dense or volatile), and iii) removing all noise. Annotated CAs represent the most commonly found in our dataset across hosts.

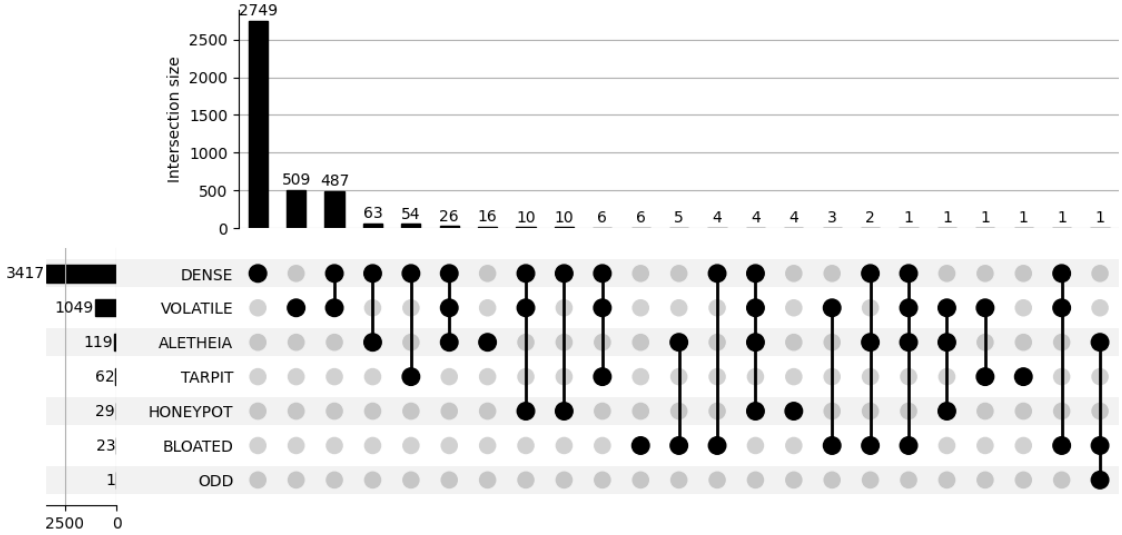


Figure 7.11: IEC 104 correlation diagram of noise labels.

(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 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.

7.4.5 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 [145]. 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 7.12 provides a representation of the overall structure of Ethernet/IP response frames, showing the major fields included for device identities.

We classify Ethernet/IP services as exposed when servers respond to the probe's `ListIdentity`

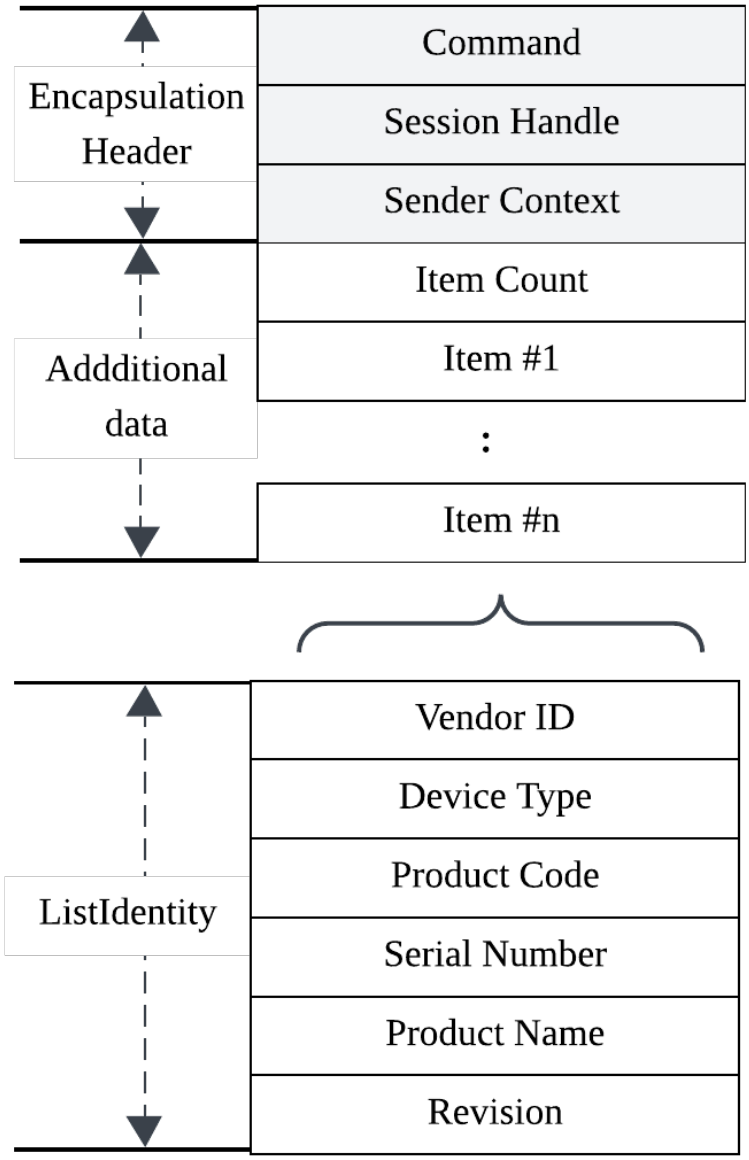


Figure 7.12: Ethernet/IP frame structure with major fields with response payload as *Additional data*. Additional data response structures vary for each command; the figure shows `ListIdentity` items and their informational fields.

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 [146]. 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 [147]. However, a single controller may have a large number of adapters and other devices, such as switches and HMIs [148]. 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 7.13 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

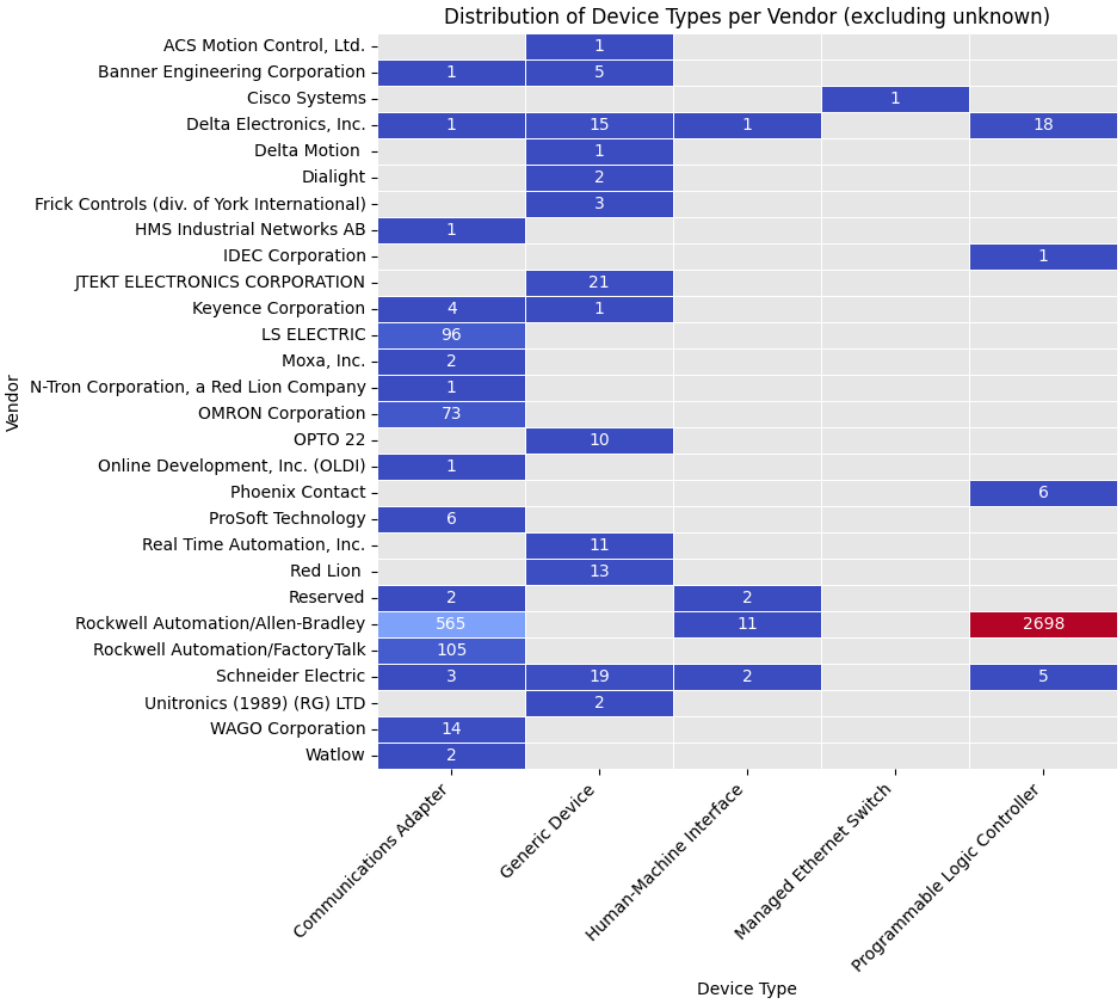


Figure 7.13: Distribution of Ethernet/IP vendors and device types.

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 7.14 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 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 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 7.15 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

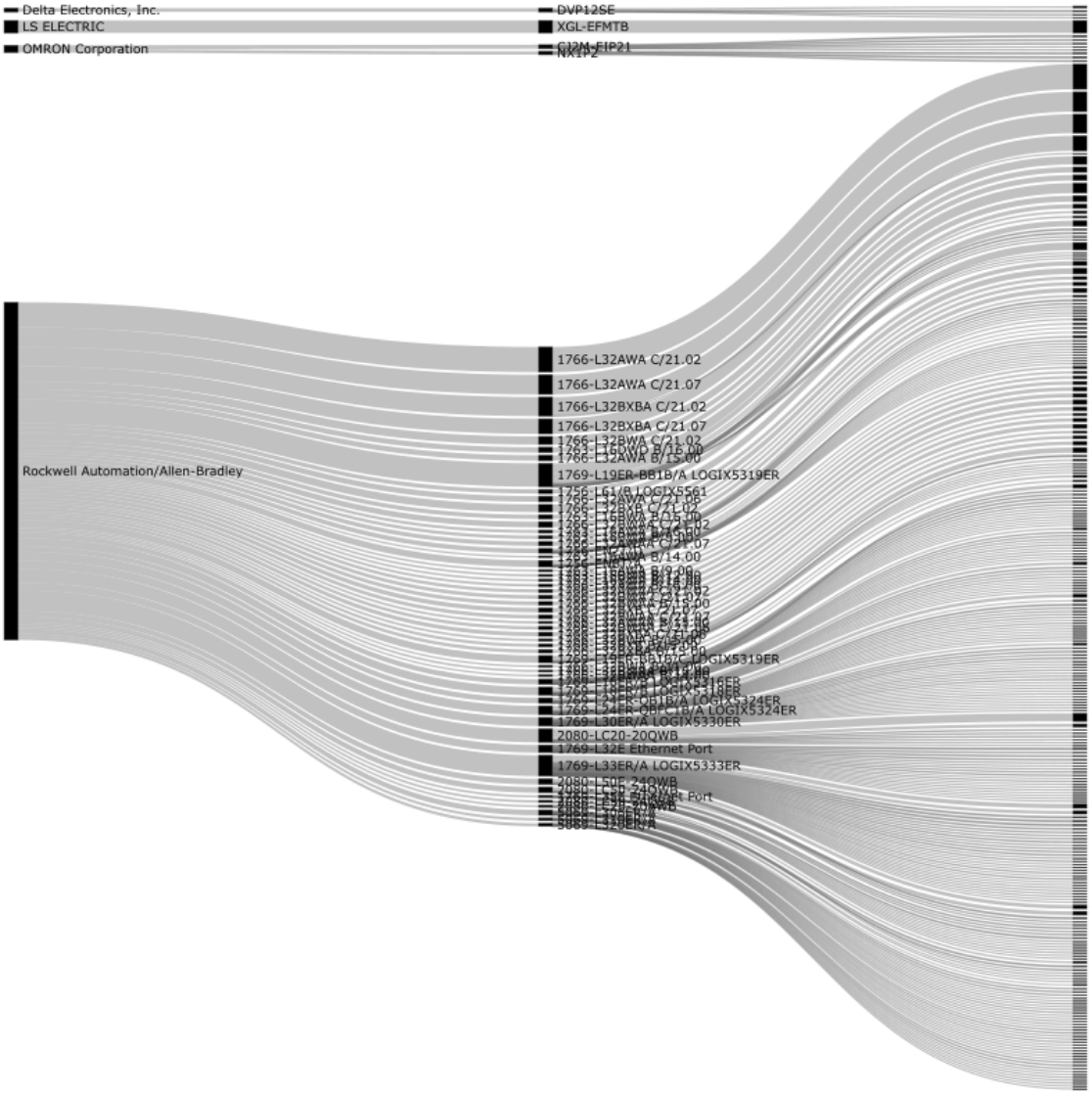


Figure 7.14: Distribution of Ethernet/IP vendors and products.

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 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 Ether-

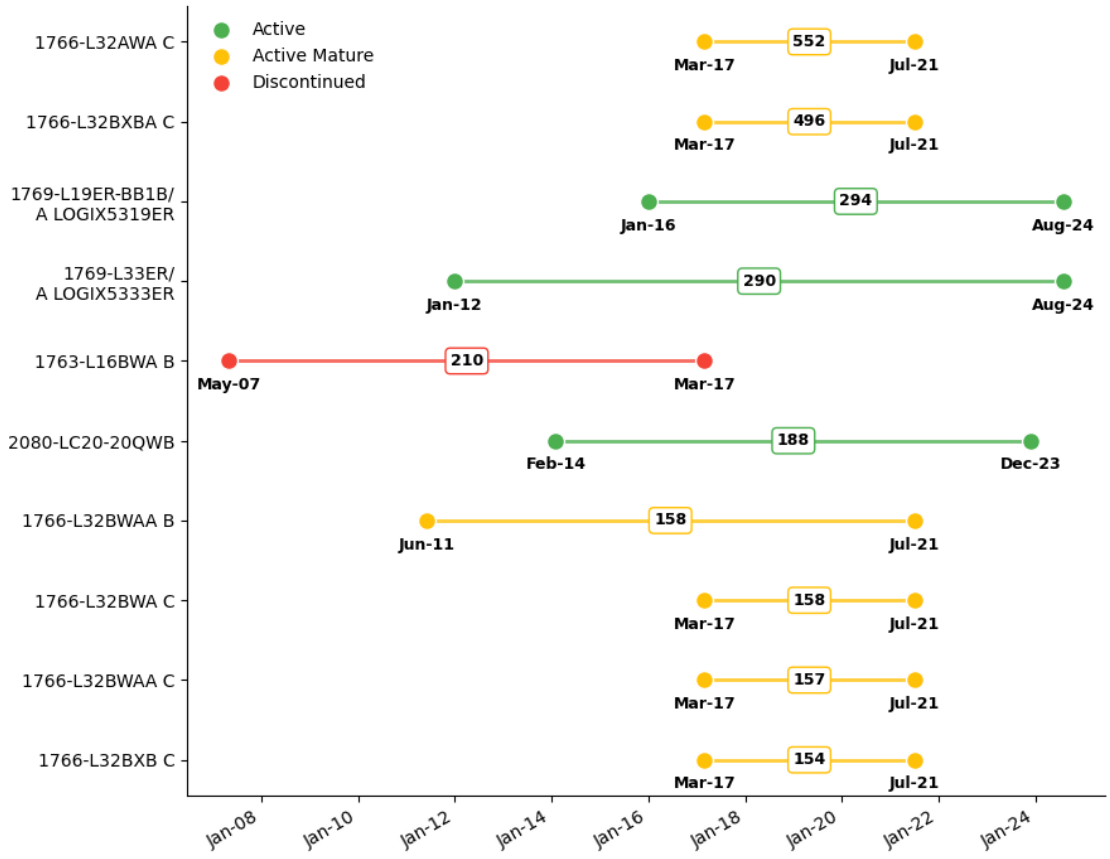


Figure 7.15: State of maturity of Ethernet/IP products found in our dataset.

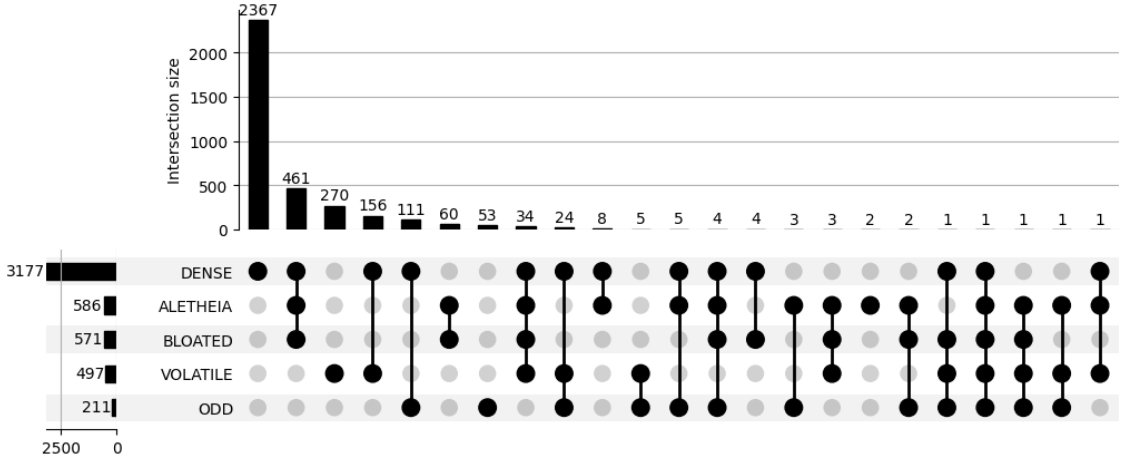


Figure 7.16: Ethernet/IP correlation diagram of noise labels.

net/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 7.16 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.

7.5 Discussion

7.5.1 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.

7.5.2 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 pro-

cesses, 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.

7.6 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 [23]. 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 [149].

While gaining momentum, the current state of the literature contains a moderate number of publications discussing the security of OT facing the Internet [150]. 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) [95], [151], [152] or use deception systems to collect information (e.g., ICS honeypots) [137], [153], [154]. Others have measured the use of deception systems in OT [32], [33], [136], and even fewer conduct active Internet surveys using stateful probes (i.e., sweep scans followed by banner-grabs) [12], [37], [39]. Authors have raised concerns about OT environments being too sensitive for traditional scans, and that communicating with these networks could be catastrophic [127]. 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. [117] 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. [2] 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 [116] 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. [93] 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 hard-coding credentials).

Regarding Internet measurements, the work of Mirian et al. [4] 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. [127] 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 IoT exposed systems, Dahlmanns et al. [39] 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. [12] 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. [37], 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 [14] 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 [7], which uses data from Censys to report on their findings. [48] 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 [155] 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 [15]; 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. [33] and Mladenov et al. [32] 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.

7.7 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.

7.8 Host distribution

Figure 7.17 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 7.17: 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.

8 Synthesis & Conclusions

This thesis explored widespread cyber-security weaknesses in exposed IoT and OT networks. In particular, the work included in this thesis focused on security issues and concerns arising from mismanagement, negligence, and obsolescence. We discussed their origin and principal contributing factors leading to their proliferation. These weaknesses were shown to be neither isolated to IoT nor limited to specific subsets of systems, but instead widespread, long-lived, and equally impactful across OT environments. This section summarizes lessons learn throughout this thesis (cf. Section 8.1), including a self-reflection on our methodologies, results, and limitations. We conclude this thesis by discussing open challenges and immediate avenues for future research (cf. Section 8.2), followed by a concise summary of the work (cf. Section 8.3).

8.1 Discussion

Throughout the first part of this thesis, we focused on exploring cyber-security weaknesses affecting IoT and OT devices exposed to the Internet. Our results showed that most security issues are ubiquitous to all sectors, including residential networks, commercial environments, industrial setups, and even critical infrastructure. The methods used to identify these issues followed current practices in the field for large-scale measurements. We gathered a single snapshot of the Internet using newly developed probes with enhancements over traditional banner-grabbing methods to gain sufficient detailed information to evaluate security aspects such as authentication, access control, and encryption. This method constituted a leap forward in identifying security issues in depth while remaining within the ethical boundaries of Internet measurements. Authentication was only tested in the frame of guest accounts and determining when identification measures were absent altogether. This included using empty credentials and self-signed certificates when possible. In addition, probes sent discovery and informational requests to gather configuration and state details from the exposed services, never attempting to manipulate or alter the device or store data. By doing so, our analyses could detect open services allowing unauthorized clients to interact with the device. Further, we could investigate encryption adoption and infer maintenance behaviors. These details help identify devices lacking features, running on deprecated software, and missing security patches.

We proposed new methods to leverage these data to uncover widespread security weaknesses, such as certificate reuses, vendor-specific behavior, and the prevalence of product obsolescence facing the Internet. With further inspection and frequent scanning, these details could prove crucial in revealing aspects only visible through longitudinal studies. For example, a limitation of our work is the lack of monitoring to observe the evolution of devices we deemed vulnerable. Chapter 4 touches on this angle, where we extended a previous study with the comparison between two scans one year apart from each other, and showing that up to 50% of OT devices remain vulnerable and unchanged for long periods, with rare cases of downgrades, and few device replacements. However, more frequent scans can be beneficial to examine other interesting factors, such as vendor-specific update adoption, rolling default configurations, or the effects of responsible disclosure campaigns on operator behavior (e.g., how long it takes for certain sectors to react to vulnerability disclosures).

Unfortunately, conducting large-scale longitudinal measurements is complex. Available methods favor cross-sectional studies; adding new ports and protocols to Internet-wide

scans comes with its own set of issues that may degrade the measurement's coverage, precission and accuracy. To mitigate these effects, authors have proposed alternative experimental setups based on distributed deployments and continuous scanning [7]. However, continuous scanning cannot be used to compare the state of the Internet at two particular points in time. Moreover, scanning strategies that opt for vertical accuracy are not yet feasible at the extremes. Even highly distributed deployments, such as those from CTI platforms (e.g., Censys, Shodan, and ZoomEye), are limited to under a thousand ports for optimal reliability, while the rest are scheduled at different rates. Beyond this point, service identification accuracy drops significantly, with the only current alternative being scaling the deployment.

We explored these challenges throughout the second part of this thesis. Conventional scanning strategies rely on stiff multi-staged approaches. This method is sequential in nature; experiments first conduct sweep scans to test for liveness, and follow with banner-grabbing probes to detect the targeted services on responsive hosts. As discussed, the measurement time window is an important factor of the experiment. As a result, the gaps between each stage degrade the coverage of the experiment: devices may become unresponsive, block connections from subsequent scans, or alter their behavior altogether. This degradation is further intensified when studies require analyzing and often crowdsourcing. The work of Izhikevich et al. [30] identified these issues and attempted to mitigate them by creating a unified approach, where handshakes from sweep scans are reused to continue with banner-grabbing probes. This method dramatically improves scanning efficiency. The authors also introduced service-guessing methods to extend the range of covered ports. Overall, their work highlighted the limitations of the current available tools and set the basis for scalable Internet-wide measurements.

Building on the lessons learned from the literature and our own methodology limitations, we propose a new framework to improve scanning efficiency, open new possibilities for complex scans, and address the current reproducibility issues that are hindering the field, thereby fostering comparisons between measurements (cf. Chapter 6). As a proof of concept, we translated the vulnerability identification methods used throughout the first part of this thesis, demonstrating how this tool can benefit the field by sharing methods and instrumentation. The main benefit this tool offers for scanning is injecting mid-scan evaluation rules. Adding logical procedures to large-scans can lead to higher completeness with lower resources. In addition, the evaluation capabilities of this tool enable researchers to identify vulnerabilities and classify hosts showing suspicious behaviors as needed, either offline or while scanning. As a consequence, researchers can develop dynamic scans with nuances particular to their measurements.

These new capabilities facilitated the work presented in Chapter 7, where we focus on identifying false positives and false negatives distorting Internet-wide measurements of exposed OT networks. One of the main challenges discussed in the literature is the presence of noise in measurement datasets [32], [33]. This includes hosts and services that, in principle, qualify for measurement but are in fact Internet artifacts (e.g., Internet telescopes and honeypots). While we developed several heuristics to detect honeypots, tarpits, MTD, and discussed how many vulnerable devices seem clustered in a few AS, verifying these findings is challenging and would benefit from further work. In particular – and similarly to the limitations in identifying widespread security weaknesses, – most noise detection methods improve their accuracy with scanning frequency.

Overall, our findings suggest that more frequent, longitudinal measurements are increasingly necessary to uncover and mitigate cyber-security weaknesses. The threat of vulner-

able Internet-exposed IoT and OT devices requires further and careful study, employing advanced methods to improve efficiency, completeness, and reliability. On the other hand, advanced scanning methods currently require scaling resources, such as distributed deployments and complex probing approaches (e.g., dynamic or continuous scanning). An alternative to the scaling problem is to invest in collaborative methods, improving reproducibility and comparisons between datasets.

8.2 Future Work

The work presented in this thesis contributes to advancing the understanding of IoT and OT exposure by identifying systemic vulnerabilities and proposing new methods to observe and monitor their evolution. In doing so, it also highlights several limitations inherent to current measurement approaches and outlines initial steps toward addressing them. These contributions open up a range of new research opportunities, many of which remain only partially explored. While achieving complete measurements remains challenging, the following issues represent pressing open problems whose resolution would significantly benefit the field.

IoT and residential networks. The work on IoT is severely hampered by the Internet's volatile nature, which limits the coverage and accuracy of current studies and makes surveying the same devices an excessively challenging task. Nevertheless, the majority of vulnerable IoT devices are located in consumer networks [156], where society consistently fails to maintain even basic cyber-security measures [75], [76]. We have shown the volume of vulnerable devices in IoT surpasses others by orders of magnitude, and every day we are reminded of this threat with new and larger volumetric attacks from IoT botnets. On the other hand, attack vectors remain largely the same: unpatched vulnerabilities, insecure credentials, reuses, etc. Attackers only need to spray these techniques to establish a network of infected devices, and then leverage the resources from those devices to continue spreading. However, we currently do not possess an efficient method to mitigate threats in IoT consumer networks. Monitoring these networks and contacting device owners is currently unfeasible. This remains an open challenge and an increasingly relevant research direction that calls for further advancements in large-scale Internet measurements.

Monitoring OT. In a similar vein to IoT, monitoring OT networks has become critical. Understanding behavioral changes in OT networks at the Internet scale is urgent and necessary. Attacks targeting OT networks need to be detected early before they can spread and cause harm. Operators need to understand the risks they face when exposing OT systems, and require the capabilities to evaluate and mitigate these risks effectively. Similarly, monitoring OT networks at scale can help identify root causes for cyber-security weaknesses at multiple levels, enabling precise analysis of systemic, societal, and technological aspects (e.g., identify when a group of devices shows systematically more weaknesses than others).

Coverage transparency and disallow lists. A frequent point of critique is the ambiguous use of the term Internet-wide, since the implied coverage varies significantly depending on measurement design and underlying biases. In practice, measurements that follow a centralized deployment scheme will observe less than 80% of the available Internet. Scanning campaigns are severely limited by pathing and blocking restrictions, and nearly 15% of the IPv4 space is reserved for local and multicast addresses, gateway usage, backbone infrastructure, etc. On top of those restrictions, several parts of the Internet are

often skipped altogether, including known networks from governmental entities, research institutions, dark networks (e.g., telescopes and sinkholes), and other address spaces besides those that actively opt-out from this type of study. While a simple solution would be to share such disallow lists, their content can be easily used to harm those included in them. This is an underexplored area that would benefit from confidentiality schemes seen in other fields.

Methods for responsible disclosures. Internet-wide measurements can observe patterns that localized scans miss. These patterns often affect large sample groups on the Internet with dozens or hundreds of different owners and network operators. Although this is a timely issue, contacting device owners is not trivial, and recommendations often go unnoticed or ignored. Conducting responsible disclosure campaigns is a common practice in the literature and an encouraged part of Internet measurements, even when messages fail to get through. On the other hand, the sheer amount of alarms from different sources with varied advice and perspectives on the issues can cause significant fatigue, among other problems. This particular aspect of large-scale measurements is currently understudied, with only a few studies discussing alternatives and best practices [74].

8.3 Conclusion

The content of this thesis studied core concepts of active Internet measurements. The main contributions are aimed at identifying cyber-security weaknesses in exposed IoT and OT networks, providing efficient and accurate methods while narrowing the limitations of the field. Our contributions enable advanced large-scale measurements to detect systemic vulnerabilities affecting fleet-wide deployments. The content of this thesis is summarized as follows.

Chapter 3 dives into the security concerns of neglecting and abandoning devices once exposed to the Internet. Our work shows that IoT and OT networks alike suffer from common issues emerging from poor maintenance. The landscape of Internet-facing devices is filled with devices lacking authentication or encryption, and the majority allow unauthorized actors to manipulate the devices. Further, this study illustrates how devices suffer from widespread mismanagement, with most devices falling several versions behind, using default configurations, and persisting despite being in a state of obsolescence.

Chapter 4 extends our previous study with a comparison of the security landscape of IoT and OT devices with a recent measurement. The results from this study show that 25% of IoT and up to OT vulnerable devices remain unchanged even after a year post-identification. We find several cases where devices have been downgraded, and only a few instances where devices are replaced altogether. Our disclosure campaign turned mostly futile, with only five out of hundreds of notified operators responding to our notifications.

Chapter 5 explored the cyber-security state of Internet-facing OPC UA servers. Due to the complexity of the protocol, we identified that the majority of exposed services were misconfigured, with settings not suitable for Internet communications. OPC UA is one of the few ICS standards built with security in mind. However, most security features are either optional or difficult to combine to create a secure deployment.

Chapter 6 proposes a new framework to conduct advanced Internet measurements, enabling researchers to conduct efficient and accurate large-scale scans. In addition, we

propose further standardization around active probing instrumentation, stronger reproducibility measures, and new strategies to compare and contrast results.

Chapter 7 presents methods to conduct noise-aware measurements targeting OT networks. The work introduces multiple evaluation metrics to detect the presence of deception systems at scale using network and host properties. This study shows that at least 7% of observations show strong indicators of noise, and up to 20% show weaker signals.

Bibliography

[1] M. Antonakakis et al., "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*, Vancouver, BC: USENIX Association, Aug. 2017, pp. 1093–1110, ISBN: 978-1-931971-40-9. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis.

[2] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security issues in scada networks," *Computers & Security*, vol. 25, no. 7, pp. 498–506, 2006, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2006.03.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404806000514.

[3] T. Krenc, O. Hohlfeld, and A. Feldmann, "An internet census taken by an illegal botnet: A qualitative assessment of published measurements," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 103–111, Jul. 2014, ISSN: 0146-4833. DOI: 10.1145/2656877.2656893. [Online]. Available: https://doi.org/10.1145/2656877.2656893.

[4] A. Mirian et al., "An internet-wide view of ics devices," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, Auckland, New Zealand: IEEE, 2016, pp. 96–103. DOI: 10.1109/PST.2016.7906943.

[5] M. Dodson, A. R. Beresford, and D. R. Thomas, "When will my plc support mirai? the security economics of large-scale attacks against internet-connected ics devices," in *2020 APWG Symposium on Electronic Crime Research (eCrime)*, 2020, pp. 1–14. DOI: 10.1109/eCrime51433.2020.9493257.

[6] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman, "Zippier ZMap: Internet-Wide scanning at 10 gbps," in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, San Diego, CA: USENIX Association, Aug. 2014. [Online]. Available: https://www.usenix.org/conference/woot14/workshop-program/presentation/adrian.

[7] Z. Durumeric et al., "Censys: A map of internet hosts and services," in *Proceedings of the ACM SIGCOMM 2025 Conference*, ser. SIGCOMM '25, São Francisco Convent, Coimbra, Portugal: Association for Computing Machinery, 2025, pp. 147–163, ISBN: 9798400715242. DOI: 10.1145/3718958.3754344. [Online]. Available: https://doi.org/10.1145/3718958.3754344.

[8] A. Mirian et al., "An Internet-wide view of ICS devices," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, Dec. 2016, pp. 96–103. DOI: 10.1109/PST.2016.7906943.

[9] G. Wan et al., "On the Origin of Scanning: The Impact of Location on Internet-Wide Scans," en, in *Proceedings of the ACM Internet Measurement Conference*, Virtual Event USA: ACM, Oct. 2020, pp. 662–679, ISBN: 978-1-4503-8138-3. DOI: 10.1145/3419394.3424214. Accessed: Jan. 6, 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3419394.3424214.

[10] E. Bou-Harb, M. Debbabi, and C. Assi, "Cyber scanning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1496–1519, 2014.

[11] *Internet census 2012*, [Online; accessed 2025-11-26], 2012. [Online]. Available: https://census2012.sourceforge.net/paper.html.

[12] R. Yaben, N. Lundsgaard, J. August, and E. Vasilomanolakis, "Towards identifying neglected, obsolete, and abandoned iot and ot devices," eng, *Proceedings of the 8th Network Traffic Measurement and Analysis Conference (TMA Conference 2024)*, vol. 1, pp. 1–10, 2024. DOI: 10.23919/TMA62044.2024.10558996.

[13] R. Yaben and E. Vasilomanolakis, “Digital ghost ships: Abandoned, neglected, and obsolete iot & ot devices exposed to the internet,” *Authorea Preprints*, vol. 1, pp. 1–12, 2025.

[14] R. Yaben and E. Vasilomanolakis, “Drifting away: A cyber-security study of internet-exposed opc ua servers,” in *2025 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2025, pp. 195–202. DOI: 10.1109/EuroSPW67616.2025.00029.

[15] R. Yaben and E. Vasilomanolakis, “Rolling the dice: A device identification and classification engine to detect vulnerable devices facing the internet,” in *2025 9th Network Traffic Measurement and Analysis Conference (TMA)*, 2025, pp. 1–4. DOI: 10.23919/TMA66427.2025.11097013.

[16] R. Yaben, M. Anguita, and E. Vasilomanolakis, “Measuring what matters: Revisiting internet exposure of ot networks,” Journal preprint, available at SSRN, 2025. DOI: 10.2139/ssrn.5974783. [Online]. Available: https://ssrn.com/abstract=5974783.

[17] V. Paxson, “Strategies for sound internet measurement,” in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, 2004, pp. 263–271.

[18] K. Thompson, G. J. Miller, and R. Wilder, “Wide-area internet traffic patterns and characteristics,” *IEEE network*, vol. 11, no. 6, pp. 10–23, 1997.

[19] W. John, S. Tafvelin, and T. Olovsson, “Passive internet measurement: Overview and guidelines based on experiences,” *Computer Communications*, vol. 33, no. 5, pp. 533–550, 2010.

[20] M. Safaei Pour, C. Nader, K. Friday, and E. Bou-Harb, “A comprehensive survey of recent internet measurement techniques for cyber security,” *Computers & Security*, vol. 128, p. 103 123, 2023, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2023.103123. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823000330.

[21] S. F. Amin, S. Athar, A. Feldmann, H. Dao, and M. Kaur, “Navigating the ethics of internet measurement: Researchers’ perspectives from a case study in the eu,” *arXiv preprint arXiv:2511.10408*, 2025.

[22] Z. Durumeric, M. Bailey, and J. A. Halderman, “An Internet-Wide view of Internet-Wide scanning,” in *23rd USENIX Security Symposium (USENIX Security 14)*, San Diego, CA: USENIX Association, Aug. 2014, pp. 65–78, ISBN: 978-1-931971-15-7.

[23] Z. Durumeric, D. Adrian, P. Stephens, E. Wustrow, and J. A. Halderman, “Ten years of zmap,” in *Proceedings of the 2024 ACM on Internet Measurement Conference*, ser. IMC ’24, Madrid, Spain: Association for Computing Machinery, 2024, pp. 139–148.

[24] Z. Shamsi et al., “Hershel: Single-packet os fingerprinting,” *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2196–2209, Aug. 2016, ISSN: 1063-6692. DOI: 10.1109/TNET.2015.2447492. [Online]. Available: https://doi-org.proxy.findit.cvt.dk/10.1109/TNET.2015.2447492.

[25] A. Cordeiro and E. Vasilomanolakis, “Towards agnostic operational technology (ot) honeypot fingerprinting,” in *2025 9th Network Traffic Measurement and Analysis Conference (TMA)*, Copenhagen, Denmark: IEEE, 2025, pp. 1–4. DOI: 10.23919/TMA66427.2025.11097018.

[26] J. François, A. Lahmadi, V. Giannini, D. Cupif, F. Beck, and B. Wallrich, “Optimizing internet scanning for assessing industrial systems exposure,” in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016, pp. 516–522. DOI: 10.1109/IWCMC.2016.7577111.

[27] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister, “Census and survey of the visible internet,” in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’08, Vouliagmeni, Greece: Association for Computing Machinery, 2008, pp. 169–182, ISBN: 9781605583341. DOI: 10.1145/1452520.1452542. [Online]. Available: https://doi.org/10.1145/1452520.1452542.

[28] J. Margolis, T. T. Oh, S. Jadhav, Y. H. Kim, and J. N. Kim, “An in-depth analysis of the mirai botnet,” in *2017 International Conference on Software Security and Assurance (ICSSA)*, 2017, pp. 6–12. DOI: 10.1109/ICSSA.2017.12.

[29] G. Gallopeni, B. Rodrigues, M. Franco, and B. Stiller, “A practical analysis on mirai botnet traffic,” in *2020 IFIP Networking Conference (Networking)*, 2020, pp. 667–668.

[30] L. Izhikevich, R. Teixeira, and Z. Durumeric, “LZR: Identifying unexpected internet services,” in *30th USENIX Security Symposium (USENIX Security 21)*, USENIX Association, Aug. 2021, pp. 3111–3128, ISBN: 978-1-939133-24-3. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/izhikevich.

[31] S. Bano et al., “Scanning the internet for liveness,” *SIGCOMM Comput. Commun. Rev.*, vol. 48, no. 2, pp. 2–9, May 2018, ISSN: 0146-4833. DOI: 10.1145/3213232.3213234. [Online]. Available: https://doi.org/10.1145/3213232.3213234.

[32] M. Mladenov, L. Erdődi, and G. Smaragdakis, “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)*, Venice, Italy: IEEE, 2025, pp. 133–152. DOI: 10.1109/EuroSP63326.2025.00017.

[33] S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, “Gotta catch’em all: A multistage framework for honeypot fingerprinting,” *Digital Threats: Research and Practice*, vol. 4, no. 3, pp. 1–28, 2023.

[34] H. Griffioen, G. Koursiounis, G. Smaragdakis, and C. Doerr, “Have you syn me? characterizing ten years of internet scanning,” 2024, pp. 149–164.

[35] S. J. Saidi et al., “A haystack full of needles: Scalable detection of iot devices in the wild,” in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC ’20, Virtual Event, USA: Association for Computing Machinery, 2020, pp. 87–100.

[36] B. Zhao et al., “A large-scale empirical study on the vulnerability of deployed iot devices,” *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1826–1840, 2022.

[37] M. Dahlmanns, J. Lohmöller, I. B. Fink, J. Pennekamp, K. Wehrle, and M. Henze, “Easing the conscience with opc ua: An internet-wide study on insecure deployments,” in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC ’20, Virtual Event, USA: Association for Computing Machinery, 2020, pp. 101–110, ISBN: 9781450381383. DOI: 10.1145/3419394.3423666. [Online]. Available: https://doi.org/10.1145/3419394.3423666.

[38] S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, “Open for hire: Attack trends and misconfiguration pitfalls of iot devices,” in *Proceedings of the 21st ACM Internet Measurement Conference*, ser. IMC ’21, Virtual Event: Association for Computing Machinery, 2021, pp. 195–215, ISBN: 9781450391290.

[39] M. Dahlmanns, J. Lohmöller, J. Pennekamp, J. Bodenhausen, K. Wehrle, and M. Henze, “Missed opportunities: Measuring the untapped tls support in the industrial internet of things,” in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS ’22, Nagasaki, Japan: Association for Computing Machinery, 2022, pp. 252–266, ISBN: 9781450391405. DOI:

10.1145/3488932.3497762. [Online]. Available: https://doi.org/10.1145/3488932.3497762.

[40] O. Gasser, Q. Scheitle, B. Rudolph, C. Denis, N. Schricker, and G. Carle, "The amplification threat posed by publicly reachable bacnet devices," und, *Journal of Cyber Security and Mobility*, 2017, ISSN: 22454578, 22451439. DOI: 10.13052/2245-1439.614.

[41] Y. Mekdad, G. Bernieri, M. Conti, and A. El Fergougui, "The rise of ics malware: A comparative analysis," in *Computer Security. ESORICS 2021 International Workshops*, S. Katsikas et al., Eds., Cham: Springer International Publishing, 2022, pp. 496–511, ISBN: 978-3-030-95484-0.

[42] M. Alllman and V. Paxson, "Issues and etiquette concerning use of shared measurement data," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07, San Diego, California, USA: Association for Computing Machinery, 2007, pp. 135–140, ISBN: 9781595939081. DOI: 10.1145/1298306.1298327. [Online]. Available: https://doi.org/10.1145/1298306.1298327.

[43] É. Vyncke, B. Donnet, and J. Iurman, *Attribution of Internet Probes*, RFC 9511, Nov. 2023. DOI: 10.17487/RFC9511. [Online]. Available: https://www.rfc-editor.org/info/rfc9511.

[44] N. V. Dijkhuizen and J. V. D. Ham, "A survey of network traffic anonymisation techniques and implementations," *ACM Comput. Surv.*, vol. 51, no. 3, May 2018, ISSN: 0360-0300. DOI: 10.1145/3182660. [Online]. Available: https://doi.org/10.1145/3182660.

[45] I. Erkek and E. Irmak, "Cyber security of internet connected ics/scada devices and services," in *2021 International Conference on Information Security and Cryptology (ISCTURKEY)*, 2021, pp. 75–80. DOI: 10.1109/ISCTURKEY53027.2021.9654285.

[46] Q. Li, X. Feng, H. Wang, and L. Sun, "Understanding the usage of industrial control system devices on the internet," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2178–2189, 2018. DOI: 10.1109/JIOT.2018.2826558.

[47] Y. Wu et al., "Icscope: Detecting and measuring vulnerable ics devices exposed on the internet," *Communications in Computer and Information Science*, vol. 1851 CCIS, pp. 1–24, 2023. DOI: 10.1007/978-3-031-37807-2_1. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85169017103&doi=10.1007%2f978-3-031-37807-2_1&partnerID=40&md5=0956b368a950c806f6d64602df62ad41.

[48] W. Xu, Y. Tao, and X. Guan, "The landscape of industrial control systems (ics) devices on the internet," eng, *2018 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, Cybersa 2018*, vol. 1, p. 8 551 422, 2018. DOI: 10.1109/CyberSA.2018.8551422.

[49] G. Guo, J. Zhuge, M. Yang, G. Zhou, and Y. Wu, "A survey of industrial control system devices on the internet," in *2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)*, 2018, pp. 197–202. DOI: 10.1109/IINTEC.2018.8695276.

[50] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast internet-wide scanning and its security applications," in *22nd USENIX Security Symposium (USENIX Security 13)*, Washington, D.C.: USENIX Association, Aug. 2013, pp. 605–620, ISBN: 978-1-931971-03-4. [Online]. Available: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric.

[51] L. Chang, K. Lu, C. Li, and Z. Zhang, "Zmap performance in open dns resolver discovery," in *2022 2nd Asia-Pacific Conference on Communications Technology*

and Computer Science (ACCTCS), 2022, pp. 80–85. DOI: 10.1109/ACCTCS53867.2022.00024.

[52] L. Markowsky and G. Markowsky, "Scanning for vulnerable devices in the internet of things," in *2015 IEEE 8th International conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)*, IEEE, vol. 1, 2015, pp. 463–467.

[53] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019, ISSN: 1553-877X. DOI: 10.1109/COMST.2019.2910750.

[54] J. Cañedo and A. Skjellum, "Using machine learning to secure iot systems," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, 2016, pp. 219–222. DOI: 10.1109/PST.2016.7906930.

[55] Shodan, *Shodan search engine*, https://www.shodan.io/, Dec. 2025.

[56] Censys, *Censys | attack surface management*, https://go.censys.com/, Dec. 2025.

[57] GreyNoise, *Greynoise | sensors and benign scanner activity*, https://www.greynoise.io/, Dec. 2025.

[58] A. Cui and S. J. Stolfo, "A quantitative analysis of the insecurity of embedded network devices: Results of a wide-area scan," in *Proceedings of the 26th Annual Computer Security Applications Conference*, 2010, pp. 97–106.

[59] Q. Li, X. Feng, L. Zhao, and L. Sun, "A framework for searching internet-wide devices," *IEEE Network*, vol. 31, no. 6, pp. 101–107, 2017. DOI: 10.1109/MNET.2017.1700034.

[60] X. Feng, Q. Li, H. Wang, and L. Sun, "Characterizing industrial control system devices on the internet," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, IEEE, 2016, pp. 1–10.

[61] R. D. Graham, *Masscan: Mass ip port scanner*, 2013. [Online]. Available: https://github.com/robertdavidgraham/masscan.

[62] VirusTotal, *Virustotal*, https://www.virustotal.com/, Dec. 2025.

[63] T. Sasaki, A. Fujita, C. H. Gañán, M. van Eeten, K. Yoshioka, and T. Matsumoto, "Exposed infrastructures: Discovery, attacks and remediation of insecure ics remote management devices," in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 2379–2396. DOI: 10.1109/SP46214.2022.9833730.

[64] P. Jose, S. J. Saidi, and O. Gasser, "Analyzing iot hosts in the ipv6 internet," *arXiv preprint arXiv:2307.09918*, 2023.

[65] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by internet-wide scanning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15, Denver, Colorado, USA: Association for Computing Machinery, 2015, pp. 542–553, ISBN: 9781450338325.

[66] *Censys | opt out of data collection*, https://support.censys.io/hc/en-us/articles/360043177092-Opt-Out-of-Data-Collection, (Accessed on 03/13/2024).

[67] Z. Durumeric, M. Bailey, and J. A. Halderman, "An {internet-wide} view of {internet-wide} scanning," in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 65–78.

[68] F. Maggi et al., "A security analysis of the data distribution service (dds) protocol," *Trend Micro Research, Inc., Japan*, pp. 15–20, 2022.

[69] *Nvd - vulnerabilities*, https://nvd.nist.gov/vuln, (Accessed on 03/06/2024).

[70] *Xep-0438: Best practices for password hashing and storage*, https://xmpp.org/extensions/xep-0438.pdf, (Accessed on 03/13/2024).

[71] V. L. Do, L. Fillatre, I. Nikiforov, and P. Willett, "Security of scada systems against cyber–physical attacks," *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 5, pp. 28–45, 2017. DOI: 10.1109/MAES.2017.160047.

[72] DNP, *Sav6 and amp flyer 2022*, [Online; accessed 2025-10-19]. [Online]. Available: https://www.dnp.org/Portals/0/Public%20Documents/SAv6%20and%20AMP%20flyer%20-%202022%20-%20Final.pdf?ver=z_i7KIkCzZDyYSWJPhU3KA%3d%3d.

[73] M. Peacock, M. N. Johnstone, and C. Valli, "An exploration of some security issues within the bacnet protocol," in *Information Systems Security and Privacy: Third International Conference, ICISSP 2017, Porto, Portugal, February 19-21, 2017, Revised Selected Papers 3*, Springer, 2018, pp. 252–272.

[74] O. Cetin, C. Ganan, M. Korczynski, and M. Van Eeten, "Make notifications great again: Learning how to notify in the age of large-scale vulnerability scanning," in *Workshop on the Economics of Information Security (WEIS)*, vol. 23, 2017.

[75] M. Fagan and M. M. H. Khan, "Why do they do what they do?: A study of what motivates users to (not) follow computer security advice," in *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, Denver, CO: USENIX Association, Jun. 2016, pp. 59–75, ISBN: 978-1-931971-31-7. [Online]. Available: https://www.usenix.org/conference/soups2016/technical-sessions/presentation/fagan.

[76] C. Herley, "More is not the answer," *IEEE Security & Privacy*, vol. 12, no. 1, pp. 14–19, 2014. DOI: 10.1109/MSP.2013.134.

[77] C. Herley, "So long, and no thanks for the externalities: The rational rejection of security advice by users," in *Proceedings of the 2009 Workshop on New Security Paradigms Workshop*, ser. NSPW '09, Oxford, United Kingdom: Association for Computing Machinery, 2009, pp. 133–144, ISBN: 9781605588452. DOI: 10.1145/1719030.1719050.

[78] F. Li et al., "You've got vulnerability: Exploring effective vulnerability notifications," in *25th USENIX Security Symposium (USENIX Security 16)*, Austin, TX: USENIX Association, Aug. 2016, pp. 1033–1050, ISBN: 978-1-931971-32-4. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/li.

[79] C. Bellman and P. C. van Oorschot, "Best practices for iot security: What does that even mean?" *arXiv preprint arXiv:2004.12179*, 2020.

[80] A. Maurushat and K. Nguyen, "Correction to: The legal obligation to provide timely security patching and automatic updates," *International Cybersecurity Law Review*, vol. 3, no. 2, pp. 495–495, Dec. 2022, ISSN: 2662-9739. DOI: 10.1365/s43439-022-00068-5.

[81] L. L. Nielsen, "What makes iot secure? a maturity analysis of industrial product manufacturers' approaches to iot security," in *HCI for Cybersecurity, Privacy and Trust*, A. Moallem, Ed., Cham: Springer International Publishing, 2022, pp. 406–421, ISBN: 978-3-031-05563-8.

[82] RIPE, *Ripestat*, [Online; accessed 2024-01-13]. [Online]. Available: https://stat-ui.stat.ripe.net/about/.

[83] R. Hiesgen, M. Nawrocki, A. King, A. Dainotti, T. C. Schmidt, and M. Wählisch, "Spoki: Unveiling a new wave of scanners through a reactive network telescope," in *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA: USENIX Association, Aug. 2022, pp. 431–448, ISBN: 978-1-939133-31-1. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/hiesgen.

[84] O. U. Foundation, *Ua part 2: Security - 4 opc ua security architecture*, [Accessed 2025-02-25]. [Online]. Available: https://reference.opcfoundation.org/Core/Part2/v105/docs/4.

[85] O. U. Foundation, *Unified architecture - landingpage - opc foundation*, [Accessed 2025-02-25]. [Online]. Available: https://opcfoundation.org/about/opc-technologies/opc-ua/.

[86] F. O. for Information Security, *Opc ua security analysis*, [Accessed 2025-02-25], Jun. 2022. [Online]. Available: https://opcfoundation.org/wp-content/uploads/2023/11/BSI-OPCUA-2022-EN.pdf#page=10.07.

[87] N. Mühlbauer, E. Kirdan, M.-O. Pahl, and G. Carle, "Open-source opc ua security and scalability," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 262–269. DOI: 10.1109/ETFA46521.2020.9212091.

[88] A. Erba, A. Müller, and N. O. Tippenhauer, "Security analysis of vendor implementations of the opc ua protocol for industrial control systems," in *Proceedings of the 4th Workshop on CPS & IoT Security and Privacy*, ser. CPSIoTSec '22, Los Angeles, CA, USA: Association for Computing Machinery, 2022, pp. 1–13, ISBN: 9781450398763. DOI: 10.1145/3560826.3563380. [Online]. Available: https://doi.org/10.1145/3560826.3563380.

[89] IANA, *Service name and transport protocol port number registry*, [Accessed 2025-04-14]. [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?&page=86.

[90] M. Henze, J. Hiller, R. Hummen, R. Matzutt, K. Wehrle, and J. H. Ziegeldorf, "Network security and privacy for cyber-physical systems," *Security and Privacy in Cyber□Physical Systems*, pp. 25–56, Nov. 2017. DOI: 10.1002/9781119226079.CH2. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/9781119226079.ch2.

[91] A. R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," *Proceedings - Design Automation Conference*, vol. 2015-July, Jul. 2015, ISSN: 0738100X. DOI: 10.1145/2744769.2747942. [Online]. Available: https://dl.acm.org/doi/10.1145/2744769.2747942.

[92] M. Krotofil and D. Gollmann, "Industrial control systems security: What is happening?" In *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, IEEE, 2013, pp. 670–675.

[93] A. Nicholson, S. Webber, S. Dyer, T. Patel, and H. Janicke, "Scada security in the light of cyber-warfare," *Computers & Security*, vol. 31, no. 4, pp. 418–436, 2012, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2012.02.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404812000429.

[94] J. Gao et al., "Scada communication and security issues," *Security and Communication Networks*, vol. 7, pp. 175–194, 1 Jan. 2014, ISSN: 1939-0122. DOI: 10.1002/SEC.698. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/sec.698.

[95] M. Nawrocki, T. C. Schmidt, and M. Wahlisch, "Uncovering vulnerable industrial control systems from the internet core," *Proceedings of IEEE/IFIP Network Operations and Management Symposium 2020: Management in the Age of Softwarization and Artificial Intelligence, NOMS 2020*, Apr. 2020. DOI: 10.1109/NOMS47738.2020.9110256.

[96] P. Cheremushkin and S. Temnikov, "Opc ua security analysis," *Kaspersky Lab ICS CERT*, 2018.

[97] ZMap, *Zmap/zgrab2: Fast application layer scanner*, [Accessed 2025-02-25]. [Online]. Available: https://github.com/zmap/zgrab2.

[98] D. C. .-. S. for cybersecurity engineering, *Technical university of denmark (dtu) - internet scanner*, [Accessed 2025-04-12]. [Online]. Available: http://130.226.254.28/.

[99] O. Foundation, *Practical security recommendations for building opc ua applications*, [Online; accessed 2025-02-18]. [Online]. Available: https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Security-Advise-EN.pdf.

[100] O. Labs, *Providing opc ua client instance certificate*, [Accessed 2025-04-13]. [Online]. Available: https://opclabs.doc-that.com/files/onlinedocs/QuickOpc/Latest/User's%20Guide%20and%20Reference-QuickOPC/Providing%20Client%20Instance%20Certificate.html.

[101] O. U. Foundation, *Certificate generator*, Dec. 2025. [Online]. Available: https://opcfoundation.github.io/UA-.NETStandard/help/certificate_generator.htm.

[102] I. S. Agency, N. S. Agency, F. B. of Investigation, E. P. Agency, T. S. Administration, and I. Partners, *Tlp:clear secure by demand: Priority considerations for operational technology owners and operators when selecting digital products*, 2025.

[103] M. Dahlmanns, *Dataset to "easing the conscience with opc ua: An internet-wide study on insecure deployments" - rwth publications*, [Online; accessed 2025-02-19]. [Online]. Available: https://publications.rwth-aachen.de/record/802060.

[104] AbuseIPDB, *Abuseipdb - ip address abuse reports - making the internet safer, one ip at a time*, Dec. 2025. [Online]. Available: https://www.abuseipdb.com/.

[105] F. Kohnhauser, D. Meier, F. Patzer, and S. Finster, "On the security of iiot deployments: An investigation of secure provisioning solutions for opc ua," *IEEE Access*, vol. 9, pp. 99 299–99 311, 2021, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3096062.

[106] V. Paxson, "Strategies for sound internet measurement," in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '04, Taormina, Sicily, Italy: Association for Computing Machinery, 2004, pp. 263–271, ISBN: 1581138210.

[107] Z. Durumeric et al., "The matter of heartbleed," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14, Vancouver, BC, Canada: Association for Computing Machinery, 2014, pp. 475–488, ISBN: 9781450332132.

[108] A. Mirian et al., "An internet-wide view of ics devices," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, 2016, pp. 96–103.

[109] J. Rüth, T. Zimmermann, and O. Hohlfeld, "Hidden treasures – recycling large-scale internet measurements to study the internet's control plane," in *Passive and Active Measurement*, D. Choffnes and M. Barcellos, Eds., Cham: Springer International Publishing, 2019, pp. 51–67.

[110] H. Kim, T. Kim, and D. Jang, "An intelligent improvement of internet-wide scan engine for fast discovery of vulnerable iot devices," *Symmetry*, vol. 10, no. 5, 2018, ISSN: 2073-8994.

[111] S. Morishita et al., "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)*, Arlington, VA, USA: IEEE, 2019, pp. 134–143.

[112] R. Singla, S. Srinivasa, N. Reddy, J. M. Pedersen, E. Vasilomanolakis, and R. Bettati, "An analysis of war impact on ukrainian critical infrastructure through network measurements," in *2023 7th Network Traffic Measurement and Analysis Conference (TMA)*, 2023, pp. 1–10.

[113] K. Claffy et al., “Workshop on overcoming measurement barriers to internet research (wombir 2021) final report,” *SIGCOMM Comput. Commun. Rev.*, vol. 51, no. 3, pp. 33–40, Jul. 2021, ISSN: 0146-4833.

[114] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis, “An architecture for large scale internet measurement,” *IEEE Communications Magazine*, vol. 36, no. 8, pp. 48–54, 1998.

[115] R. Yaben and E. Vasilomanolakis, *Ricyaben/tma-2025-poster: Early implementation of DICE for the TMA conference (poster session)*, May 2025. [Online]. Available: https://github.com/RicYaben/tma-2025-poster.

[116] S. Ghosh and S. Sampalli, “A survey of security in scada networks: Current issues and future challenges,” *IEEE Access*, vol. 7, pp. 135 812–135 831, 2019. DOI: 10.1109/ACCESS.2019.2926441.

[117] K. Coffey, R. Smith, L. Maglaras, and H. Janicke, “Vulnerability analysis of network scanning on scada systems,” *Security and Communication Networks*, vol. 2018, no. 1, p. 3 794 603, 2018.

[118] C.-W. Ten, C.-C. Liu, and G. Manimaran, “Vulnerability assessment of cybersecurity for scada systems,” *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1836–1846, 2008.

[119] D. Upadhyay and S. Sampalli, “Scada (supervisory control and data acquisition) systems: Vulnerability assessment and security recommendations,” *Computers & Security*, vol. 89, p. 101 666, 2020.

[120] S. Karnouskos, “Stuxnet worm impact on industrial cyber-physical system security,” in *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, Melbourne, VIC, Australia: IEEE, 2011, pp. 4490–4494. DOI: 10.1109/IECON.2011.6120048.

[121] R. Langner, “Stuxnet: Dissecting a cyberwarfare weapon,” *IEEE security & privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[122] R. Khan, P. Maynard, K. McLaughlin, D. Laverty, and S. Sezer, “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*, ser. ICS-CSR ’16, Belfast, United Kingdom: BCS Learning & Development Ltd., 2016, pp. 1–11, ISBN: 9781780173573. DOI: 10.14236/ewic/ICS2016.7. [Online]. Available: https://doi.org/10.14236/ewic/ICS2016.7.

[123] P. Kozak, I. Klaban, and T. Šlajs, “Industroyer cyber-attacks on ukraine’s critical infrastructure,” in *2023 International Conference on Military Technologies (ICMT)*, Brno, Czech Republic: IEEE, 2023, pp. 1–6. DOI: 10.1109/ICMT58149.2023.10171308.

[124] A. Adamov, A. Carlsson, and T. Surmacz, “An analysis of lockergoga ransomware,” in *2019 IEEE East-West Design & Test Symposium (EWDTS)*, Batumi, Georgia: IEEE, 2019, pp. 1–5. DOI: 10.1109/EWDTS.2019.8884472.

[125] R. Yaben, E. Vasilomanolakis, and M. Anguita, *Measuring what matters: Revisiting internet exposure of ot networks*, Technical University of Denmark, Zenodo, Dec. 2025. DOI: 10.5281/zenodo.17977303. [Online]. Available: https://doi.org/10.5281/zenodo.17977303.

[126] RIPE, *Ripe atlas*, [Online; accessed 2025-12-12], Dec. 2025. [Online]. Available: https://atlas.ripe.net/.

[127] X. Feng, Q. Li, H. Wang, and L. Sun, “Characterizing industrial control system devices on the internet,” eng, *Proceedings - International Conference on Network*

Protocols, Icnp, vol. 2016-, p. 7 784 407, 2016, ISSN: 10921648, 26433303. DOI: 10.1109/ICNP.2016.7784407.

[128] R. Singla, S. Srinivasa, N. Reddy, J. M. Pedersen, E. Vasilomanolakis, and R. Bettati, "An analysis of war impact on ukrainian critical infrastructure through network measurements," in *2023 7th Network Traffic Measurement and Analysis Conference (TMA)*, Naples, Italy: IEEE, 2023, pp. 1–10. DOI: 10.23919/TMA58422.2023.10199005.

[129] N. DeMarinis, S. Tellex, V. P. Kemerlis, G. Konidaris, and R. Fonseca, "Scanning the internet for ros: A view of security in robotics research," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada: IEEE, 2019, pp. 8514–8521. DOI: 10.1109/ICRA.2019.8794451.

[130] J. Knight, J. Davidson, A. Nguyen-Tuong, J. Hiser, et al., "Diversity in cybersecurity," *Computer*, vol. 49, no. 04, pp. 94–98, 2016.

[131] A. Männel et al., "Lessons learned from operating a large network telescope," in *Proceedings of the ACM SIGCOMM 2025 Conference*, ser. SIGCOMM '25, São Francisco Convent, Coimbra, Portugal: Association for Computing Machinery, 2025, pp. 826–841, ISBN: 9798400715242. DOI: 10.1145/3718958.3754347. [Online]. Available: https://doi.org/10.1145/3718958.3754347.

[132] L. Metongnon and R. Sadre, "Beyond telnet: Prevalence of iot protocols in telescope and honeypot measurements," in *Proceedings of the 2018 Workshop on Traffic Measurements for Cybersecurity*, ser. WTMC '18, Budapest, Hungary: Association for Computing Machinery, 2018, pp. 21–26, ISBN: 9781450359108. DOI: 10.1145/3229598.3229604. [Online]. Available: https://doi.org/10.1145/3229598.3229604.

[133] J. Francois, O. Festor, et al., "Activity monitoring for large honeynets and network telescopes," *International Journal on Advances in Systems and Measurements*, vol. 1, no. 1, pp. 1–13, 2008.

[134] M. S. Pour, J. Khoury, and E. Bou-Harb, "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, 2022, pp. 1–9. DOI: 10.1109/NOMS54207.2022.9789827. [Online]. Available: https://doi.org/10.1109/NOMS54207.2022.9789827.

[135] K. Borders, L. Falk, and A. Prakash, "Openfire: Using deception to reduce network attacks," in *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, Nice, France: IEEE, 2007, pp. 224–233. DOI: 10.1109/SECCOM.2007.4550337.

[136] M.-R. Zamiri-Gourabi, A. R. Qalaei, and B. A. Azad, "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*, ser. ICSS, San Juan, PR, USA: Association for Computing Machinery, 2019, pp. 30–37, ISBN: 9781450377195. DOI: 10.1145/3372318.3372322. [Online]. Available: https://doi.org/10.1145/3372318.3372322.

[137] S. Maesschalck, V. Giotsas, and N. Race, "World wide ics honeypots: A study into the deployment of conpot honeypots," in *Industrial Control System Security Workshop*, virtual: ICSS, 2021, pp. 1–10.

[138] M. Foundation, *Conpot: Ics/scada honeypot*, [Online; accessed 2025-11-26], Dec. 2025. [Online]. Available: https://github.com/mushorg/conpot.

[139] UHH-ISS, *Uhh-iss/honeygrove: A multi-purpose, modular medium-interaction honeypot based on twisted.* [Online; accessed 2025-11-27], Dec. 2025. [Online]. Available: https://github.com/UHH-ISS/honeygrove?tab=readme-ov-file.

[140] J.-H. Cho et al., “Toward proactive, adaptive defense: A survey on moving target defense,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 709–745, 2020.

[141] C. Lei, H.-Q. Zhang, J.-L. Tan, Y.-C. Zhang, and X.-H. Liu, “Moving target defense techniques: A survey,” *Security and Communication Networks*, vol. 2018, no. 1, p. 3 759 626, 2018.

[142] H. Griffioen and C. Doerr, “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)*, San Francisco, CA, USA: IEEE, 2023, pp. 2551–2565. DOI: 10.1109/SP46215.2023.10179467.

[143] L. Alt, R. Beverly, and A. Dainotti, “Uncovering network tarpits with degreaser,” in *Proceedings of the 30th Annual Computer Security Applications Conference*, ser. ACSAC ’14, New Orleans, Louisiana, USA: Association for Computing Machinery, 2014, pp. 156–165, ISBN: 9781450330053. DOI: 10.1145/2664243.2664285. [Online]. Available: https://doi.org/10.1145/2664243.2664285.

[144] S. Walla and C. Rossow, “Malpity: Automatic identification and exploitation of tarpit vulnerabilities in malware,” in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, Stockholm, Sweden: IEEE, 2019, pp. 590–605. DOI: 10.1109/EuroSP.2019.00049.

[145] ODVA, *The common industrial protocol (cip™) and the family of cip networks*, [Online; accessed 2025-10-20], Feb. 2016. [Online]. Available: https://www.odva.org/wp-content/uploads/2020/06/PUB00123R1_Common-Industrial_Protocol_and_Family_of_CIP_Networks.pdf.

[146] ODVA, *Odva vendor id data*, [Online; accessed 2025-08-30], Dec. 2025. [Online]. Available: https://marketplace.odva.org/vid.dat.

[147] R. Automation, *Rockwell - ethernet/ip network devices*, [Online; accessed 2025-08-30], Dec. 2025. [Online]. Available: https://literature.rockwellautomation.com/idc/groups/literature/documents/um/enet-um006_-en-p.pdf.

[148] ODVA, *Securing ethernet/ip™ networks*, [Online; accessed 2025-09-01], Dec. 2011. [Online]. Available: https://www.odva.org/wp-content/uploads/2020/05/PUB00269R1.1_ODVA-Securing-EtherNetIP-Networks.pdf#page=6.66.

[149] D. Myers, E. Foo, and K. Radke, “Internet-wide scanning taxonomy and framework,” eng, *Conferences in Research and Practice in Information Technology Series*, vol. 161, pp. 61–65, 2015, ISSN: 14451336.

[150] Scopus, *Scopus - document search results*, [Online; accessed 2025-12-14], Dec. 2025. [Online]. Available: https://bit.ly/44saqQd.

[151] C. Fachkha, E. Bou-Harb, A. Keliris, N. D. Memon, and M. Ahamad, “Internet-scale probing of cps: Inference, characterization and orchestration analysis,” in *NDSS*, San Diego, CA, USA: NDSS, 2017, pp. 1–15.

[152] G. Barbieri, M. Conti, N. O. Tippenhauer, and F. Turrin, “Assessing the use of insecure ics protocols via ixp network traffic analysis,” in *2021 International Conference on Computer Communications and Networks (ICCCN)*, Athens, Greece: IEEE, 2021, pp. 1–9. DOI: 10.1109/ICCCN52240.2021.9522219.

[153] S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, “Interaction matters: A comprehensive analysis and a dataset of hybrid iot/ot honeypots,” in *Proceedings of the 38th Annual Computer Security Applications Conference*, ser. ACSAC ’22, Austin, TX, USA: Association for Computing Machinery, 2022, pp. 742–755, ISBN: 9781450397599. DOI: 10.1145/3564625.3564645. [Online]. Available: https://doi.org/10.1145/3564625.3564645.

[154] A. Jicha, M. Patton, and H. Chen, “Scada honeypots: An in-depth analysis of conpot,” in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, Tucson, AZ, USA: IEEE, 2016, pp. 196–198. DOI: 10.1109/ISI.2016.7745468.

[155] T. Ashley, S. N. G. Gourisetti, N. Brown, and C. Bonebrake, “Aggregate attack surface management for network discovery of operational technology,” *Computers & Security*, vol. 123, p. 102 939, 2022, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2022.102939. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404822003315.

[156] A. Mangino, M. S. Pour, and E. Bou-Harb, “Internet-scale insecurity of consumer internet of things: An empirical measurements perspective,” *ACM Trans. Manage. Inf. Syst.*, vol. 11, no. 4, Oct. 2020, ISSN: 2158-656X. DOI: 10.1145/3394504. [Online]. Available: https://doi.org/10.1145/3394504.

[157] M. Antonakakis et al., “Understanding the mirai botnet,” in *26th USENIX security symposium (USENIX Security 17)*, 2017, pp. 1093–1110.

[158] S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, “Deceptive directories and “vulnerable” logs: A honeypot study of the ldap and log4j attack landscape,” in *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2022, pp. 442–447. DOI: 10.1109/EuroSPW55150.2022.00052.

[159] J. Zheng and A. Namin, “A survey on the moving target defense strategies: An architectural perspective,” *Journal of Computer Science and Technology*, vol. 34, pp. 207–233, 1 2019. DOI: 10.1007/s11390-019-1906-z.

[160] F. Zhang, S. Zhou, Z. Qin, and J. Liu, “Honeypot: A supplemented active defense system for network security,” in *Parallel and Distributed Computing, Applications and Technologies, PDCAT Proceedings*, 2003, pp. 231–235.

[161] J.-H. Cho et al., “Toward proactive, adaptive defense: A survey on moving target defense,” *IEEE Communications Surveys and Tutorials*, vol. 22, pp. 709–745, 1 2020. DOI: 10.1109/COMST.2019.2963791.

[162] N. Saputro, “A brief review on network identity-based moving target defense,” in *International Conference on Information Networking*, vol. 2023-Janua, 2023, pp. 610–615, ISBN: 9781665462686. DOI: 10.1109/ICOIN56518.2023.10048901.

[163] M. Khosravi-Farmad, A. Ramaki, and A. Bafghi, “Moving target defense against advanced persistent threats for cybersecurity enhancement,” in *2018 8th International Conference on Computer and Knowledge Engineering, ICCKE 2018*, 2018, pp. 280–285, ISBN: 9781538695692. DOI: 10.1109/ICCKE.2018.8566531.

[164] A. Javadpour, F. Ja’fari, T. Taleb, M. Shojafar, and C. Benzaïd, “A comprehensive survey on cyber deception techniques to improve honeypot performance,” *Computers and Security*, vol. 140, 2024. DOI: 10.1016/j.cose.2024.103792.

[165] M. Bercovitch, M. Renford, L. Hasson, A. Shabtai, L. Rokach, and Y. Elovici, “Honeygen: An automated honeytokens generator,” in *Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics*, IEEE, 2011, pp. 131–136.

[166] L. Spitzner, *Honeypots: tracking hackers*. Addison-Wesley Longman Publishing Co., Inc., 2002.

[167] L. Spitzner, “The honeynet project: Trapping the hackers,” *IEEE Security & Privacy*, vol. 1, no. 2, pp. 15–23, 2003.

[168] K. D. Mitnick and W. L. Simon, *The art of deception: Controlling the human element of security*. John Wiley & Sons, 2003.

[169] E. Cranford, C. Gonzalez, P. Aggarwal, S. Cooney, M. Tambe, and C. Lebiere, “Adaptive cyber deception: Cognitively informed signaling for cyber defense,” 2020.

[170] A. Shinde and P. Doshi, “Modeling cognitive biases in decision-theoretic planning for active cyber deception,” in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 2024, pp. 1718–1726.
[171] E. A. Cranford, C. Gonzalez, P. Aggarwal, M. Tambe, S. Cooney, and C. Lebiere, “Towards a cognitive theory of cyber deception,” *Cognitive Science*, vol. 45, no. 7, e13013, 2021.
[172] P. statement, *Prisma statement*, 2025. [Online]. Available: https://www.prisma-statement.org/.
[173] T. M. Corporation, *ENGAGE Matrix*, https://engage.mitre.org/matrix/, Accessed: 28 November 2025, 2025.
[174] T. M. Corporation, *ATT&CK Matrix for Enterprise*, https://attack.mitre.org/matrices/enterprise/, Accessed: 28 November 2025, 2025.
[175] G. Lame, “Systematic literature reviews: An introduction,” *Proceedings of the Design Society: International Conference on Engineering Design*, vol. 1, no. 1, pp. 1633–1642, 2019. DOI: 10.1017/dsi.2019.169.
[176] A. Miguel-Diez et al., *Exploring the Landscape of Honeypots in the Fight Against Cyber Threats: A Systematic Mapping of Literature*. 2024, vol. 957 LNNS, pp. 179–190, ISBN: 9783031750151. DOI: 10.1007/978-3-031-75016-8_17.
[177] M. J. Grant and A. Booth, “A typology of reviews: An analysis of 14 review types and associated methodologies,” *Health Information and Libraries Journal*, vol. 26, no. 2, pp. 91–108, 2009. DOI: 10.1111/j.1471-1842.2009.00848.x.
[178] B. Kitchenham and P. Brereton, “A systematic review of systematic review process research in software engineering,” *Information and software technology*, vol. 55, no. 12, pp. 2049–2075, 2013.
[179] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic mapping studies in software engineering,” in *12th international conference on evaluation and assessment in software engineering (EASE)*, BCS Learning & Development, 2008.
[180] Scopus, *Scopus*, [Online; accessed 2025-03-11], Mar. 2025. [Online]. Available: https://www.scopus.com/.
[181] Critical Appraisal Skills Programme (CASP), *Casp checklists: Making sense of evidence*, Accessed January 2025, Oxford: CASP, 2018. [Online]. Available: https://casp-uk.net/casp-tools-checklists/.
[182] S. Keele et al., “Guidelines for performing systematic literature reviews in software engineering,” Technical report, ver. 2.3 ebse technical report. ebse, Tech. Rep., 2007.
[183] A. Carrera-Rivera, W. Ochoa, F. Larrinaga, and G. Lasa, “How-to conduct a systematic literature review: A quick guide for computer science research,” *MethodsX*, vol. 9, p. 101 895, 2022.
[184] M. Torquato and M. Vieira, “Moving target defense in cloud computing: A systematic mapping study,” *Computers and Security*, vol. 92, 2020. DOI: 10.1016/j.cose.2020.101742.
[185] M. M. Efendi, Z.-A. Ibrahim, M. A. Zawawi, F. A. Rahim, N. M. Pahri, and A. Ismail, “A survey on deception techniques for securing web application,” in *Proceedings - 5th IEEE International Conference on Big Data Security on Cloud, BigDataSecurity 2019, 5th IEEE International Conference on High Performance and Smart Computing, HPSC 2019 and 4th IEEE International Conference on Intelligent Data and Securit*, 2019, pp. 328–331, ISBN: 9781728100067. DOI: 10.1109/BigDataSecurity-HPSC-IDS.2019.00066.

[186] X. Han, N. Kheir, and D. Balzarotti, “Deception techniques in computer security: A research perspective,” *ACM Computing Surveys*, vol. 51, 4 2019. DOI: 10.1145/3214305.

[187] M. McQueen and W. Boyer, “Deception used for cyber defense of control systems,” in *Proceedings - 2009 2nd Conference on Human System Interactions, HSI ’09*, 2009, pp. 624–631, ISBN: 9781424439607. DOI: 10.1109/HSI.2009.5091050.

[188] N. Al-Gharabally, N. El-Sayed, S. Al-Mulla, and I. Ahmad, “Wireless honeypots: Survey and assessment,” in *Proceedings of the 2009 Conference on Information Science, Technology and Applications, ISTA ’09*, 2009, pp. 45–52, ISBN: 9781605584782. DOI: 10.1145/1551950.1551969.

[189] L. Ge et al., “Toward effectiveness and agility of network security situational awareness using moving target defense (mtd),” in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 9085, 2014, ISBN: 9781628410228. DOI: 10.1117/12.2050782.

[190] K. Heckman and F. Stech, *Cyber counterdeception: How to detect denial & deception (D&D)*. 2015, vol. 56, pp. 103–140. DOI: 10.1007/978-3-319-14039-1_6.

[191] S.-W. Fang, A. Portante, and M. Husain, *Moving target defense mechanisms in cyber-physical systems*. 2015, pp. 63–90, ISBN: 9781498700993. DOI: 10.1201/b19311.

[192] K. Farris and G. Cybenko, “Quantification of moving target cyber defenses,” in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 9456, 2015, ISBN: 9781628415728. DOI: 10.1117/12.2182176.

[193] G. Briskin, D. Fayette, N. Evancich, V. Rajabian-Schwart, A. Macera, and J. Li, *Design considerations for building cyber deception systems*. 2016, pp. 69–95, ISBN: 9783319326993. DOI: 10.1007/978-3-319-32699-3_4.

[194] G.-L. Cai, B.-S. Wang, W. Hu, and T.-Z. Wang, “Moving target defense: State of the art and characteristics,” *Frontiers of Information Technology and Electronic Engineering*, vol. 17, pp. 1122–1153, 11 2016. DOI: 10.1631/FITEE.1601321.

[195] S. Hassan and R. Guha, “Modelling of the state of systems with defensive deception,” in *Proceedings - 2016 International Conference on Computational Science and Computational Intelligence, CSCI 2016*, 2017, pp. 1031–1036, ISBN: 9781509055104. DOI: 10.1109/CSCI.2016.0197.

[196] V. Urias, W. Stout, J. Luc-Watson, C. Grim, L. Liebrock, and M. Merza, “Technologies to enable cyber deception,” in *Proceedings - International Carnahan Conference on Security Technology*, vol. 2017-Octob, 2017, pp. 1–6, ISBN: 9781538615850. DOI: 10.1109/CCST.2017.8167793.

[197] C. D. Faveri, A. Moreira, and E. Souza, “Deception planning models for cyber security,” in *Proceedings of the 2017 17th International Conference on Computational Science and Its Applications, ICCSA 2017*, 2017, ISBN: 9781538638934. DOI: 10.1109/ICCSA.2017.8000014.

[198] C. Lei, H.-Q. Zhang, J.-L. Tan, Y.-C. Zhang, and X.-H. Liu, “Moving target defense techniques: A survey,” *Security and Communication Networks*, vol. 2018, 2018. DOI: 10.1155/2018/3759626.

[199] P. Chen et al., *MTD techniques for memory protection against zero-day attacks*. 2019, vol. 11830 LNCS, pp. 129–155. DOI: 10.1007/978-3-030-30719-6_7.

[200] J. Pawlick, E. Colbert, and Q. Zhu, “A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy,” *ACM Computing Surveys*, vol. 52, 4 2020. DOI: 10.1145/3337772.

[201] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, “A survey of moving target defenses for network security,” *IEEE Commu-*

nications Surveys and Tutorials, vol. 22, pp. 1909–1941, 3 2020. DOI: 10.1109/COMST.2020.2982955.

[202] C. Gonzalez, P. Aggarwal, E. Cranford, and C. Lebiere, "Design of dynamic and personalized deception: A research framework and new insights for cyberdefense," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, vol. 2020-Janua, 2020, pp. 1825–1834, ISBN: 9780998133133.

[203] R. Navas, F. Cuppens, N. B. Cuppens, L. Toutain, and G. Papadopoulos, "Mtd, where art thou? a systematic review of moving target defense techniques for iot," *IEEE Internet of Things Journal*, vol. 8, pp. 7818–7832, 10 2021. DOI: 10.1109/JIOT.2020.3040358.

[204] H. Zhang, B. Liu, and H. Wu, "Smart grid cyber-physical attack and defense: A review," *IEEE Access*, vol. 9, pp. 29 641–29 659, 2021. DOI: 10.1109/ACCESS.2021.3058628.

[205] L. Zhang and V. Thing, "Three decades of deception techniques in active cyber defense - retrospect and outlook," *Computers and Security*, vol. 106, 2021. DOI: 10.1016/j.cose.2021.102288.

[206] M. Zhu, A. Anwar, Z. Wan, J.-H. Cho, C. Kamhoua, and M. Singh, "A survey of defensive deception: Approaches using game theory and machine learning," *IEEE Communications Surveys and Tutorials*, vol. 23, pp. 2460–2493, 4 2021. DOI: 10.1109/COMST.2021.3102874.

[207] Ł. Jalowski, M. Zmuda, and M. Rawski, "A survey on moving target defense for networks: A practical view," *Electronics (Switzerland)*, vol. 11, 18 2022. DOI: 10.3390/electronics11182886.

[208] Y. Zheng, Z. Li, X. Xu, and Q. Zhao, "Dynamic defenses in cyber security: Techniques, methods and challenges," *Digital Communications and Networks*, vol. 8, pp. 422–435, 4 2022. DOI: 10.1016/j.dcan.2021.07.006.

[209] P. Mohan, S. Dixit, A. Gyaneshwar, U. Chadha, K. Srinivasan, and J. Seo, "Leveraging computational intelligence techniques for defensive deception: A review, recent advances, open problems and future directions," *Sensors*, vol. 22, 6 2022. DOI: 10.3390/s22062194.

[210] M. Nguyen and S. Debroy, "Moving target defense-based denial-of-service mitigation in cloud environments: A survey," *Security and Communication Networks*, vol. 2022, 2022. DOI: 10.1155/2022/2223050.

[211] B. Amro, S. Salah, and M. Moreb, "A comprehensive architectural framework of moving target defenses against ddos attacks," *Journal of Cyber Security and Mobility*, vol. 12, pp. 605–627, 4 2023. DOI: 10.13052/jcsm2245-1439.1248.

[212] R. Sun, Y. Zhu, J. Fei, and X. Chen, "A survey on moving target defense: Intelligently affordable, optimized and self-adaptive," *Applied Sciences (Switzerland)*, vol. 13, 9 2023. DOI: 10.3390/app13095367.

[213] J. Tan et al., "A survey: When moving target defense meets game theory," *Computer Science Review*, vol. 48, 2023. DOI: 10.1016/j.cosrev.2023.100544.

[214] A. Deshpande and S. Gupta, "Genai in the cyber kill chain: A comprehensive review of risks, threat operative strategies and adaptive defense approaches," in *3rd IEEE International Conference on ICT in Business Industry and Government, ICTBIG 2023*, 2023, ISBN: 9798350343274. DOI: 10.1109/ICTBIG59752.2023.10456106.

[215] X. Qin, F. Jiang, M. Cen, and R. Doss, "Hybrid cyber defense strategies using honey-x: A survey," *Computer Networks*, vol. 230, 2023. DOI: 10.1016/j.comnet.2023.109776.

[216] D. L. Antunes and S. L. Sanchez, “The age of fighting machines: The use of cyber deception for adversarial artificial intelligence in cyber defence,” in *ACM International Conference Proceeding Series*, 2023, ISBN: 9798400707728. DOI: 10.1145/3600160.3605077.

[217] H. Yi, F. Li, R. Wang, N. Hu, and Z. Tian, “A survey of deception defense: Approaches used to counter malicious behavior,” in *2023 IEEE 12th International Conference on Cloud Networking, CloudNet 2023*, 2023, pp. 418–422, ISBN: 9798350313062. DOI: 10.1109/CloudNet59005.2023.10490043.

[218] K. Silaen, M. Meyliana, H. Warnars, H. Prabowo, A. Hidayanto, and M. Anggreainy, “Usefulness of honeypots towards data security: A systematic literature review,” in *IWAIIP 2023 - Conference Proceeding: International Workshop on Artificial Intelligence and Image Processing*, 2023, pp. 422–427, ISBN: 9798350382914. DOI: 10.1109/IWAIIP58158.2023.10462777.

[219] P. Chouhan and G. Aujla, “Deception technology for active defence: Background and opportunities,” in *2024 IEEE International Conference on Communications Workshops, ICC Workshops 2024*, 2024, pp. 1183–1188, ISBN: 9798350304053. DOI: 10.1109/ICCWorkshops59551.2024.10615759.

[220] S.-H. Lee, K. Kim, Y. Kim, and K.-W. Park, “Mtd-diorama: Moving target defense visualization engine for systematic cybersecurity strategy orchestration,” *Sensors*, vol. 24, 13 2024. DOI: 10.3390/s24134369.

[221] D. Commey, B. Mai, S. Hounsinou, and G. Crosby, “Securing blockchain-based iot systems: A review,” *IEEE Access*, vol. 12, pp. 98 856–98 881, 2024. DOI: 10.1109/ACCESS.2024.3428490.

[222] A. Abdi et al., “Security control and data planes of sdn: A comprehensive review of traditional, ai, and mtd approaches to security solutions,” *IEEE Access*, vol. 12, pp. 69 941–69 980, 2024. DOI: 10.1109/ACCESS.2024.3393548.

[223] D. Haighton and S. Leblanc, “Game theory applied to deception in network security,” in *1st International Conference on Computing, Internet of Things and Microwave Systems, ICCIMS 2024*, 2024, ISBN: 9798350351736. DOI: 10.1109/ICCIMS61672.2024.10690784.

[224] D. Georgoulias, J. M. Pedersen, M. Falch, and E. Vasilomanolakis, “Botnet business models, takedown attempts, and the darkweb market: A survey,” *ACM Comput. Surv.*, vol. 55, no. 11, Feb. 2023, ISSN: 0360-0300. DOI: 10.1145/3575808. [Online]. Available: https://doi.org/10.1145/3575808.

[225] S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, “Gotta catch ’em all: A multistage framework for honeypot fingerprinting,” *Digital Threats*, vol. 4, no. 3, Oct. 2023. DOI: 10.1145/3584976. [Online]. Available: https://doi.org/10.1145/3584976.

[226] A. Vetterl and R. Clayton, “Bitter harvest: Systematically fingerprinting low- and medium-interaction honeypots at internet scale,” in *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, Baltimore, MD: USENIX Association, Aug. 2018.

[227] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving target defense: creating asymmetric uncertainty for cyber threats*. Springer Science & Business Media, 2011, vol. 54.

[228] E. M. Hutchins, M. J. Cloppert, R. M. Amin, et al., “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains,” *Leading Issues in Information Warfare & Security Research*, vol. 1, no. 1, p. 80, 2011.

[229] F. Araujo, K. W. Hamlen, S. Biedermann, and S. Katzenbeisser, “From patches to honey-patches: Lightweight attacker misdirection, deception, and disinformation,”

in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 942–953.
[230] A. Juels and T. Ristenpart, “Honey encryption: Encryption beyond the brute-force barrier,” *IEEE security & privacy*, vol. 12, no. 4, pp. 59–62, 2014.
[231] A. M. Mongardini, M. La Morgia, S. Jajodia, L. V. Mancini, and A. Mei, “Dard: Deceptive approaches for robust defense against ip theft,” *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 5591–5606, 2024.
[232] B. Whaley, *Stratagem: deception and surprise in war*. Artech, 2007.

A A Systematic Meta-Survey of Cyber Deception: Unified Taxonomy and Research Directions

A.1 Introduction

The modern digital landscape is defined by a persistent and asymmetric conflict between attackers and defenders. For decades, the foundational paradigm of cybersecurity has relied on a *castle-and-moat* model centered on static defenses such as firewalls, intrusion detection systems, and antivirus software. However, a series of high-impact cyber attacks has systematically demonstrated that this defensive model is fundamentally inadequate against modern threats: the MIRAI botnet weaponized Internet of Things (IoT) devices for massive distributed denial-of-service attacks [157], large-scale ransomware outbreaks such as WannaCry caused widespread disruption, the SolarWinds (Sunburst) campaign subverted trusted software updates, the TRITON malware targeted industrial safety systems, and the Log4Shell vulnerability exposed a near-universal attack surface [158].

These incidents collectively reveal a fundamental truth about modern cybersecurity: Static prevention-centric defenses are necessary but insufficient. The security community faces an inherent asymmetry. Defenders must successfully protect every potential entry point and asset, while attackers need only identify and exploit a single weakness [159], [160], [161]. Consequently, the field has increasingly adopted an *assume breach* mindset that acknowledges perimeter penetration as inevitable [162], [163]. In response, cybersecurity research has shifted toward dynamic and proactive defense strategies designed to reverse this asymmetry. Cyber-deception has emerged as a promising paradigm that moves beyond constructing impenetrable barriers to actively misleading, confusing, and entrapping adversaries who have already penetrated the network perimeter [164]. Drawing on centuries of military deception doctrine, cyber-deception involves the deliberate introduction of false, misleading, or ambiguous information and resources to manipulate attacker perceptions, decisions, and actions [161], [164].

The cyber-deception landscape spans a diverse array of techniques serving distinct strategic objectives. Honey tokens [165] and other fake credentials, database entries, or API keys act as high-fidelity tripwires for detection; interactive honeypots [166] and honeynets [167] provide controlled environments to observe and analyze adversary behavior [160]; tarpits deliberately slow connection attempts to exhaust automated scanning tools and botnets [143]; and Moving Target Defense (MTD) strategies dynamically alter the attack surface, invalidating reconnaissance data and forcing adversaries to re-scan and re-target systems [161], [162]. Beyond these technical mechanisms, advanced deception strategies exploit human cognitive biases through breadcrumbing [168], signaling [169], and psychological manipulation to guide attacker behavior toward defensive objectives [170], [171]. We provide a detailed overview and systematization of these in Sec. A.3.

Over the past two decades, both academic research and practical deployment of cyber-deception techniques have grown substantially, generating an extensive body of primary research literature. This expansion has, in turn, spawned a significant secondary wave of survey papers, systematic reviews, and mapping studies attempting to organize and synthesize the field's knowledge. However, this survey literature itself has become increasingly fragmented and difficult to navigate. Researchers and practitioners entering

the field now confront dozens of surveys, each employing different scopes, incompatible classification schemes, and varying levels of methodological rigor.

This fragmentation creates substantial obstacles to developing comprehensive understanding. Certain subfields, particularly MTD and honeypots, have received extensive survey coverage, with multiple papers examining overlapping topics from different perspectives. Conversely, important deception techniques including tarpits, honey encryption, and bias exploitation strategies remain largely absent from systematic analysis. Different surveys propose incompatible taxonomic structures, making it difficult to compare findings or integrate insights across the literature. Many surveys focus primarily on enabling methodologies such as game theory or machine learning applications rather than comprehensively examining deception techniques themselves. Furthermore, methodological quality varies dramatically, with numerous surveys lacking explicit literature search strategies, inclusion criteria, or quality assessment procedures, limitations that compromise their reliability and reproducibility.

To our knowledge, no prior work has provided a systematic meta-analysis to synthesize this fragmented survey landscape and develop a holistic understanding of how cyber-deception research has been organized, what gaps persist, and where the field should direct future efforts. Our goal is not to resurvey all primary deception research, but to critically examine and unify the existing survey layer itself, which has become a bottleneck for understanding the field due to incompatible taxonomies and weak methodologies. This paper addresses that gap by presenting the first comprehensive meta-survey of the cyber-deception survey literature, systematically analyzing how the field has surveyed, organized, and synthesized itself. We employ a rigorous methodology based on the PRISMA framework [172] to identify 258 candidate papers, screen them, and conduct an in-depth analysis of 46 survey papers spanning general cyber-deception, MTD-specific topics, honeypot research, and related areas. Our analysis reveals the field's evolution, assesses taxonomic approaches, identifies coverage gaps, evaluates methodological rigor, and synthesizes persistent research challenges. Our contributions are the following:

- **Systematic Taxonomy Analysis:** We provide the first comprehensive comparative analysis of cyber-deception taxonomies, examining classification frameworks proposed for general cyber-deception, MTD, and honeypots. This analysis identifies common themes, fundamental divergences, and opportunities for integration (Sec. A.4).
- **Comprehensive Meta-Analysis of Survey Literature:** We conduct an in-depth meta-survey of general cyber-deception (Sec. A.5.1), MTD (Sec. A.5.2), and honeypot surveys (Sec. A.5.3), analyzing research themes, temporal trends, methodological rigor, and key challenges in each area. We identify significant gaps where important techniques such as tarpits, honey encryption, and honey patches receive little or no survey coverage, and critically assess survey methodology (literature search strategies, inclusion criteria, quality assessment procedures), exposing widespread weaknesses that limit reliability and reproducibility. Finally, we synthesize open challenges and future research directions, highlighting consensus on standardized evaluation frameworks, comprehensive metrics, AI integration, hybrid system development, real-world validation, and human factors research (Sec. A.5).
- **New Unified Taxonomy:** Building on our comprehensive analysis of existing taxonomies and identified gaps, we propose a new, unified taxonomy for cyber-deception that integrates disparate models from the literature into a single, comprehensive

framework addressing the limitations of prior classification schemes (Sec. A.6). We also provide an interactive online version of this taxonomy to help researchers and deception operators explore technique options and make better design choices[1].

Taken together, these contributions yield a unified taxonomy that is not merely a relabeling of existing schemes but novel in three ways. First, it consolidates taxonomic schemes from general cyber-deception, MTD, and honeypot research into a single cross-domain framework. Second, it elevates psychological and cognitively oriented deception, together with underexplored techniques such as bias exploitation, tarpits, honey encryption, honey patches, and honeytokens, to first-class elements of the design space. Third, it explicitly aligns deception goals and deployment locations with the MITRE Engage [173] and MITRE ATT&CK [174] frameworks, enabling integration into operational workflows and security tooling.

A.2 Methodology

Generally, literature reviews come in two forms: *traditional* and *systematic*. Traditional methods are mostly subjective, relying on the author's expertise to select the most representative work in the field to present a topic. These methods are useful to describe particular issues and concepts from a narrow perspective, but are difficult to reproduce and often suffer from bias and systematic errors [175]. Systematic literature reviews provide formal and structured approaches to select, review, and extract concepts from the literature, reducing the risk of bias and increasing the reproducibility of the study [176]. Nowadays, there are many established methods to conduct systematic literature reviews for other areas of science, such as PRISMA [175], [176] and SALSA [177].

From the available frameworks, PRISMA is one of the most widely used and rigorously validated reporting standards for evidence-synthesis studies: It provides a transparent, reproducible workflow for identification, screening, eligibility, and inclusion, and has been successfully adapted beyond healthcare into computer science, and software engineering [172], [178], [179]. Therefore, we adopted PRISMA as the methodological basis for our review.

A.2.1 Eligibility criteria

We used Scopus [180] as our primary bibliographic indexer to identify candidate surveys on cyber-deception. The search targeted publication types explicitly labeled as surveys, taxonomies, reviews, or SoKs, and accepted linguistic variants of core deception-related terms. Matching terms included `decoy`, `canary`, `honey*`, `deception`, `deceptive`, `MTD`, `breadcrumbs`, and `tarpit`, together with modifiers and expanded forms (e.g., `honeypots`, `honey tokens`, `moving target {defense OR defence}`). Only studies published before 2025 were considered. Applying this query yielded 258 records (Identification). The exact Scopus query is provided in the Appendix (Listing 1).

Two reviewers ($R1$ and $R2$) independently screened all records based on titles, abstracts, and, when necessary, a brief inspection of the full content. Reviewers applied a lightweight quality-assessment protocol informed by PRISMA, the Critical Appraisal Skills Programme (CASP) [181]—a set of widely used checklists for assessing the rigor, credibility, and relevance of empirical studies—and established methodological guidance [178], [179], [182], [183]. The full set of appraisal questions is listed in the Appendix (Table A.8). Papers mislabeled as surveys/taxonomies/SoKs or lacking a section on defensive deception were excluded. In addition, reviewers removed 13 papers for language reasons (12 written in

[1] Our interactive taxonomy is available at https://deception.compute.dtu.dk/.

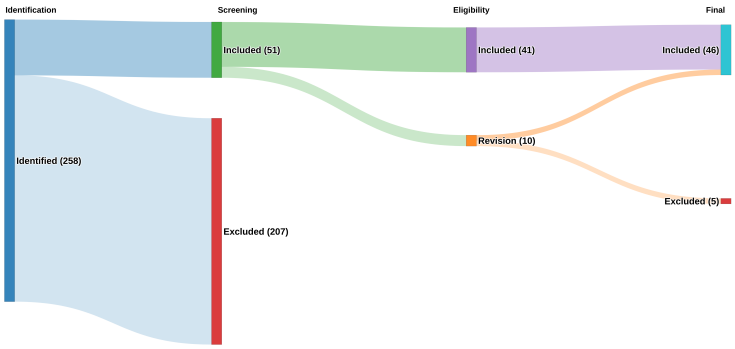


Figure A.1: PRISMA Flow Diagram for study Identification, Screening, Eligibility, and Inclusion.

Chinese and 1 in Turkish). After this initial screening, 51 papers remained. From these, $R1$ and $R2$ agreed on 41 papers. The remaining 10 conflicting items (3 exclusive to $R1$, 7 exclusive to $R2$) were forwarded to a third reviewer ($R3$). Reviewer $R3$ resolved the classification conflicts by excluding five papers from the list, yielding 46 studies for the final qualitative synthesis. These were grouped into three conceptual categories commonly represented across the literature: cyber-deception, honeypots, and Moving Target Defense (MTD). No additional searches (e.g., backward/forward snowballing) were performed beyond this stage. A summary of the identification, screening, eligibility, and inclusion steps is shown in Figure A.1. Restricting the search to Scopus and English-language surveys offers a reproducible baseline and captures the dominant body of work, but may omit some regional or venue-specific surveys; we therefore interpret our findings as characterizing the main trends in the indexed literature rather than claiming exhaustive global coverage.

A.2.2 Review process

The review team consisted of six reviewers. The remaining papers were then shuffled and allocated using pairwise permutations of reviewers, guided by maximum Euclidean distance to evenly distribute the workload. This strategy helped mitigate potential biases, as rigid reviewer groups may otherwise form opinionated islands, whereas shuffling promotes diverse perspectives through varied reviewer pairings.

Each reviewer independently examined their assigned studies to determine the paper type, topic, key contributions, and the research, analysis, and evaluation methods employed. All surveys were additionally annotated using a shared code book, designed to capture common characteristics, support the development of a unified deception taxonomy, and ensure consistent assessment across reviewers. The code book evolved during the process as reviewers encountered new characteristics and was subsequently harmonized before merging the individual assessments. The final consolidated version is presented in Table A.1.

Although our own review procedure incorporates structured assessment, multi-reviewer triangulation, and a code book, such a methodological approach is mostly absent in prior cyber-deception surveys. As detailed in Table A.2, existing surveys rarely document their screening strategies or quality-assessment processes, and when methodology is provided, it often relies on simplistic keyword searches in single indexers [184] or opaque, non-reproducible reviewer decisions [185]. These weaknesses contribute to inconsistent coverage and, as noted by Han [186], can result in biased or incomplete representations of the available evidence. This broader context motivated the design of our multi-reviewer

Group	Label	Code	Description
Taxonomy	Taxonomy present	TY1	Study includes or proposes a taxonomy or classification
Topic	Cyber-deception	TP1	Focuses on cyber operations, traps, or deceptive cyber systems
	Honeypot	TP2	Focuses on honeypot mechanisms
	MTD	TP3	Concerns Moving Target Defense mechanisms
Research method	Quantitative	RM1	Empirical, measurement-based, or statistical analysis
	Qualitative	RM2	Conceptual analysis, narrative synthesis, or descriptive study
Review method	Traditional	RE1	Narrative review without systematic protocol
	Systematic	RE2	Review following explicit methodology
Evaluation	Evaluation present	EV1	Study includes evaluation or comparison

Table A.1: Codebook used to annotate and classify defensive-deception surveys.

workflow and unified coding protocol.

After individual assessments were completed, reviewers jointly merged their evaluations using the annotations and the standardized code book. Table A.2 provides the complete list of the reviewed surveys, together with their final classification and identified characteristics. In addition to merging reviewers' assessments, we included the publication year of each study to observe how methodological practices have evolved over time. As shown in Table A.2, the chronological distribution reveals that, despite an increasing number of surveys published in recent years, there is no corresponding growth in methodological rigor: Only a small fraction employ systematic review procedures, and even fewer incorporate any form of empirical evaluation. Several of the most recent surveys (2021–2024) continue to rely exclusively on narrative or descriptive methods, with little evidence of quantitative analysis, structured appraisal, or reproducible methodology. This absence of methodological progression underscores the need for more rigorous and empirically grounded survey practices in the cyber-deception domain.

A.3 Background: Concepts and Techniques

Cyber-deception is a proactive defense paradigm that complements traditional reactive, perimeter-based mechanisms such as firewalls and anti-virus software [159], [160]. Rather than focusing solely on blocking or detecting attacks, cyber-deception deliberately introduces ambiguous, false, or misleading information and resources into a system to manipulate an attacker's perceptions and actions [161], [164]. By exploiting the inherent information asymmetry in cybersecurity, where defenders know their own systems while attackers must infer them through reconnaissance [163], cyber-deception seeks to mislead, confuse, and entrap adversaries who have already breached the perimeter [159], [161], [162]. Intentionally providing deceptive artifacts enables defenders to increase attacker uncertainty, guide adversaries toward predetermined paths, and gain critical intelligence

		Topic			Research Method		Review Method			
Study	**Tax.**	**CD**	**HP**	**MTD**	**Qual.**	**Quant.**	**Trad.**	**Syst.**	**Eval**	**Year**
Zhang et al. [160]	✓		✓		✓			✓		2003
McQueen and Boyer [187]	✓	✓			✓		✓			2009
Al-Gharabally et al. [188]			✓		✓		✓			2009
Ge et al. [189]				✓	✓			✓		2014
Heckman and Stech [190]	✓	✓			✓		✓		✓	2015
Fang et al. [191]	✓			✓	✓		✓			2015
Farris and Cybenko [192]				✓	✓		✓			2015
Briskin et al. [193]	✓	✓			✓					2016
Cai et al. [194]	✓			✓		✓	✓		✓	2016
Hassan and Guha [195]		✓			✓			✓		2017
Urias et al. [196]	✓	✓			✓		✓		✓	2017
Faveri et al. [197]		✓				✓		✓	✓	2017
Khosravi-Farmad et al. [163]	✓			✓	✓		✓			2018
Lei et al. [198]	✓			✓	✓		✓			2018
Zheng and Namin [159]				✓	✓		✓			2019
Han et al. [186]		✓			✓			✓	✓	2019
Chen et al. [199]				✓		✓	✓			2019
Efendi et al. [185]		✓				✓	✓			2019
Pawlick et al. [200]	✓	✓				✓				2020
Sengupta et al. [201]	✓			✓		✓	✓		✓	2020
Gonzalez et al. [202]		✓			✓		✓		✓	2020
Torquato and Vieira [184]				✓	✓		✓		✓	2020
Cho et al. [161]	✓			✓	✓		✓			2020
Navas et al. [203]				✓		✓		✓	✓	2021
Zhang et al. [204]	✓			✓	✓		✓			2021
Zhang and Thing [205]		✓			✓			✓		2021
Zhu et al. [206]	✓	✓			✓		✓			2021
Jalowski et al. [207]	✓			✓	✓			✓		2022
Zheng et al. [208]	✓			✓	✓					2022
Mohan et al. [209]	✓	✓			✓		✓			2022
Nguyen and Debroy [210]	✓			✓		✓	✓			2022
Saputro [162]				✓	✓		✓			2023
Amro et al. [211]	✓			✓	✓		✓			2023
Sun et al. [212]	✓			✓	✓		✓			2023
Tan et al. [213]				✓		✓	✓			2023
Deshpande and Gupta [214]		✓			✓			✓	✓	2023
Qin et al. [215]		✓				✓		✓	✓	2023
Antunes and Sanchez [216]		✓			✓		✓			2023
Yi et al. [217]		✓			✓		✓			2023
Silaen et al. [218]			✓		✓			✓		2023
Javadpour et al. [164]	✓		✓		✓		✓		✓	2024
Chouhan and Aujla [219]	✓	✓				✓	✓			2024
Lee et al. [220]				✓	✓		✓			2024
Commey et al. [221]	✓	✓			✓		✓		✓	2024
Abdi et al. [222]				✓	✓		✓			2024
Haighton and Leblanc [223]		✓			✓		✓			2024

Table A.2: Studies sorted by publication year (ascending). A checkmark (✓) indicates that the corresponding feature is present. Topic columns: CD = Cyber-Deception, HP = Honeypot, MTD = Moving Target Defense. Review Method: Traditional, Systematic.

about attack methodologies.

A.3.1 Goals of Cyber-Deception

Cyber-deception techniques serve multiple strategic objectives that collectively strengthen an organization's security posture.

Improving Detection of Malicious Activities

One of the most immediate benefits of cyber-deception is enhanced threat detection. Traditional security systems often struggle with distinguishing malicious activity from legitimate user behavior, leading to alert fatigue among security teams. Deception-based detection mechanisms address this challenge by creating resources that have no legitimate use case: Any interaction with these resources is, by design, suspicious or malicious [160], [218].

This approach provides high-fidelity alerts that warrant immediate investigation. Unlike traditional anomaly detection systems that must establish baselines of normal behavior and risk flagging legitimate but unusual activities, deception-based detection offers binary clarity: Authorized users have no reason to access deceptive resources, so any access represents a security event. This dramatically improves the signal-to-noise ratio in security operations centers, allowing analysts to focus their limited time and attention on genuine threats rather than sorting through thousands of potential false positives [164], [165].

Intelligence Gathering on Attacker Capabilities and Intentions

Beyond detection, cyber-deception provides a unique opportunity to observe attacker behavior in a controlled environment, gathering high-value threat intelligence that informs defensive strategies [160], [188]. When attackers interact with deceptive resources, defenders can monitor their tactics, techniques, and procedures (TTPs) in real-time without risking actual assets or alerting the adversary to their observation.

This intelligence serves multiple purposes. First, it reveals the skill levels and sophistication of the attacker, helping organizations understand the level of risk. Second, it exposes the specific tools, exploits, and methodologies employed by adversaries, enabling defenders to patch vulnerabilities, update detection signatures, and strengthen defenses against observed attack patterns. Third, it provides insights into attacker objectives and motivations by revealing which systems they target, what data they seek, and how persistent they are in pursuit of their goals. This strategic intelligence is invaluable for threat modeling, risk assessment, and resource allocation decisions [164], [218].

Wasting Attacker Time and Resources

Cyber-deception imposes costs on adversaries by consuming their limited time, computational resources, and attention. In the economic model of cybersecurity, defenders seek to increase the cost of successful attacks to the point where they become economically nonviable or operationally impractical [161], [224]. Deception achieves this by introducing friction into the attack process.

When attackers encounter deceptive resources, they must invest effort in exploring them, attempting exploitation, and ultimately determining whether they represent genuine targets or decoys. This time spent on false leads is time not spent advancing toward real objectives. For automated attacks and botnets that rely on speed and scale, even small delays per target can significantly reduce overall campaign effectiveness.

Increasing Attacker Uncertainty and Operational Risk

Perhaps the most strategic benefit of cyber-deception is its ability to fundamentally undermine attacker confidence and decision-making. When adversaries cannot reliably distinguish between genuine and deceptive resources, their reconnaissance data becomes

suspect, their situational awareness degrades, and every action carries the risk of revealing their presence to defenders [159], [162].

This uncertainty compounds throughout the attack lifecycle. Attackers must question whether discovered vulnerabilities are real or honeypots, whether stolen credentials will provide access or trigger alerts, whether exfiltrated data is genuine or fabricated, and whether observed system configurations reflect reality or deliberate misdirection. This cognitive burden slows attack progression, increases the likelihood of operational mistakes, and forces adversaries to adopt more cautious (and therefore likely more time-consuming and detectable) approaches [198], [213].

A.3.2 Cyber-Deception Techniques

Different cyber-deception techniques excel at achieving specific objectives, though many provide multiple benefits simultaneously.

Honeypots and Honeynets

Honeypots are functional, but decoy systems intentionally configured with apparent vulnerabilities to attract and engage attackers [166]. They range from low-interaction honeypots that simulate limited services (such as SSH or HTTP servers responding to basic commands) to high-interaction honeypots that provide complete operating systems and applications for attackers to explore. Honeynets extend this concept by creating entire networks of interconnected honeypot systems that simulate realistic organizational infrastructure [167].

The primary value of honeypots lies in their intelligence-gathering capabilities. By providing a safe, sandboxed environment where attackers can operate freely while being comprehensively monitored, defenders gain unparalleled visibility into adversary behavior. Every command executed, file accessed, tool uploaded, and lateral movement attempt is logged for analysis. This produces rich datasets about attacker TTPs that inform threat models, guide security investments, and enable proactive defense improvements. The effectiveness of honeypots depends critically on their realism [225], [226] and proper isolation. They must be convincing enough that attackers invest significant effort in exploitation, yet sufficiently isolated that compromised honeypots cannot be used as pivots to attack genuine systems.

Honey Tokens

Honey tokens are pieces of false but believable information strategically placed within systems to act as high-fidelity tripwires, with the primary goal of improving detection [165]. These can include fake database entries, API keys, internal URLs, or user credentials. The key characteristic of honey tokens is that they are monitored and the interaction with them triggers an alarm. Another important aspect is that they are designed to be attractive to attackers who have gained some level of access but are indistinguishable from legitimate data through automated scanning or passive observation.

Honey tokens are particularly effective for detection because they require no complex behavioral analysis: Any access or use of a honey token is definitively malicious, providing immediate, actionable alerts with virtually zero false positives.

Tarpits

Tarpits are services deliberately designed to respond extremely slowly to connection attempts, effectively trapping automated tools and consuming attacker computational resources [143]. Network tarpits operate at various protocol layers. An SSH tarpit might accept connections but then delay authentication handshakes indefinitely, tying up scanning tools. Web server tarpits might serve content one byte at a time or introduce long delays between HTTP responses. In each case, the objective is to make attacking sys-

tems appear available while actually imposing severe time and resource penalties on attackers who attempt interaction.

Tarpits are effective against automated attacks that rely on speed and parallelism. A vulnerability scanner expecting rapid responses may allocate threads or processes to each target; tarpits can exhaust these resource pools by refusing to complete connections quickly.

Moving Target Defense (MTD)
Moving Target Defense represents a fundamentally different approach to cyber-deception that focuses on making the attack surface unpredictable rather than presenting false targets [227]. MTD techniques dynamically and continuously alter system properties such that attackers' reconnaissance data rapidly becomes obsolete, forcing them to repeatedly re-scan and re-target systems while increasing their risk of detection. MTD operates across multiple dimensions. Network-layer MTD techniques randomize IP addresses, shuffle port numbers, or dynamically remap network topologies, ensuring that network scans quickly lose validity. Host-layer MTD approaches rotate operating system configurations, randomize memory layouts, or migrate virtual machines between physical hosts. Application-layer MTD techniques diversify software implementations, randomize execution paths, or mutate code structures to prevent exploit reuse. Data-layer MTD methods encrypt or transform data formats to prevent unauthorized interpretation. The temporal dimension of MTD is equally important. Proactive MTD changes configurations on predetermined schedules, ensuring regular invalidation of attacker reconnaissance regardless of detected threats. Reactive MTD responds to detected anomalies or attacks by immediately changing configurations to disrupt ongoing operations and force attackers to restart their campaigns.

MTD's effectiveness stems from its impact on the attack economics and the cyber kill chain [228]. Attackers typically invest significant effort in reconnaissance to map networks, identify vulnerabilities, and plan exploitation strategies. MTD undermines this investment by ensuring reconnaissance data has a limited shelf life. This forces attackers to choose between continuously re-scanning (increasing their detection risk and resource expenditure) or acting on potentially stale information (increasing their failure risk).

Honey Patches
Unlike traditional patching practices that aim to eliminate all vulnerabilities as quickly as possible, honey patching strategically manages vulnerability disclosure and remediation to create controlled deception opportunities [229]. Its main goal is to frustrate an attacker's ability to definitively determine whether their exploit attempt has actually succeeded or failed on the true system. The core deception occurs when an attempted exploit is detected. Instead of blocking the attack outright, the honey patch executes a transparent and highly efficient redirection, funneling the attacker's efforts toward an unpatched decoy system. This crucial step allows the exploit to appear successful to the attacker, maintaining the illusion that the main system has been compromised while keeping it perfectly safe.

Once the attacker is successfully engaged with the decoy, it begins to serve as a high-value intelligence asset. This decoy is equipped with aggressive software monitors that are dedicated to collecting critical information about the attacker's tools, techniques, and movements.

Honey Encryption and Data Manipulation
A more data-centric approach to deception involves honey encryption and related data manipulation techniques. These methods protect sensitive information by ensuring that attackers who successfully steal encrypted data cannot determine whether they have

obtained genuine content or decoys. Unlike traditional encryption that returns random-looking data when decrypted with incorrect keys, honey encryption produces plausible-looking but false plaintext for every possible decryption key, making it impossible for attackers to verify successful decryption [230].

This approach is particularly valuable for protecting high-value data such as password databases or intellectual property. When attackers attempt brute-force decryption, every tried key produces believable results, preventing them from knowing when, or if, they have succeeded. Frameworks like [231] align with this strategy, leveraging deception to create uncertainty and undermine the value of exfiltrated data if analyzed with automatic tools such as document clustering and topic modeling, thus protecting them even after they have been exfiltrated.

Bias Exploitation

Beyond technical deception mechanisms, sophisticated defensive strategies exploit human cognitive biases and decision-making patterns to manipulate attacker behavior [170], [171]. This psychological dimension of cyber-deception recognizes that attackers, particularly human operators, make decisions under uncertainty and are subject to the same cognitive limitations and biases that affect all human judgment. A crucial aspect of bias exploitation is that it operates on two levels: disinformation (deceptive signals) and factual information (truthful but strategically revealed signals). Disinformation involves actively projecting false information to mislead attackers about system characteristics, defensive capabilities, or organizational posture. Conversely, selectively revealing truthful information can also serve defensive deception objectives by manipulating attacker decision-making through carefully controlled transparency. This approach exploits the fact that attackers must constantly assess risk, allocate resources, and prioritize targets, decisions that can be influenced by specific revelation of genuine facts. The strategic use of factual signaling relies on information asymmetry and selective emphasis. While the information conveyed is true, defenders control which truths are visible, their context, and their prominence. These psychological concepts are operationalized by two primary mechanisms: *breadcrumbing* and *signaling*.

- **Breadcrumbing.** Breadcrumbing techniques place strategically crafted false clues, hints, or apparent vulnerabilities that lead attackers along predetermined paths toward defensive objectives [168]. Unlike passive deception mechanisms that merely present false targets, breadcrumbing actively shapes attacker decision making by exploiting natural information seeking behavior. False vulnerability indicators can be planted to draw attention toward honeypots or away from high value assets. Breadcrumbing can involve any type of data, often combining multiple deception techniques as part of a coordinated trail.
- **Signaling.** Signaling techniques involve deliberately communicating information to attackers, to influence their beliefs, decisions, and actions [169]. Unlike breadcrumbs that attackers discover through exploration, signals are actively transmitted to shape attacker perception and behavior. This can be dynamic changes during the attack or injected notifications.

A.4 Taxonomy Review

A critical step toward developing a comprehensive understanding of cyber-deception is analyzing how it has been classified and organized in the literature. Over the years, numerous taxonomies have been proposed to structure the vast landscape of deceptive techniques. These approaches vary in scope: some address cyber-deception at a high conceptual level, while others focus on specific subfields, namely MTD and Honeypots.

Table A.3: Summary of Taxonomies in General Cyber-Deception

Category	Papers	Core Characteristics	Pros	Cons
Military & Conceptual Strategy-Based	[185], [187], [190], [193], [195], [196], [206], [219]	Based on classical military doctrine. Divides deception into **Simulation** (showing the false) and **Dissimulation** (hiding the real). Focuses on high-level strategic goals.	• Conceptually grounded and intuitive. • Provides a clear, high-level starting point.	• Not native to the cyber domain. • Lacks technical granularity. • Categories can be ambiguous and overlapping.
Multi-Dimensional & Orthogonal	[186], [205]	Classifies techniques along multiple independent axes (e.g., goal, layer, deployment, kill-chain phase). Aims for a comprehensive and systematic framework.	• Provides a rich, structured, and holistic view. • Allows for systematic comparison and unification of concepts.	• Can be complex and difficult to apply in practice. • Assumed orthogonality of dimensions may not hold true. • May obscure high-level strategic intent.
Biologically-Inspired	[200], [209]	Uses a formal, tree-like structure based on biological analogies.	• Novel perspective with a formal, logical structure.	• Completeness is not formally proven. • Categories are not mutually exclusive in practice. • Relies on unique characteristics; potentially restrictive definitions.
Niche & Domain-Specific	[202], [214], [216]	Tailored to specific subfields like game theory, Generative AI, or adversarial attacks on AI models.	• High specificity and detail for the target domain.	• Not generalizable to the broader field of cyber-deception. • Sometimes uses the term “taxonomy” informally.
Honey-X Focused Enumerations	[215], [217]	Provides descriptive lists of honey-based techniques (honeypots, honeynets, honeytokens) rather than a formal classification structure.	• Simple and accessible for newcomers. • Clearly shows relationships between honey-based tools.	• Not a formal taxonomy; lacks classification dimensions. • Scope is limited to honey-based methods. • Definitions can be overlapping and incomplete.

The prominence of these two areas is reflected in a substantial body of literature that includes entire surveys dedicated exclusively to them.

To provide a structured overview, this section first examines taxonomies that address cyber-deception in general terms, then explores the more specialized frameworks developed for MTD and honeypots.

A.4.1 General Cyber-Deception

The cyber-deception literature presents multiple taxonomies, each offering distinct perspectives for classifying and understanding deceptive techniques. Analysis of existing works reveals that these taxonomies can be organized into five principal categories: (1) taxonomies derived from military doctrine and strategic theory, (2) taxonomies employing multidimensional classification frameworks, (3) taxonomies inspired by biological systems and evolutionary principles, (4) taxonomies designed for domain-specific applications, and (5) taxonomies focused on Honey-X technologies. Additionally, several works provide descriptive overviews of the field without proposing formal taxonomic structures. In the following sections, we examine each category in detail, highlighting their key characteristics, contributions, and limitations. Table A.3 provides a comparative summary of the taxonomies proposed across these categories.

Military Deception Adaptations

The most prevalent approach to classifying cyber-deception adapts classical deception theory from military doctrine. Taxonomies introduced by several works [185], [187], [190], [193], [195], [196], [206], [219] are predominantly founded on Whaley’s strategic frame-

work [232], which distinguishes between two fundamental categories: *dissimulation*, hiding the real, and *simulation*, showing the false. Dissimulation focuses on concealing genuine assets and information through techniques such as *masking* (obscuring an object's presence), *repackaging* (disguising an object's true nature), and *dazzling* (confusing an observer to prevent accurate identification). Conversely, simulation involves presenting false artifacts to mislead adversaries through techniques including *mimicking* (imitating a genuine object), *inventing* (creating novel false objects), and *decoying* (luring attackers toward false targets to divert attention from genuine assets).

Additional works within this category build upon similar strategic principles. For instance, [193] categorizes deception according to tactical objectives such as luring, misleading, or delaying attackers. Similarly, [219] proposes a framework based on the fundamental principles of creating ambiguity, distraction, and legitimacy to manipulate adversary perceptions. [190] offers an alternative but conceptually related model that frames deception within a matrix of *denial* (concealing facts) and *deception* (revealing fictions).

The principal strength of this approach lies in its foundation upon well-established deception theory, making it conceptually intuitive. The high-level distinction between *hiding* and *showing* provides a clear and accessible analytical framework. However, this clarity comes at a price. A critical limitation is that the framework originates outside the cyber domain and consequently struggles to accommodate dynamic or complex modern techniques such as MTD. Furthermore, the high level of abstraction inherent in these taxonomies often lacks the granularity required for direct technical implementation, and the proposed subtypes (e.g., repackaging versus mimicking) can exhibit ambiguity and conceptual overlap, complicating precise classification.

Multidimensional and Orthogonal Frameworks
To address the limitations of single-axis classification models, several works [186], [205] propose multidimensional taxonomies that classify deception techniques along multiple independent (orthogonal) dimensions.

Specifically, [186] introduces a formal classification system comprising four orthogonal dimensions: the *objective* of the deception (e.g., prevention, detection), the *asset* being protected (e.g., data, service), the *operational layer* (e.g., network, application), and the *deployment strategy* (e.g., built-in, isolated). In contrast, [205] employs a two-dimensional taxonomy that maps techniques against both the Cyber Kill Chain [228] phase they target and the deception layer at which they operate. This model successfully unifies diverse deception domains, including honeypots, honeytokens, and MTD, within a single coherent framework.

These taxonomies provide comprehensive and systematic classification, facilitating rigorous comparison and organization of diverse research contributions across the field. However, practical application presents challenges. The authors of [186] acknowledge difficulties in mapping specific publications to their proposed categories. The assumed orthogonality of dimensions may not hold consistently for real-world techniques, and the complexity introduced by multiple dimensions can obscure the high-level strategic intent that remains more transparent in military-based models.

Biologically Inspired Taxonomies
A third, more novel approach draws upon analogies from biological defensive mechanisms to construct a formal hierarchical classification. This category is exemplified by the framework introduced in [200] and subsequently adopted and extended in the [209] survey. This taxonomy structures deception as a tree with disjoint properties, where the first-level division is based on the treatment of private information. This distinction sepa-

rates the field into *Crypsis*, which involves concealing information and is further subdivided to classify techniques including Perturbation, Obfuscation, and MTD, and *Mimesis*, which involves creating false realities and encompasses Honey-X techniques and Attacker Engagement strategies.

The primary advantage of this model lies in its formal logical structure, which provides clear systematic hierarchy. The biological analogy offers a novel and intellectually compelling conceptual lens for understanding cyber-deception. However, the taxonomy's completeness lacks formal verification. The framework derives from biological analogy rather than from principles that would guarantee comprehensive coverage of conceivable cyber-deception techniques. Furthermore, as acknowledged by [209], the categories are not truly mutually exclusive, with certain techniques potentially fitting multiple branches.

Domain-Specific Frameworks
A fourth category of works [202], [214], [216] develops taxonomies tailored to specific subdomains of cyber-deception rather than addressing the field comprehensively. Examples include frameworks for deception in attacker-defender game theory [202], applications of Generative AI for offensive and defensive deception [214], and classification of adversarial attacks against AI models [216].

These taxonomies offer high specificity and technical depth, providing relevant and actionable classification schemes for researchers operating within particular niches. However, their applicability is inherently limited by design, precluding generalization to the broader cyber-deception landscape.

Honey-X Focused Enumerations
Finally, two works [215], [217] provide descriptive overviews of honey-based techniques rather than formal structured taxonomies. In particular, [217] enumerates various "Honey-X" technologies, including honeypots, honeynets, and honeytokens. [215] extends this enumeration to emerging concepts such as honeyfarms, honeyclouds, and honeywebs.

The primary value of these enumerations lies in their accessibility. They serve as effective introductions for researchers new to the domain, clearly delineating the fundamental components of honey-based deception and their basic interrelations. The principal limitation is that these are not formal taxonomies. They lack systematic frameworks for classification and comparison, offering instead narrative summaries without rigorous analytical structure. Their scope is deliberately limited to honey-based mechanisms, excluding other major deception strategies such as MTD or obfuscation. Furthermore, as noted in [215], the definitions of enumerated categories exhibit overlap and incompleteness, and the works fail to abstract beyond individual examples to derive generalizable principles.

A.4.2 Moving Target Defense

Unlike the broader cyber-deception landscape, MTD literature demonstrates remarkable consensus in its classification approaches. The prevailing taxonomy addresses three fundamental questions:

- **What to move?** (Spatial Dimension) refers to the system property being changed, typically organized by system layer: Network (IP addresses, ports), Platform (operating systems), Application (code), and Data (format, encoding).
- **When to move?** (Temporal Dimension) describes the trigger mechanism for change, typically divided into: *Proactive* (time-based, predetermined intervals), *Reactive* (event-based, triggered by detection), and *Hybrid* (combining both approaches).
- **How to move?** (Mechanistic Dimension) refers to the underlying implementation technique: *Shuffling* (randomizing system properties), *Diversity* (deploying multiple

variants), and *Redundancy* (maintaining multiple replicas).

While this framework provides a common conceptual vocabulary, surveyed papers apply it with varying degrees of completeness and specificity. Tab. A.4 reveals three distinct organizational approaches, ranging from single-dimensional to comprehensive multidimensional technical frameworks.

Single-Dimension Taxonomies
Several papers [159], [189], [191], [192], [198], [208], [210], [222] adopt single-dimensional taxonomies that emphasize the system layers as their primary organizing principle. These taxonomies focus exclusively on the *What* dimension, creating a hierarchical catalog of MTD techniques organized by deployment layer: Network, Platform/Host, Application, and Data. The *When* (temporal trigger) and *How* (implementation mechanism) dimensions are treated as secondary attributes describing techniques within each layer.

The principal advantage of this approach is its practical intuitiveness and accessibility. It provides defenders with a straightforward map for identifying which infrastructure components can be moved. However, this layer-centric focus risks underestimating critical temporal and operational aspects of defense, which are often equally important for effective deployment and analysis.

Two-Dimensional Taxonomies
A second group of works represents an evolution toward greater sophistication, constructing taxonomies along two of the three core dimensions. For instance, [213] builds its taxonomy on the *What* and *When* dimensions, while [163] emphasizes *What* and *How*. This approach offers more nuanced analysis than single-dimensional models by highlighting the crucial interdependencies between MTD aspects. However, these taxonomies remain inherently incomplete by design, as they omit the third dimension.

Three-Dimensional Taxonomies.
The most comprehensive approaches are presented in papers [161], [194], [201], [212], [220], which employ the complete *What–When–How* framework, treating all three dimensions as co-equal classification axes. This approach provides the most robust and systematic model for describing and comparing MTD techniques across the full design space.

Despite the framework's conceptual strengths, significant challenges persist across technical MTD taxonomies. Terminology remains inconsistent, with category definitions varying across papers.

Beyond these framework-based taxonomies, additional approaches exist with more limited scope. [162] provides a deep domain-specific taxonomy restricted to network-layer MTD techniques. Finally, [184], [211] do not provide a taxonomy of MTD techniques. Instead, they offer a broader classification scheme for organizing the entire field of study. These papers categorize the literature based on its contribution, creating buckets for topics like technology, architecture, performance metrics, testbeds, and evaluation strategies.

A.4.3 Honeypots
Among the four surveys addressing honeypots, only two [160], [164] propose formal taxonomies, each presenting a distinct classification approach.

Goal-Oriented and Application-Based Classification
The taxonomy presented in [160] classifies honeypots according to their intended function and high-level objectives within a security architecture. The framework employs a two-axis classification scheme. The first axis categorizes honeypots by their *Security Goals*: Prevention honeypots, which divert and deceive attackers away from genuine assets;

Table A.4: Summary of MTD Taxonomy Groups

Category	Papers	Core Characteristics	Pros	Cons
Single-Dimension	[159], [189], [191], [192], [198], [208], [210], [222]	Primarily focuses on one dimension, overwhelmingly the "What" (system layer), or limits scope to one domain (e.g., networking).	• Intuitive, practical, and aligns well with system architecture. • Allows for deep analysis in a specific domain.	• Can oversimplify techniques by downplaying other dimensions. • Not comprehensive across the entire MTD space.
Two-Dimensions	[163], [213]	Creates a matrix by combining two of the three core dimensions (e.g., "What & When" or "What & How").	• Offers a more sophisticated analysis than single-dimension models, highlighting key relationships.	• Inherently incomplete by omitting the third critical dimension, which can limit its explanatory power.
Three-Dimensions	[161], [194], [201], [212], [220]	Utilizes the full, co-equal "What, When, How" framework to create a complete classification space.	• The most comprehensive, robust, and systematic model for describing and comparing any MTD technique.	• Can suffer from ambiguous definitions and overlapping categories, making clean classification challenging.

Detection honeypots, which identify intrusions through monitoring system activity; Reaction honeypots, which function as sandboxes for post-incident investigation and analysis; and Research honeypots, which gather intelligence on attacker methodologies and tools. The second axis classifies honeypots by their *Application Goals*, yielding specialized categories such as Anti-spam honeypots (designed to capture unsolicited emails), DDoS honeypots (which detect and redirect distributed denial-of-service traffic), and Worm honeypots (built to capture and analyze automated malware).

The primary strength of this taxonomy is its pioneering establishment of a goal-oriented framework for honeypot classification. However, the framework exhibits significant limitations. It is largely conceptual and reflects network architectures prevalent in the early 2000s, potentially rendering it outdated in contemporary contexts. Additionally, the taxonomy was not validated through systematic literature analysis, limiting confidence in its completeness and applicability.

Technique and Theory-Based Classification.
The second approach, outlined in [164], provides a more nuanced framework by establishing a clear distinction between high-level strategic principles and low-level implementable techniques. The taxonomy comprises two complementary levels: individual honeypots and honeynets. For individual honeypots, it enumerates practical techniques operators can implement, including Advanced Mimicking (creating highly believable decoy systems), Fake Cooperation (engaging with attackers in ostensibly helpful ways), and Honeytoken Bait (planting fake data or credentials that trigger alerts upon use). Concerning honeynets (networks of honeypots), they are categorized based on simulation, which involves creating attractive and believable decoys to present false assets, and dissimulation, which focuses on concealing the true nature of the network and monitoring infrastructure.

The primary advantage of this model is its explicit separation of high-level strategy from implementable techniques, facilitating both conceptual understanding and practical deployment. However, the framework exhibits notable limitations. The categories for individual honeypot techniques are not mutually exclusive: *Advanced Mimicking*, for instance, often functions as a prerequisite for other techniques rather than as an independent classification. Conversely, the high-level strategic framework for honeynets is described as too abstract and philosophical to provide direct, actionable guidance for developers and practitioners.

A.5 Meta survey of literature

Our analysis in Sec. A.4 reveals a landscape characterized by a deep but narrow focus on established techniques, namely Honeypots and in particular Moving Target Defense. In this section, first, we explore the findings of the surveys related to cyber-deception in general, MTD, and Honeypots. Then, we highlight the critical gap in literature of the other deceptive techniques we identify in our taxonomy.

A.5.1 General Cyber-Deception Surveys

As a first step, we analyze surveys that address cyber-deception broadly, encompassing multiple deception techniques and approaches rather than focusing exclusively on a single technology. These works aim to provide comprehensive overviews of the field, though they vary significantly in scope, methodology, and analytical perspective. Our analysis examines the dominant research themes explored in these surveys, identifies the critical gaps and open challenges they consistently highlight, traces the evolution of research perspectives over time, and discusses the methodological limitations that constrain their contributions to the field. This examination shows a rapidly maturing discipline that has progressed from foundational conceptualization through formal optimization to contemporary concerns about AI integration and hybrid defense strategies, yet one that remains hindered by methodological inconsistencies and systematic coverage gaps. Tab. A.5 summarizes the analyzed papers addressing cyber-deception in general, providing an overview of their scope, key contributions, and limitations.

Table A.5: Analysis of Surveys on Cyber Deception

Paper	Scope & Focus	Findings	Future Works	Key Limitations
[200]	Game Theory (GT) models.	• Formal GT-based taxonomy. • Links deception types to game models.	• Deeper research into mimesis, advanced game models, and including human psychology.	• Narrow scope (GT only). • lacks practical effectiveness metrics.
[217]	Intro to Honeypots, -tokens, -nets.	• Presents modern case studies like PhantomFS & IoT honeypots.	• Enhance decoys with ML and deploy more advanced honeypots in cloud environments.	• Narrow scope. • Shallow analysis. • No methodology.
[206]	Deception via GT & ML.	• Multi-faceted taxonomy. • GT enables strategy, ML enables execution.	• Balance realism vs. cost • Study adverse effects. • GT–ML synergy.	• No reproducible methodology. • Model-centric focus.
[185]	Deception for web applications.	• Applies military deception taxonomy to web security.	• Create testbeds to evaluate attacker reactions.	• Over-relies on a single taxonomy. • Lacks systematic evaluation.
[190]	Strategic counterdeception.	• Introduces the “deception chain.”	• Need for technical detection tools and CTI integration.	• Highly strategic. • Lacks technical depth.
[197]	Deception planning models.	• Feature-based comparison of ten planning models.	• Tool support and improved risk analysis.	• Very small sample size.
[186]	Systematic intrusion deception review.	• Four-dimensional taxonomy with evaluation analysis.	• Automated deployment and reproducibility.	• Deliberately narrow scope.
[219]	Industry-oriented overview.	• Maps deception to standards and deployment practices.	• AI-driven, scalable Honey-X solutions.	• Abstract. • Lacks implementation details.
[187]	ICS deception framework.	• Maps deception to control-system security dimensions.	• Use as a guide for domain-specific studies.	• Highly theoretical. • No validation.
[193]	Deception design methodology.	• Introduces deception narrative and plot.	• Formal deception languages and evaluation.	• Planner-centric. • Not technical.
[202]	Personalized deception framework.	• Behavior-aware defense via cognitive models.	• Extend to real-world deployments.	• Conceptual. • Lacks empirical validation.

Continued on next page

Paper	Scope & Focus	Findings	Future Work	Key Limitations
[223]	GT for camouflage & MTD.	• Critique of six papers. • Introduces privacy concepts.	• More realistic non-zero-sum models.	• Very narrow scope.
[214]	GenAI and Cyber Kill Chain.	• Frames GenAI as dual-use technology.	• AI-driven deception environments.	• High-level. • Weak threat-defense linkage.
[215]	Hybrid deception strategies.	• Identifies lack of integrated hybrid systems.	• Better metrics and realism-cost tradeoffs.	• Weak methodology.
[209]	ML/DL for deception.	• Autonomous deception increases attacker cost.	• Reduce overhead. • Optimize timing.	• Qualitative. • ML-heavy focus.
[195]	System state modeling.	• Formal state-machine lifecycle model.	• None stated.	• Abstract. • No empirical grounding.
[221]	Blockchain-IoT deception.	• Identifies deception as an underexplored area.	• Broad research agenda.	• Speculative and high-level.
[196]	Host/network deception overview.	• Links research to commercial tools.	• Hybrid solutions and better metrics.	• Shallow analysis.
[216]	Deception for adversarial AI.	• Exploits weaknesses in adversarial ML.	• Explainability and meta-learning.	• No deception-specific metrics.
[205]	30-year honeypot & MTD review.	• 2D taxonomy across kill chain phases.	• Automation and orchestration.	• Omits human factors.

Common Research Themes

Surveys addressing cyber-deception in general exhibit several dominant research themes that reflect both the field's theoretical foundations and its practical evolution toward more sophisticated defensive capabilities.

Game-Theoretic Modeling. A significant research trend involves applying formal game-theoretic methods to model deception interactions rigorously. Works such as [200], [206], [223] review how Stackelberg games and signaling games can model attacker-defender dynamics, enabling the optimization of defensive strategies through mathematical frameworks. These approaches treat cyber-deception as a strategic decision problem, allowing defenders to compute optimal deception policies under assumptions about adversary objectives and capabilities.

Adaptive Deception Through Artificial Intelligence. Multiple surveys explore the integration of AI techniques to enhance deception capabilities and automation. Studies including [206], [209] examine how machine learning and deep learning can enable autonomous deception frameworks that improve both honeypot credibility and MTD strategy selection through data-driven adaptation. More recently, [214], [216] investigate the dual-use nature of Generative AI as both a tool enabling attackers to generate novel threats and as a mechanism for defenders to design sophisticated AI-driven deceptive defenses.

Cognitive and Behavioral Models. Moving beyond purely technical or game-theoretic approaches, some surveys explore psychological dimensions of deception. Notably, [202] introduces a novel framework integrating security games with cognitive models to create personalized deceptions tailored to individual attacker behavioral patterns and decision-making processes. This perspective recognizes that real adversaries exhibit cognitive biases and bounded rationality rather than the perfect rationality typically assumed in formal models.

Domain-Specific Applications. Some surveys focus on applying cyber-deception principles to specialized operational contexts. For instance, [185] examines deception techniques specifically designed to counter application-layer attacks such as SQL injection, while [187] develops a conceptual framework for deploying cyber-deception in Industrial

Control Systems (ICS), where operational constraints and safety requirements necessitate adapted approaches. Additionally, [221] explores deception as a defense mechanism for Blockchain and Internet of Things (BIoT) ecosystems, addressing the unique challenges of decentralized and resource-constrained environments. Finally, [196], [219] bridge academic research and commercial practice by connecting theoretical concepts to industry frameworks such as MITRE Engage [173] (formerly MITRE Shield) and analyzing the commercial deception technology market, highlighting the translation of research into operational tools.

Research Gaps and Open Challenges
When analyzed collectively, surveys addressing general cyber-deception reveal a consistent set of unresolved challenges that constrain the field's theoretical maturation and practical adoption.

Insufficient Metrics and Evaluation Frameworks. Multiple surveys [186], [196], [197], [205], [206], [209], [215] identify the lack of quantifiable and standardized metrics as a critical limitation. Beyond simple detection rates, researchers call for measures that capture attacker confusion, wasted adversary effort, defender cost–benefit ratios, and operational overhead. Some works propose more nuanced evaluation approaches: [185] advocates frameworks that monitor attacker psychological reactions (e.g., suspicion, frustration, disbelief) as primary success indicators, while [186] emphasizes reproducible real-world experiments that jointly evaluate detection performance and false negatives. Similarly, [206], [215] argue for moving from purely simulation-based studies to testbeds with realistic datasets and operational conditions. As we show in the MTD and honeypot literature (Secs. A.5.2 and A.5.3), this evaluation gap recurs as a cross-cutting challenge across the entire cyber-deception survey landscape.

Enhanced Integration with AI for Automation and Realism. Recent surveys increasingly emphasize the need for more dynamic, adaptive, and believable deception scenarios that leverage AI to respond in real-time to attacker behavior [214], [221]. Static deception systems risk detection by sophisticated adversaries who can identify patterns or inconsistencies; AI-driven adaptation promises to maintain deception credibility over extended engagements. Beyond realism, surveys such as [186], [205], [219] advocate for AI and machine learning to automate the deployment, management, and monitoring of deception systems, thereby reducing operational burden on security teams. A particularly compelling research direction discussed in [216] involves exploiting the inherent vulnerabilities of AI models through deception techniques specifically designed to counter AI-powered attacks. As adversaries increasingly employ machine learning for reconnaissance, vulnerability detection, and attack automation, defenses that can mislead or corrupt adversary AI systems represent a critical frontier in the ongoing security arms race.

Integrated Hybrid Systems. Several papers identify the future of cyber-deception as necessarily involving the combination of complementary techniques into unified defensive frameworks. [215] surveys existing hybrid systems and concludes that truly integrated implementations remain underdeveloped, with most research focusing on isolated techniques rather than their synergistic combination. Looking toward operational integration, [193] calls for tighter coupling of deception techniques with existing Command and Control (C2) frameworks and Security Operations Centers (SOCs) to enhance operational relevance. Deception cannot function as a standalone capability but must integrate seamlessly with incident response workflows, threat intelligence platforms, and security

orchestration tools. At a higher architectural level, [205] highlights the need for orchestration platforms capable of coordinating multiple deception techniques across different system layers within a cohesive strategic framework. Such platforms would enable defenders to deploy layered deception strategies that simultaneously operate at network, application, and data levels, creating more robust defenses that are harder for adversaries to circumvent.

Human Factors and Psychological Dimensions. Many surveys critique the field's predominant focus on technical mechanisms while neglecting the fundamentally human nature of adversarial interactions. [200], [206] note that game-theoretic models typically assume perfectly rational attackers, overlooking the reality of human cognitive biases, emotional responses, and bounded rationality. These works advocate for integrating psychological and behavioral models to create more realistic adversary representations that account for how humans actually make decisions under uncertainty and stress. Additionally, [190] addresses the under-researched area of counter-deception, the study of how defenders can detect when adversaries are deceiving them. This represents a critical gap in understanding bidirectional deception scenarios, where both attackers and defenders may employ deceptive tactics simultaneously.

Ethical Considerations. As deception techniques become increasingly proactive and sophisticated, potentially affecting third parties or operating in legally ambiguous spaces, [190], [200], [221] emphasize the urgent need to explore ethical implications and develop clear legal frameworks governing the deployment of deceptive defenses. Questions arise about the legitimacy of active deception tactics, the potential for collateral effects on innocent users, data privacy concerns in honeypot systems, and the appropriate boundaries between defensive and offensive cyber operations. Indeed, without addressing these ethical and legal dimensions, practical adoption of advanced deception techniques may face significant institutional and regulatory barriers.

Evolution of Research Perspectives Over Time

The analysis of the survey literature, when organized chronologically, reveals a clear three-phase evolution in the research community's perspective on cyber-deception.

Phase 1: Foundational Concepts and Frameworks (2009-2016). The first phase is characterized by highly theoretical and conceptual work. Early papers focused on establishing foundational ideas and justifying deception as a viable defensive paradigm. Research from this period was primarily concerned with proposing conceptual frameworks by adapting military strategic concepts and formal logic to the cyber domain [190], [193], and suggesting, at a high level, how deception could be applied to new specialized contexts such as Industrial Control Systems [187].

Phase 2: Formalization and Optimization (2017-2021). The second phase is characterized by a concerted effort to make deception techniques,particularly MTD and honeypots, more rigorous, intelligent, and effective. This period began with attempts to create abstract models, such as state machines, to formally define the deception process and lifecycle [195]. The research perspective then quickly shifted to how it is possible to optimize deception. This optimization drive was dominated by two major parallel trends. First, a significant wave of research applied game-theoretic modeling to optimize defender strategies and model attacker rationality [200], [202], treating cyber-deception as a mathematical decision problem to solve with formal analysis and optimal solution computation. Second, a parallel trend focused on using AI and machine learning to create intelligent

and autonomous deception frameworks [206] that could adapt dynamically to adversary behavior without constant human oversight. This phase also saw the field mature sufficiently to produce the first rigorous systematic literature reviews aimed at organizing the now substantial body of research on deception planning [197] and intrusion deception techniques [186].

Phase 3: Integration and New Arms Races (2022-Present). The third and current phase demonstrates a paradigm shift beyond optimizing isolated techniques toward integration, practical deployment, and responding to emerging threats. It is marked by mature retrospective surveys that consolidate the trends of the previous phase, such as comprehensive reviews of machine learning applications [209] and game-theoretic approaches [223]. Simultaneously, the current perspective is defined by three emergent themes that characterize contemporary research directions. First is hybridization and integration, exemplified by [215], which explicitly surveys hybrid strategies combining MTD, Honey-X techniques, and machine learning, even as it acknowledges that truly integrated implementations remain in their infancy. Second is a move toward maturity and practice, exemplified by papers such as [219] that focus on the commercial deception market and integration with industry standards like MITRE Shield. This practical orientation also manifests in expansion into new operational domains like BIoT [221], where deception techniques must be adapted to real-world constraints and requirements. Finally, the most contemporary papers [214], [216] confront the AI arms race, analyzing the dual-use nature of Generative AI as both a new threat requiring AI-powered defensive decoys and as an adversary itself, where deception can be weaponized to attack and mislead adversary machine learning models.

This clear progression, from isolated concepts, through a phase of optimizing specific techniques, to the current drive for integrated, adaptive, and AI-aware systems, demonstrates the field's rapid maturation. This evolution underscores the pressing need for a comprehensive framework, as proposed in this meta-survey, to unify these disparate but now converging research threads and provide a coherent map for the next phase of development.

Critical Limitations in Survey Literature

The analyzed survey literature on cyber-deception exhibits systematic limitations that constrain its reliability, comprehensiveness, and utility for both researchers and practitioners.

Methodological Rigor Deficiencies. A substantial portion of the surveyed works lack the methodological rigor expected in systematic literature reviews. Specifically, [187], [190], [193], [195], [196], [214], [217], [219], [223] present no formal explicitly declared methodology for literature selection and analysis, rendering their findings potentially subjective and their processes non-reproducible. Without transparency about search strategies, inclusion criteria, and selection processes, readers cannot assess comprehensiveness or replicate the analysis. Even when methodologies are present, they often exhibit significant weaknesses that compromise their reliability: [191] employs only a single keyword across two databases, severely limiting coverage and raising concerns about systematic bias in paper selection. Similarly, [215] claims to conduct a "systematic literature review" yet lacks clearly defined filtering criteria and reproducible selection processes. Additionally, [197], while systematic in its approach, analyzes only ten papers, severely limiting the generalizability of its findings and raising questions about whether such a small sample can adequately represent the broader research landscape. These methodological deficiencies highlight a fundamental tension in the survey literature: Many papers

function more as expert reviews or position papers than as systematic, evidence-based syntheses.

Narrow Scope and Topical Bias. Our analysis reveals a significant and persistent publication bias within the cyber-deception survey literature. The field's-eye view provided by existing surveys is not holistic but rather heavily skewed toward a few mature techniques and enabling methodologies. This bias manifests in several interrelated ways.

First, many surveys claiming to address cyber-deception comprehensively actually focus narrowly on the two most well-established and mature technique families: honeypots and MTD. The volume of research on these topics is so substantial that it has spawned dedicated survey categories (which we analyze separately in the following subsections). While this specialization provides valuable depth, it creates an implicit hierarchy where well-studied techniques receive disproportionate attention in surveys, while emerging or less-formalized techniques remain underexplored.

Second, a large portion of the literature focuses on enabling methodologies rather than on deception techniques themselves. This includes numerous reviews dedicated to game theory [200], [206], [223], machine learning [206], [209], [216], and high-level strategic planning [193], [197]. This methodological emphasis means that even when various deception techniques are mentioned, they often serve as illustrative examples for formal models rather than being the primary subject of analysis.

These biases create critical gaps in survey coverage that affect the field's development. Even systematic reviews with rigorous methodologies, such as [186], explicitly exclude certain technique categories to maintain a focused scope. Consequently, several important deception categories identified in our taxonomy remain almost entirely absent from survey literature or receive only superficial treatment. For example, tarpits and honey encryption are not the primary focus of any surveyed work, despite their distinct operational characteristics and use cases. Bias exploitation techniques, including breadcrumbing and strategic signaling, lack systematic review as a cohesive domain, though individual components occasionally appear in isolation (e.g., cognitive modeling in [202]). Even foundational techniques such as honey tokens lack dedicated surveys, instead being subsumed into broader umbrella categories like ”Honey-X” [215] or ”intrusion deception” [186], where they receive limited individual analysis.

This work directly addresses these gaps by situating both heavily-surveyed and underexplored techniques within a unified analytical framework, providing a more balanced and comprehensive view of the cyber-deception landscape.

Theory-Practice Gap. A recurring limitation across the survey literature is the disconnect between theoretical models and practical implementation considerations. This gap manifests in several ways that limit the actionable value of survey findings for practitioners.

Some works [187], [195], [202], [216] propose conceptually interesting frameworks that lack empirical validation or sufficient implementation details to assess real-world utility. While these theoretical contributions advance conceptual understanding, their practical applicability remains uncertain without evidence from deployment scenarios or realistic testbeds.

Other surveys [190], [193], [197] explicitly target strategic or military audiences, emphasizing high-level planning and design considerations while deliberately omitting technical implementation details. This strategic perspective is valuable for decision-makers but

leaves engineers and security practitioners without the technical guidance needed for actual deployment.

Finally, some papers [200], [206] focus extensively on model validity and theoretical properties, such as game-theoretic equilibria, while neglecting practical considerations critical for practitioner adoption. These include performance overhead, system usability, deployment complexity, operational costs, integration requirements, and scalability limitations. Without addressing these practical factors, even theoretically sound approaches can face adoption barriers in production environments where resource constraints, legacy system compatibility, and operational continuity requirements dominate decision-making.

This theory-practice gap reflects a broader challenge in security research: balancing academic rigor and theoretical advancement with the practical needs of organizations seeking to implement defensive technologies.

The Missing Metric: Data Quality and Granularity. While the literature widely critiques the absence of effectiveness and cost metrics, we uncover a more nuanced but equally critical blind spot: the lack of frameworks to quantify the granularity and quality of the collected threat intelligence. Cyber-deception systems, particularly honeypots, serve a dual purpose: They are not only defensive barriers but also high-fidelity sensors intended to extract Attacker TTPs (Tactics, Techniques, and Procedures). Research and surveys regarding advanced automated analysis of honeypot logs and Cyber Threat Intelligence extraction are limited. Similarly, existing evaluation frameworks almost exclusively focus on binary outcomes (e.g., detection success vs. failure) or system impact (e.g., latency overhead), neglecting the informational value of the engagement.

There is currently no standardized metric to differentiate between the quality of data captured by a low-interaction simulation versus a high-interaction environment. Without metrics to measure this data granularity, it is impossible for practitioners to objectively calculate the Return on Investment (ROI) of deploying complex, resource-intensive high-interaction systems over simpler alternatives. Future evaluation frameworks must therefore move beyond simple success rates to include a metric that quantifies the depth, context, and actionability of the forensic data collected.

A.5.2 Moving Target Defense

Moving Target Defense has emerged as a prominent proactive security paradigm, attracting substantial research attention across diverse application domains. In this section, we analyze the existing MTD survey landscape, examining its primary contributions, the consensus on open challenges, and the methodological limitations. Tab. A.6 summarizes these results.

Table A.6: Analysis of MTD-Specific Survey Papers

Paper	Scope & Focus	Findings	Future Work	Key Limitations
[162]	Brief review of network identity-based MTD.	• New 7-attribute classification for network identity MTD.	• None mentioned.	• No methodology. • Very narrow scope. • Descriptive, not analytical.
[211]	MTD techniques for DDoS mitigation.	• New framework mapping ”what, how, when” to DDoS techniques and technologies (SDN, NFV, etc.).	• Need for hybrid MTD frameworks, AI/ML integration, and standardized evaluation in real-world scenarios.	• Purely descriptive. • The proposed framework is not evaluated or validated. • No implementation guide.

Continued on next page

Paper	Scope & Focus	Findings	Future Work	Key Limitations
[201]	Survey of MTD in Software-Defined Networking (SDN).	• New qualitative and quantitative metrics to evaluate MTDs.	• Exploring hybrid surface shifting. • Study integration with real systems.	• Weak methodology. • Proposed metrics ignore test environments and overhead cost.
[207]	Practical view of wired network MTD (explicitly excludes IoT/wireless).	• Applies ”What, How, When” & Kill Chain frameworks. • Highlights lack of practical real-world implementations.	• Better metrics, real-world testbeds, hardware-accelerated MTD, and hybrid strategies.	• Deliberately limited scope (wired only). • Not fully reproducible methodology.
[212]	Survey of MTD with ML integration.	• Analysis of MTD solutions combined with ML.	• Integration with existing protection technologies.	• No evaluation of ML-MTD techniques against traditional ones. • Consider honeypots as a kind of MTD.
[159]	Architectural perspective of MTD (OS, software, network layers).	• Provides a formal classification of MTD strategies based on their architectural layer of deployment.	• Need for mathematical models, risk/cost analysis, SDN-based MTD, and lightweight MTD for IoT.	• Weak, non-reproducible methodology (unspecified keywords, subjective filtering).
[213]	Intersection of MTD and Game Theory.	• Dual-classification of MTD (spatial/temporal) and game models.	• Balancing time/event strategies, models for unknown attacks, and scalable models for large networks.	• No explicit methodology. • Focuses on models, not practical effectiveness. • Neglects attacker adaptation.
[208]	Broad survey of MTD techniques, evaluation, and applications.	• Evaluation approaches to assess the effectiveness of MTD.	• Call for MTD techniques integration with traditional security systems.	• Talks about what to measure but doesn’t provide standardized metrics.
[163]	MTD against APT.	• Mapping of MTD techniques to different steps of the CKC.	• Evaluation of security level of MTD techniques using graphical security models.	• Consider only IP randomization. • Other MTD techniques are not evaluated.
[184]	Systematic Mapping Study of MTD in Cloud Computing.	• Maps research by area & MTD type. • Finds most work is on platform/network.	• Need for cloud-specific MTD theories, unified evaluation frameworks, and multi-layer MTD.	• Weak search string (might miss non-”cloud” papers). • Risk of manual classification bias.
[191]	MTD for Cyber-Physical Systems (CPS).	• Maps specific MTD techniques to real-world attacks against CPS.	• Implement MTD in real CPS; create hybrid and new CPS-tailored methods.	• No methodology. • Limited to CPS.
[198]	Broad, comprehensive survey of MTD.	• Organizes MTD into Strategy Formulation, Transformation Mechanisms, and Effectiveness Evaluation.	• Research optimal dynamic timing and unified quantitative criteria.	• Complete absence of a structured methodology. • Future directions are abstract.
[210]	MTD for DoS mitigation in Cloud/SDN environments.	• Categorizes MTD for DoS (IP shuffling, VM migration) and surveys testbeds/metrics used for evaluation.	• Optimize MTD timing, integrate with IDS/IPS, address gap in AI/ML integration, expand to SD-WAN.	• No explicit methodology. • Implies selection is based on author relevance.
[194]	High-level survey of MTD state of the art.	• Proposes a ”cube model” (Layer, Functionality, Area) to organize MTD. • Categorizes evaluation into 4 types.	• Improve practicality, develop hybrid MTD, better comparison metrics, and apply game theory/SDN.	• No explicit methodology. • Model is high-level. • Not a systematic review.
[199]	Book chapter on MTD for memory protection (ASLR/D-SLR).	• Evolution from static to dynamic ASLR/DSLR, using Chameleon and SAL-ADS as case studies.	• More adaptive ASLR/D-SLR, feedback control mechanisms for cost, and better binary instrumentation.	• No formal methodology (book chapter). • A very narrow, deep technical review.
[220]	Proposal for an MTD visualization engine.	• Adapts ”What, How, When” and a 4-stage workflow (Detect, Respond, Reconfigure, Restore) to its tool.	• None mentioned for the MTD field (only for their tool).	• It’s a systems paper with a supportive literature review. • No methodology.

Continued on next page

Paper	Scope & Focus	Findings	Future Work	Key Limitations
[203]	Systematic review of MTD for IoT.	• Entropy-related metrics for evaluation. • Lack of real-world deployment for many techniques.	• Call for real-world deployment. • Need to use stronger cryptographic solutions.	• Papers covered up to July 2020. • The entropy metrics are qualitative.
[192]	Quantification of MTD via an expert opinion survey.	• Provides a coarse ordering of 24 MTD techniques by cost/effectiveness.	• Calibrate survey results against empirical/testbed data.	• Respondent fatigue. • Potential expert bias. • Lack of threat models.
[222]	Systematic review on SDN.	• Exploration of MTD techniques to improve SDN security solutions.	• Call for better overhead trade-off for MTD techniques in SDN context.	• Narrow scope: focus on SDN.
[204]	Broad review of Smart Grid security; includes a section on MTD.	• Identifies MTD as a key defense for smart grids, linking cyber techniques to physical consequences.	• None mentioned for MTD.	• Weak methodology (just ”last 5 years”). • MTD is only a small part of the review. • Limited scope.
[189]	Part review, part proposal for an MTD service framework.	• New framework with an optimal user-server mapping mechanism.	• None mentioned (discussion of gaps, but not as future work).	• Framework relies on static, known values for risk/vulnerability.
[161]	Comprehensive survey of MTD as a proactive defense.	• Uses the Shuffling, Diversity, Redundancy (SDR) framework. • Explicitly distinguishes MTD from deception.	• Interplay with other defenses, need for realistic testbeds, optimal hybrid MTD, and comprehensive metrics.	• No explicit methodology. • Uneven analysis (e.g., ’what to move’ is weak).

Common Themes in Findings and Contributions
The analysis of the MTD-focused survey literature identifies three categories of contributions: the systematic organization of the field through taxonomies, the application of MTD principles to specialized domains, and the critical examination of evaluation methodologies.

Development of Taxonomies. A significant portion of the surveyed literature dedicates substantial effort to creating structured taxonomies that manage the field’s inherent complexity, aiming to bring order to the diverse landscape of MTD techniques and approaches.

Domain Specific Application. The MTD survey literature exhibits substantial fragmentation, with numerous papers providing deep-dive analyses of specific environments. Cloud computing represents a major focus area, exemplified by systematic mapping studies such as [184] analyzing MTD applications in cloud environments, and specialized surveys like [210] examining DoS mitigation specifically. This pattern repeats across Software-Defined Networking (SDN) [201], [222] and particularly in Internet of Things (IoT) and Cyber-Physical Systems (CPS) [191], [203], [204], where surveys map MTD techniques across all system layers, from network infrastructure down to controller and physical components. This domain-specific focus extends to highly specialized technical problems, including dedicated reviews of DDoS mitigation [210], [211], network identity management [162], and memory protection mechanisms (ASLR/DSLR) [199]. While this specialization demonstrates the field’s maturation in applying general principles to practical challenges, it simultaneously contributes to the siloed nature of MTD literature, making comprehensive understanding difficult for researchers outside specific subdomains.

Evaluation Methodologies. Although most surveys identify evaluation as a critical research gap, few make the analysis of evaluation practices itself a primary contribution. Several surveys provide valuable overviews of the evaluation landscape, systematically categorizing testbed approaches (simulation-based, hardware platforms, cloud testbeds) and evaluation metrics (performance overhead, security effectiveness) [194], [198], [201],

[203], [208], [210]. The most distinctive contribution in this category is [192], which departs from traditional literature surveys by implementing a novel expert survey methodology, proposing a framework to quantify and rank MTD techniques based on aggregated expert assessments of cost and effectiveness.

Findings and Open Challenges
Despite their fragmentation across domains, MTD-focused surveys demonstrate remarkable consensus in identifying open problems.

Standardized Evaluation and Cost Analysis. Nearly all MTD surveys identify the absence of standardized evaluation frameworks and rigorous cost–benefit analysis as a central gap [161], [184], [194], [198], [207], [208]. Echoing the broader evaluation challenges discussed in Sec. A.5.1, current studies rely largely on ad-hoc, simulation-based experiments that make it difficult to compare techniques or demonstrate practical value. In addition to security effectiveness, multiple works stress the need for models that capture trade-offs between security gains and operational costs, including performance overhead, economic impact, and deployment risks [159], [192], [207], [213], so that practitioners can make evidence-based adoption decisions.

Real-World Applications. The call for standardized evaluation directly connects to the second major theme: the persistent need to transition MTD from theoretical promise to practical deployment. Several works critique the over-reliance on simulation environments and advocate for real-world applications and realistic large-scale testbeds for proper validation [161], [201], [203], [207], [211]. This emphasis on practicality extends to implementation considerations. For instance, [207] identifies hardware-accelerated MTD (leveraging SmartNICs) as a critical research direction for mitigating the performance overhead that currently impedes production adoption. Similarly, [201], [222] highlight scalability challenges that must be addressed for real-world SDN deployments.

Hybridization and Integration. The literature consistently frames MTD's future as necessarily hybrid and integrated. Surveys identify the need for research into hybrid MTD strategies that combine multiple defensive techniques, such as shuffling and diversity, to create more resilient and multi-layered defenses [161], [163], [194], [201], [207], [211]. Beyond combining MTD techniques internally, researchers also emphasize understanding the interplay between MTD and complementary defensive mechanisms. For example, integrating MTD with traditional intrusion detection and prevention systems (IDS/IPS) could enable more sophisticated cue-driven adaptation [161], [210].

Intelligent and Lightweight MTD. Finally, surveys consistently call for MTD to become simultaneously smarter and more resource-efficient. A dominant theme is the integration of Artificial Intelligence and Machine Learning to enable more adaptive, autonomous, and real-time MTD strategies capable of selecting optimal defensive moves without human intervention [210], [211], [212], [222]. In parallel, given the proliferation of resource-constrained computing environments, researchers urgently advocate for developing lightweight MTD techniques suitable for IoT, CPS, and smart grid deployments [159], [191], [212], [222]. These domains present unique challenges: They require effective security but operate under severe constraints in processing power, memory, and energy consumption.

The Evolution of MTD Survey Perspectives over Time
The analysis of MTD-focused surveys shows a distinct and logical evolution in the literature.

Phase 1: Foundational Organization (2014-2020). This initial phase centered on defining and structuring the emerging MTD field. Unlike the broader deception landscape, this period achieved rapid taxonomic consensus around three fundamental questions: What to move, When to move, and How to move. Surveys from this phase built this framework with varying levels of completeness. Many foundational works offered practical, single-dimension taxonomies focused primarily on the What dimension, cataloging techniques by system layer [159], [191], [192], [198]. Concurrently, more comprehensive surveys introduced the complete three-dimensional What-When-How framework [161], [194], [201], establishing the organizational structure that would guide subsequent research.

Phase 2: Domain Specialization (2020-Present). The second and current phase shifts the research perspective toward understanding how to apply MTD to specific domains. This led to a wave of surveys focused on narrow and practical environments, creating the highly fragmented landscape we see today. We find works dedicated to Cloud Computing [184], [210], Software-Defined Networking (SDN) [163], [210], [222], IoT and CPS [203], [204], and game theory applications [213].

Recurring Limitations in the Survey Literature

The MTD survey literature itself exhibits several limitations that constrain its utility.

Lack of Rigorous Methodology. Most MTD surveys are qualitative reviews that present findings without explicit, systematic, or reproducible paper-selection procedures. Many do not specify search keywords, databases, or inclusion/exclusion criteria [159], [161], [162], [194], [198], [201], [204], [208], [210], [212], [213], and even works that attempt to define a methodology often acknowledge overly restrictive search strings or other limitations [184], [207]. This mirrors the methodological deficiencies observed in general cyber-deception surveys (Sec. A.5.1) and weakens confidence in the completeness and neutrality of their findings.

Highly Fragmented Scope. The MTD survey landscape is dominated by narrowly scoped papers. Most reviews target specific application domains such as cloud computing [184], [210], SDN [201], [222], or CPS and smart grids [191], [204], or focus on particular threat models and techniques, including DDoS mitigation [211], network identity management [162], memory protection [199], or game-theoretic approaches [213]. Even surveys with systematic methodologies, such as [203], typically analyze relatively small paper sets, reflecting the narrow boundaries of their chosen focus areas and making it difficult to obtain a holistic view of MTD from any single source.

Non-Traditional Survey Papers. The landscape is further complicated by papers that deviate from traditional survey formats. For instance, [189], [220] are primarily proposals for novel MTD systems and frameworks: their literature review components serve to motivate design choices rather than provide exhaustive, unbiased analyses of the field. Similarly, [192] represents an expert opinion survey rather than a literature review, with findings derived from aggregated practitioner perspectives rather than systematic synthesis of published research.

A.5.3 Honeypots

Unlike the MTD literature, which saw a rapid consensus-driven organization, honeypot surveys are fragmented across time and purpose, as we summarize in Tab. A.7.

Common Themes in Findings and Contributions

Honeypot survey literature primarily contributes classification schemes and, more recently, identifies integration opportunities with modern security systems.

Table A.7: Analysis of Honeypot-Focused Survey Papers

Paper	Scope & Focus	Findings	Future Work	Key Limitations
[164]	Qualitative survey on honeypot/honeynet.	• Proposes two taxonomies: one for honeypots (mimicking) and one for honeynets (Simulation/Dissimulation).	• AI/ML for adaptive honeynets. • Need for robust evaluation metrics.	• No formal review methodology reported. • Narrative synthesis of prior work. • Proposed taxonomies have overlapping categories and limited guidance on boundaries, which may hinder practical application.
[160]	Foundational, conceptual review of honeypots as an active defense system.	• Provides a high-level conceptual model (data capture/control) and a goal-based classification.	• (Historical) Need for integration (single device), virtualization, and distribution (honeynets).	• No formal review methodology. • High-level overview rather than systematic survey. • Predates virtualization, containerization, and cloud-native deployments (2003), limiting coverage of later developments.
[218]	SLR using PRISMA on honeypots for data security.	• Categorizes modern use by domain (IoT, cloud, ICS) and integration with AI, CTI, and MITRE ATT&CK.	• Maintenance/resource costs. • Risk of recognition. • AI integration. • Need for evaluation methods.	• The synthesis of the 38 papers remains relatively high-level, with limited comparative analysis of individual techniques.
[188]	Survey and assessment of wireless honeypots (pre-WPA2 era).	• Classifies wireless honeypots by interaction level & deception method.	• (Implied) Need for effectiveness metrics, performance impact analysis, and attacker behavior studies.	• Relies on secondary reports rather than primary empirical evaluation. • No formal review methodology documented. • Focuses on pre-WPA2 wireless environments (2009), which limits applicability to current WLAN deployments.

Definition of Taxonomies. Early surveys focused on establishing conceptual clarity through taxonomic frameworks. The foundational work of [160], published in 2003, established an early conceptual model that distinguished honeypots based on their data capture capabilities and control mechanisms (how much attacker freedom they permit), while classifying them by high-level goals such as research, detection, or threat intelligence gathering. Similarly, the 2009 survey [188] adapted these foundational concepts to wireless environments, categorizing wireless honeypots by interaction level and specific deception methods unique to wireless contexts, such as spawning fake access points, emulating network traffic patterns, and simulating authentication protocols.

Modernization and Integration. More modern surveys shift from defining honeypots to enhancing them. In 2024, [164] proposed detailed taxonomies for performance-improving techniques that extend honeypot capabilities beyond basic deception. These include Advanced Mimicking (techniques for creating more realistic system behaviors and responses), Fake Cooperation (methods for prolonging attacker engagement by simulating successful exploits), and Honeytoken Bait (strategic placement of attractive but fake credentials, data, or resources to lure attackers). The systematic review presented in [218] takes a different modernization approach by categorizing contemporary honeypot applications according to two dimensions: operational domain and integration context with complementary security technologies. This integration perspective is particularly significant, as [218] documents how honeypots are increasingly embedded within AI/ML systems for automated threat analysis, Cyber Threat Intelligence platforms for indicator sharing and correlation, and the MITRE ATT&CK [174] framework for attack pattern mapping and threat hunting.

Future Directions

Despite the 20-year span of the surveyed literature, there is a remarkable and persistent consensus on the field's most critical research gaps.

Lack of Standardized Metrics. The absence of standardized evaluation frameworks dominates the literature from 2003 to 2024, appearing as the primary future-work recommendation across nearly all surveyed papers. The critique in [188] regarding missing metrics for effectiveness is echoed almost identically in recent works [164], [218], which call for comprehensive measures of deception realism, intelligence value, operational cost, and attacker dwell time. This long-standing gap reflects the broader evaluation challenges discussed in Sec. A.5.1 and continues to hinder comparative studies and evidence-based honeypot deployment decisions.

Need for Intelligent Adaptation. Modern papers [164], [218] identify intelligent adaptation as the second major gap, reflecting contemporary interest in autonomous security systems. Both papers advocate for AI integration to create adaptive honeynets that can learn from attacker interactions and dynamically improve deception strategies without manual reconfiguration. Additionally, [218] identifies a set of practical, real-world challenges that complicate honeypot deployment beyond technical capabilities. These include high maintenance costs for keeping deception environments current with legitimate system updates and patches, significant resource demands of high-interaction honeypots that require full operating systems and application stacks, the specialized expertise needed to configure realistic deception scenarios, and the persistent risk of attacker recognition and evasion through fingerprinting techniques, behavioral analysis, or community knowledge sharing.

Evolution of Honeypot Surveys
The analysis of these honeypot surveys, spanning over two decades, illustrates a clear two-phase evolution.

Phase 1: Foundational Concepts (2003-2009). Early works [160], [188] focused on defining the emerging field during a period when honeypots transitioned from niche security tools to recognized research instruments. These papers established conceptual models that distinguished honeypots from intrusion detection systems and firewalls, defined basic classifications including goal-based taxonomies (research vs. production honeypots) and interaction-level distinctions (low, medium, high interaction), and began systematically applying honeypot concepts to emerging domains like wireless networks and mobile systems.

Phase 2: Modern Integration and Evaluation Crisis (2020-Present). Current works [164], [218] shift focus from definition to enhancement, operating in an environment where honeypots are established but require modernization to address sophisticated threats. This phase emphasizes improved believability through advanced mimicking techniques that defeat fingerprinting attempts [164] and integration with modern security ecosystems including AI-driven threat analysis and MITRE ATT&CK framework mapping for standardized attack pattern documentation [218]. However, this modern phase is still constrained by the evaluation gap noted above: recent papers [164], [218] repeat the same calls for standardized metrics already identified in 2009 [188], indicating that the field still lacks robust methods for demonstrating honeypot effectiveness.

Recurring Limitations
Honeypot survey literature exhibits two primary limitations that constrain its impact and reliability.

Methodology Weakness. The most significant limitation is insufficient methodological rigor in literature review processes. Three of the four surveys [160], [164], [188] are qualitative reviews lacking formal, systematic, or reproducible paper-selection methods, which prevents independent replication and makes it difficult to assess potential gaps in coverage. In contrast, [218] adopts a PRISMA-based methodology with documented search strings, multi-database queries, and explicit selection criteria, but its analysis remains largely descriptive and surface-level. This combination of informal methods and shallow synthesis mirrors the broader survey-methodology issues observed for general cyber-deception and MTD (Secs. A.5.1 and A.5.2).

Obsolescence and Flawed Contributions. Furthermore, the existing honeypot survey literature suffers from temporal and methodological limitations that reduce its current practical utility. Foundational papers from 2003 [160] and 2009 [188] remain important historically, but their taxonomies predate virtualization, containerization, and cloud-native honeypots, and their deployment recommendations assume pre-cloud infrastructure models. As a result, they provide only limited guidance for contemporary environments. More recent work such as [164] faces different challenges. Its central contribution—the taxonomy of honeypot enhancement techniques—contains overlapping and ambiguously defined categories, so that several techniques can naturally fall into multiple classes without clear assignment criteria, and the high level of abstraction requires substantial interpretation before the taxonomy can be applied in practice. Taken together, these limitations leave practitioners with little up-to-date, methodologically robust survey guidance for understanding the current state of honeypot research and making evidence-based deployment decisions.

A.6 Our Taxonomy

A.6.1 Gaps in Existing Taxonomies

Despite two decades of research producing numerous classification schemes for cyber-deception, the analysis in Sec. A.4 reveals that no existing taxonomy provides the comprehensive, unified, and operationally actionable framework necessary to guide both research and practice. The fragmentation of the survey literature has produced taxonomies that, while individually valuable, collectively fail to address four critical gaps that hinder the field's maturation.

Incomplete Technical Coverage

The most significant gap is the systematic exclusion or superficial treatment of important deception techniques. Existing surveys exhibit strong bias toward two mature technique families, honeypots and MTD, while ignoring other approaches with distinct operational characteristics and proven practical value.

Tarpits, despite their effectiveness in resource exhaustion attacks and fundamentally different operational principles from honeypots, receive no dedicated coverage in any surveyed work and appear only as brief mentions in broader discussions. Honey encryption remains almost entirely absent from systematic surveys, despite operating on principles distinct from both honeypots and MTD. Honey patches similarly lack coverage despite their unique approach of strategically managing vulnerability disclosure and remediation. Most critically, bias exploitation receive minimal systematic attention despite their growing importance in sophisticated defense strategies. When mentioned at all, these approaches appear as superficial observations. The few works that address cognitive dimensions [190], [202] treat them as narrow research questions rather than as fundamental deception categories.

Incompatible Classification Schemes

Beyond incomplete coverage, existing taxonomies employ fundamentally incompatible organizational principles that prevent integration and comparison across the literature.

The most prominent incompatibility exists between strategy-based taxonomies adapted from military doctrine [185], [187], [190], [193], [195], [196], [206], [219] and technically-oriented frameworks organized by system implementation [159], [186], [205]. Military-derived taxonomies classify deception according to strategic intent, providing intuitive high-level categories but struggling to accommodate dynamic techniques like MTD that do not map cleanly onto these conceptual divisions. Conversely, technical taxonomies organized by deployment layer excel at describing implementation details but obscure strategic objectives. A honeypot and a tarpit might both operate at the network layer, yet serve fundamentally different defensive purposes.

Absence of Operational Guidance

While existing taxonomies successfully classify techniques according to various criteria, they largely fail to provide the operational guidance necessary for practitioners making deployment decisions. This gap between taxonomic classification and practical action manifests in two critical ways. First, many taxonomies employ ad-hoc, inconsistent, or vaguely defined goal categories that provide little actionable direction. Terms like ”prevention,” ”detection,” ”monitoring,” ”distraction,” and ”confusion” appear across surveys with varying definitions and scope. Without standardized goal definitions, practitioners cannot reliably determine which techniques serve their particular security needs. Most critically, previous taxonomies exist in isolation from the operational frameworks and standards that guide real-world security practice. Defenders working within established security operations workflows, mapped to frameworks like the MITRE ATT&CK matrix [174] for threat

modeling or the Cyber Kill Chain [228] for understanding attack progression, find little guidance about how taxonomic classifications translate into operational deployment decisions within these existing frameworks.

Domain-Specific Models Without Cross-Domain Integration
The cyber-deception literature has evolved into distinct domain-specific research communities (e.g., cloud security, IoT, and SDN) each developing specialized taxonomies and vocabulary optimized for their particular context. While this specialization has enabled deep technical expertise within domains, it has simultaneously created silos that impede knowledge transfer and prevent the development of unified conceptual frameworks applicable across operational contexts. Surveys targeting specific domains [184], [187], [191], [203], [204], [210] rarely reference or build upon classification schemes developed in other domains, even when addressing fundamentally similar deception mechanisms. Each domain has reinvented taxonomic wheels rather than adapting and extending cross-domain frameworks. This siloing manifests most visibly in terminological inconsistency: The same deception technique may be described using entirely different vocabulary depending on the domain literature in which it appears. Perhaps most problematically, domain-specific silos prevent researchers from identifying generalizable principles that transcend particular operational contexts. A deception technique developed for cloud environments might prove equally applicable to IoT systems if expressed at an appropriate level of abstraction, yet domain-specific framing obscures this potential.

A.6.2 A Three-Dimensional Unified Taxonomy

To address these systematic gaps, we propose a new comprehensive taxonomy grounded in three core design principles: dimensional separation to avoid conflating distinct aspects of deception, operational pragmatism to bridge research and practice, and systematic completeness to ensure comprehensive coverage of the technique landscape. Our taxonomy employs three dimensions that separate concerns traditionally conflated in existing frameworks: the Deception Technique dimension, the Deception Goal dimension, and the Deception Placement dimension.

The Core Technique Dimension. We establish deception technique as the primary organizing dimension, creating the taxonomic backbone based on operational mechanics, how techniques actually function, rather than their strategic intent or deployment context. This choice reflects our observation that strategic goals and deployment contexts are better treated as labels or attributes that can be applied to techniques rather than as primary classification criteria. A honeypot remains fundamentally a honeypot whether deployed for intelligence gathering or threat detection, whether running in a cloud environment or on physical hardware. Its essential nature is defined by its operational mechanism, not by why it's deployed or where it runs. By making technique the core dimension, we create a stable classification scheme that remains valid across varying strategic objectives and deployment contexts.

Orthogonal Labeling Dimensions. The Goal and Placement dimensions function as orthogonal labeling systems that contextualize techniques without constraining their classification. Any technique can be labeled with multiple goals it serves and multiple placement locations where it can be deployed, without affecting its position in the core technique taxonomy.

This three-dimensional structure provides multiple advantages. First, it allows practitioners to navigate the taxonomy from multiple entry points: Security teams can start with a goal and identify candidate techniques, or start with a deployment context and discover

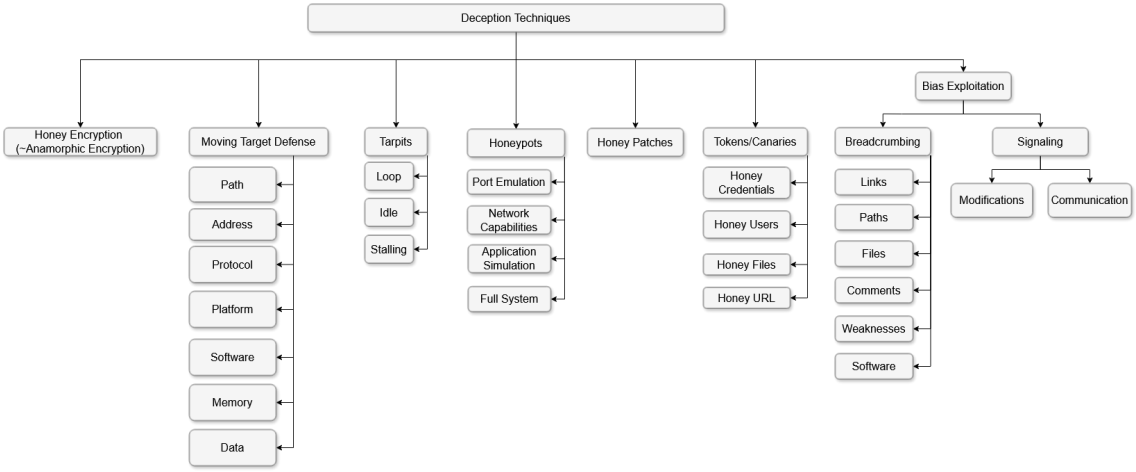


Figure A.2: Cyber Defense Deception Techniques Taxonomy.

applicable techniques, or start with a technique they are familiar with and understand its full range of applications. Second, it facilitates gap analysis: Defenders can examine which goals are addressed by their current deception deployments and which placements lack coverage, identifying gaps in their defensive posture. Third, our taxonomy supports progress toward addressing the evaluation crisis: By providing categorical structure and standardized vocabulary, our taxonomy creates the foundation necessary for community-wide comparative benchmarking efforts that can enable meaningful effectiveness comparison and scientific maturation of the field.

A.6.3 The Core Dimension: Deception Techniques

The core of our taxonomy categorizes cyber-deception into distinct technical approaches, as illustrated in Fig. A.2. We hierarchically organize techniques from major families, distinguished by fundamentally different operational principles, down to implementation-level variations. Specifically, we explicitly exclude ambiguous terms like *decoy*, which is used inconsistently in literature, and overarching terms like *Honey-X*, which conflate dissimilar mechanisms. Instead, we classify techniques based on their operational definitions.

Honeypots

We classify honeypots by implementation depth, the extent of functionality provided to attackers, rather than traditional low/medium/high interaction terminology that lacks clear definitions. At the foundational level, we define *Port Emulation* (e.g., reactive telescopes), where systems show only open ports and fail any further interaction. This progresses to *Network Capabilities*, which includes protocol emulation without application logic, followed by *Application Simulation*, where specific services are simulated. Finally, *Full System* implementations allow for the complete execution and logging of an attack within a high-fidelity environment. This classification enables practitioners to select appropriate fidelity levels based on their intelligence gathering requirements and resource constraints while maintaining clear taxonomic boundaries.

Moving Target Defense

We focus classification on what asset is being mutated rather than when to move or how to move. This reflects our observation that temporal strategies (proactive vs. reactive triggers) and implementation methods (shuffling vs. redundancy) are orthogonal attributes applicable to almost any MTD technique. By focusing on the specific asset being protected, we provide technique-centered abstraction that avoids mixed-dimension complexity. In particular, we categorize these techniques by the system layer they manipulate. The *Network Layer* covers communication surface changes, including Address (IP,

Port, MAC hopping) and Path (routing, flow table randomization). The *Platform Layer* includes OS rotation and virtualization-based migration. At the *Software/Application Layer*, techniques cover software diversification and middleware diversity. The *Runtime/System Layer* focuses on memory and process behavior, such as ASLR, ISR, and dynamic runtime environment transformation. Finally, the *Data Layer* involves data randomization, including changes to format, syntax, and representation.

Distinct Specialized Techniques
We elevate previously marginalized techniques to first-class taxonomic status, addressing the incomplete coverage gap. Rather than subsuming these under broader *Honey-X* categories or omitting them entirely, we provide a dedicated classification that acknowledges their distinct operational principles.

Tarpits. Unlike honeypots that gather intelligence through realistic engagement, tarpits are resource-consumption traps designed to waste attacker time with minimal defender investment. We classify their operation into three distinct mechanisms. First, *Loops* involve the deliberate creation of routing or protocol cycles to trap automated scanners in endless repetitions. Second, *persistence* mechanisms prevent connection termination, by ignoring close requests, remaining idle, or resending keep-alive packets, the system forces the attacker's connection to remain open indefinitely. Finally, *Stalling* exploits protocol standards by maximizing response latency at every step, technically maintaining compliance while reducing the speed of automated attacks to a crawl.

Honeytokens (Canaries). Honeytokens are passive deceptive assets, such as fake credentials, database entries, API keys, or documents, embedded directly within real production systems. They are distinct from honeypots (which are separate environments) as they serve a purely detection role: Since they have no legitimate business function, any interaction with them is by definition an anomaly that triggers a high-fidelity alert.

Honey Encryption. This technique is distinct because it targets the confidentiality medium rather than creating fake assets. Honey encryption generates plausible-looking but fake data when decrypted with an incorrect password. This denies the attacker the ability to know if they have succeeded, protecting the real data even after exfiltration.

Honey Patches. Unlike honeypots that simulate vulnerable systems, Honey Patches apply a fake fix to a real vulnerability in a production system. The patch appears to rectify the flaw to standard checks, but internally redirects any exploit attempts against that specific vulnerability to a contained environment or logger, allowing the defender to monitor the attack safely on a live system.

Bias Exploitation and Psychological Deception
A major contribution of our taxonomy is establishing Bias Exploitation as a fundamental top-level category alongside technical mechanisms. Existing taxonomies, when addressing cognitive dimensions at all, treat them as secondary attributes or marginal considerations. We argue this is fundamentally misguided. Cyber attacks involve human decision-makers (even when using automated tools), and human cognition is subject to biases and manipulability that technical deception alone cannot fully exploit. By establishing psychological deception as a first-class category with detailed subcategories, we provide a systematic organization for a previously fragmented area and signal that effective defense requires integrating both technical and psychological approaches. We divide this category into *Breadcrumbing* and *Signaling*.

Breadcrumbing. It is a passive, static approach involving the strategic placement of assets, such as files, comments, or links to create a path that manipulates the attacker's perspective. While honeytokens are strictly fake entities designed to trigger an alert upon interaction, breadcrumbs can utilize both fake and real assets, including valid files, actual software comments, or existing system configurations, to build credibility. Breadcrumbs do not need to be monitored and trigger alarms. Furthermore, while a honeytoken is often a standalone trap, breadcrumbs are typically arranged in a chain to guide the attacker away from high-value targets and towards a desired destination (such as a honeypot).

Signaling. Signaling is a reactive form of bias exploitation, where the system responds dynamically to the attacker's presence. Unlike static defenses, signaling alters the environment during an intrusion to shape the attacker's decision-making. We categorize signaling into *Communication* and *Modification*. The first approach involves direct interaction with the attacker: The system sends explicit messages, such as administrative warnings or fabricated error logs, intended to provoke an immediate reaction (e.g., causing the attacker to panic, retreat, or rush their execution). The Modification approach involves indirect manipulation requiring attacker deduction. The system alters its state in real-time, for example, changing file permissions, hiding previously visible directories, or subtly corrupting data content.

This category also anticipates future developments: As adversaries employ AI, psychological techniques may target not only human operators but also training data and decision algorithms of adversary AI systems.

A.6.4 Taxonomy Labels: Goal and Placement

While our core taxonomy distinguishes techniques based on their operational mechanics, a purely technical classification is insufficient for operational planning. To solve the problem of fragmented terminology and to bridge the gap between academic research and industrial practice, we overlay our core structure with two standardized labeling dimensions: *Deception Goal* and *Deception Placement*. We deliberately selected the MITRE frameworks [173], [174] for three critical advantages: (1) leveraging existing practitioner familiarity rather than introducing new vocabulary, (2) enabling direct integration with existing operational workflows and tooling, and (3) benefiting from ongoing framework maintenance as MITRE evolves to address new threats. Defenders already using ATT&CK matrices can directly map our taxonomy into their existing processes, while security operations centers can incorporate deception techniques using the same vocabulary they employ for threat modeling and detection.

Deception Goal. We adopt the MITRE Engage matrix [173] to resolve the ambiguity of ad-hoc goal definitions in existing taxonomies. Rather than vague terms like *distraction* or *confusion*, MITRE Engage provides precise operations with defined success criteria such as *Expose* (reveal attacker presence, capabilities, or intent), *Affect* (Degrade attacker operations or decision-making), and *Elicit* (Induce specific attacker behaviors for observation). Techniques receive multi-label goal annotations, acknowledging that they often serve multiple objectives simultaneously. For example, a honeypot might support Expose (revealing presence upon access) and Elicit (inducing behaviors through crafted vulnerabilities). This avoids problematic assumptions that single goals define primary classification, while enabling practitioners to select techniques aligned with specific defensive objectives.

Deception Placement. We utilize the MITRE ATT&CK matrix [174] to provide precise deployment vocabulary, replacing generic terms like *host-based* or *network-based* with detailed attack-stage and tactic-specific placement. MITRE ATT&CK maps the placement of defenses directly to the attack surface (e.g., Enterprise, Mobile, ICS). For example, honeytokens map directly to the Credential Access tactic. This alignment enables a clearer visualization of defensive coverage, helping practitioners identify which layers of their technology stack are protected by deceptive assets and where coverage gaps remain.

To support practical application, we have mapped these Goal and Placement labels to every specific technique within our taxonomy tree. As this detailed integration exceeds the spatial constraints of a static figure, the complete interactive mapping is available for consultation online (see Sec. A.1 and Appendix A.8).

A.7 Conclusion

This work presents the first systematic meta-survey of the cyber-deception literature, analyzing 46 survey papers to address the field's growing fragmentation and provide a synthesis of its evolution, taxonomic approaches, and persistent research challenges. Our analysis reveals deep but narrow expertise across three research families: general cyber-deception, Moving Target Defense (MTD), and honeypots. The MTD literature exhibits strong taxonomic consensus yet remains fragmented across application domains, the honeypot literature continues to face long-standing evaluation challenges, and critical techniques such as tarpits, honey encryption, honey patches, and bias exploitation strategies remain largely absent from survey coverage.

To address these gaps, we introduced a unified taxonomy that integrates disparate classification schemes from across the literature. This taxonomy is grounded in the MITRE Engage and MITRE ATT&CK frameworks and is not merely a re-labeling of existing schemes, but is novel because it consolidates the fragmented taxonomies developed for general cyber-deception, MTD, and honeypots into a single cross-domain framework; it treats psychological and cognitively oriented deception – together with underrepresented techniques such as bias exploitation, tarpits, honey encryption, honey patches, and honeytokens – as first-class elements of the design space rather than peripheral add-ons; and it explicitly anchors deception goals and deployment locations in the MITRE Engage and MITRE ATT&CK matrices, so that they map cleanly onto vocabularies already used by practitioners for threat modeling, hunting, and incident response. Taken together, this taxonomy provides a common vocabulary that can bridge research silos and systematically organize both well-studied and under-explored deception techniques within a single coherent framework.

Our most significant cross-cutting finding is the lack of standardized evaluation frameworks and quantitative metrics. Despite repeated calls in the literature, the community still lacks rigorous methods to measure deception effectiveness and cost-benefit tradeoffs, and most work is validated only in small-scale or simulated settings. This limits comparability across techniques and weakens the empirical basis for deployment decisions.

The surveyed works converge on future directions such as AI- and ML-driven adaptive deception, hybrid combinations of complementary mechanisms, and tighter integration with existing security ecosystems. Our analysis highlights a central paradox: without resolving the evaluation gap, it remains difficult to determine whether such advances provide real security gains or simply add complexity. By synthesizing the survey landscape, identifying coverage and methodological gaps, and establishing a unified taxonomic framework, this meta-survey provides a consolidated map of the cyber-deception field and a foundation

```
TITLE-ABS-KEY(
  ( cyber OR security )
  AND ( defense OR defend OR defence )
  AND ( deception OR deceptive OR honey OR mtd
        OR ( moving AND target AND defense )
        OR ( moving AND target AND defence )
        OR tarpit OR decoy OR canary OR canaries OR breadcrumb )
  AND ( survey OR taxonomy OR review OR ( state AND of AND knowledge )
        OR sok OR classification )
)
AND PUBYEAR < 2025
```

Listing 1: Scopus query for eligible candidate studies.

for more rigorous, measurable progress.

A.8 Appendix

This appendix provides supplementary material to support the transparency and reproducibility of our methodology. In particular, we document the exact bibliographic search query used to identify candidate surveys and the critical appraisal questions that guided the initial screening and assessment of their methodological quality.

A.8.1 Scopus Search Strategy

Our starting point for identifying candidate defensive-deception surveys was the Scopus bibliographic index, as described in Section A.2. The search string was designed around three main groups of terms.

- **Security context**: terms such as *cyber* and *security* to ensure that retrieved papers are grounded in cybersecurity or closely related domains.
- **Defensive-deception terminology**: linguistic variants capturing the core technique families of interest, including *deception*, *deceptive*, the stem *honey* (to cover honeypots, honeytokens, honeynets, etc.), *MTD* and the explicit phrase *moving target defense/defence*, as well as related concepts such as *tarpit*, *decoy*, *canary*, *canaries*, and *breadcrumb*.
- **Survey-type publication labels**: descriptors such as *survey*, *taxonomy*, *review*, *state of knowledge*, *SoK*, and *classification*, which explicitly indicate that the work is intended as a secondary study rather than a primary research contribution.

The query is restricted to the `TITLE-ABS-KEY` fields so that only terms appearing in the title, abstract, or author keywords are considered, which balances recall and precision and avoids relying on full-text availability. We explicitly include both American and British spellings of *defense/defence* to reduce language-induced omissions. The `PUBYEAR < 2025` predicate enforces the temporal cut-off of our study and matches the scope discussed in Section A.2.

Applying this query in Scopus yielded 258 records before screening, which were then filtered according to the eligibility criteria and language restrictions presented in the main text. The exact Scopus query that was used for our methodology (cf. Section A.2) is provided in Listing 1. We include it verbatim to enable direct reuse: future meta-surveys can, for example, extend the time window by adjusting the `PUBYEAR` constraint or augment the search with additional domain-specific keywords while preserving comparability with our study.

A.8.2 Quality Appraisal Questions

To support a consistent and transparent assessment of methodological quality, we employed a lightweight critical appraisal protocol during the initial screening of candidate surveys (cf. Section A.2). The protocol is based on the Critical Appraisal Skills Programme (CASP) checklist for systematic reviews and the PRISMA reporting guidelines, which we adapted to the context of secondary studies on defensive deception and related security topics.

The resulting questions focus on three broad aspects:

- **Basic study design**: whether the paper is structured in a way that is appropriate for a systematic review or mapping study, and whether it clearly articulates its research questions.
- **Methodological soundness**: whether the literature search, screening, selection, and data-extraction procedures are described in sufficient detail to be reproducible and to justify the completeness of the evidence base.
- **Trustworthiness of the results**: whether the synthesis, discussion, and reported limitations support a credible and useful interpretation of the evidence.

The full set of appraisal questions is listed in the table below. They are grouped into three sections that mirror these aspects: Section A addresses the basic validity of the study design; Section B focuses on the soundness and transparency of the review methodology; and Section C examines whether the reported results and limitations can be considered trustworthy and actionable.

In our study, these questions were used as a qualitative checklist rather than as a numerical scoring rubric. During initial screening, each candidate survey was evaluated against the items in Table A.8 to identify severe methodological weaknesses, such as the absence of an explicit search strategy, unclear inclusion and exclusion criteria, or a lack of discussion of limitations. These appraisals informed the exclusion of papers that were either mislabeled as surveys/taxonomies/SoKs or lacked any substantive treatment of defensive deception, and they provided structured support for the multi-reviewer conflict-resolution process described in Section A.2.

Beyond informing inclusion and exclusion decisions, the checklist also served as an interpretive lens in our qualitative synthesis of the final set of surveys. In particular, it helped us to systematically characterize differences in methodological rigor across studies and to highlight recurring weaknesses in existing survey practice, as discussed in the main text. We make the checklist explicit here so that it can be reused or adapted by future meta-surveys in cyber-deception and related security domains.

A.8.3 Interactive Cyber Deception Matrix

To make our unified taxonomy practically usable, we provide an interactive web-based matrix[2] that we maintain as a living resource. The matrix instantiates the taxonomy presented in the main text (Sec. 6) and augments each concrete technique with metadata about its goals and deployment contexts.

Each column in the matrix corresponds to a deception approach (for example, honeypots, honeytokens, tarpits, honey patches, or signaling), while the cards to the specific techniques. Each card summarizes: (i) the technique's MITRE Engage goal labels, (ii) MITRE ATT&CK tactic(s), (iii) detailed description and (iv) the related resources. The

[2] https://deception.compute.dtu.dk/matrix

ID	Critical Appraisal Questions for Initial Screening
Section A: Is the basic study design valid for a systematic review?	
A1	Did the systematic review address a clearly formulated research question?
A2	Did the researchers search for appropriate study design(s) to answer the research question?
Section B: Is the systematic review methodologically sound?	
B3	Were all the relevant primary research studies likely to have been included in the systematic review?
B3a	Searching for primary research studies.
B3b	Screening primary research studies from the search.
B3c	Selecting primary research studies to include in the systematic review.
B3d	Summarizing the search and its outputs.
B4	Did the researchers extract and present information from the individual primary research studies appropriately and transparently?
B4a	Extraction of data.
B4b	Presentation of data.
Section C: Are the results of the systematic review trustworthy?	
C5	Did the researchers analyse the pooled results of the individual primary research studies appropriately?
C5a	Subgroup analysis.
C6	Did the researchers report any limitations of the systematic review, and do these limitations cover all issues identified during critical appraisal?
C7	Would the benefits of acting upon the results outweigh any potential disadvantages, harms, or additional resource demands?

Table A.8: Questions derived from CASP checklist for systematic review [181] and PRISMA checklist [172]

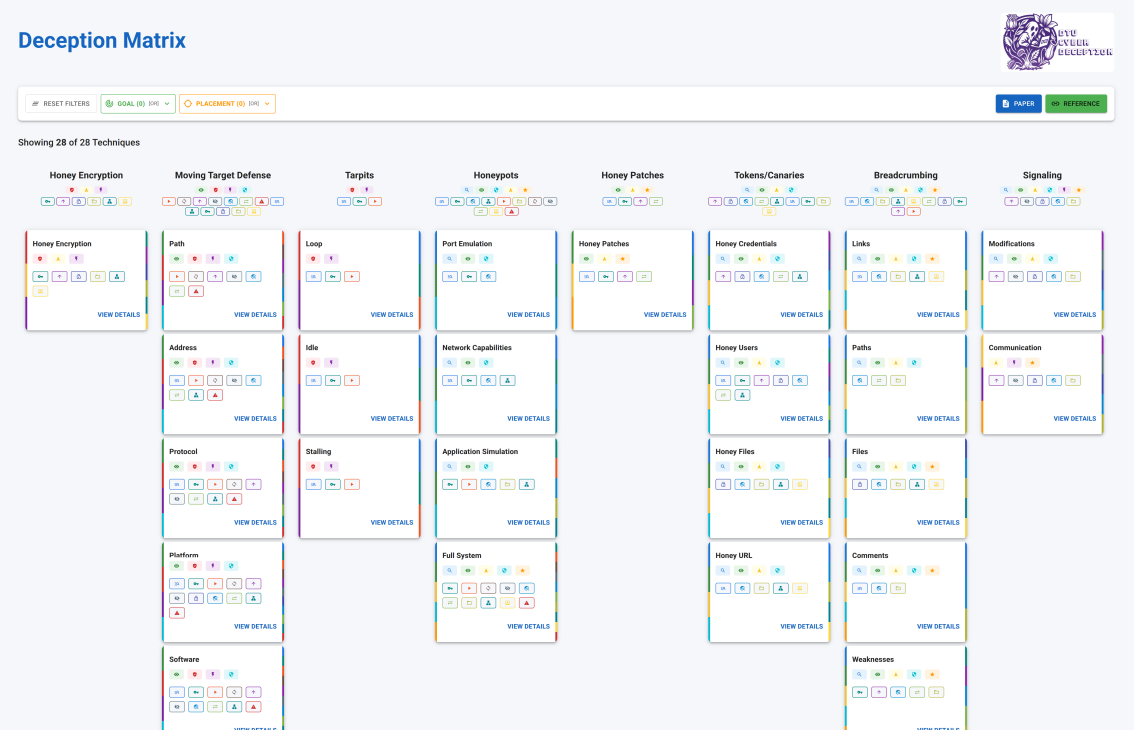


Figure A.3: Cyber deception taxonomy matrix: overview of the interactive web interface.

interface supports filtering and search across goals and placements metadata, allowing users to restrict the view to techniques that, for instance, support a particular goal (such as *Expose* or *Affect*), are mapped to a specific ATT&CK tactic, or have publicly available implementations.

We include two screenshots to give a visual impression of the resource; however, we encourage readers to explore the online matrix directly. Figure A.3 shows the default overview with multiple deception families visible, while Fig. A.4 zooms in on the Tarpits and Stalling branch to illustrate how fine-grained variants are encoded and linked to the underlying literature and tools.

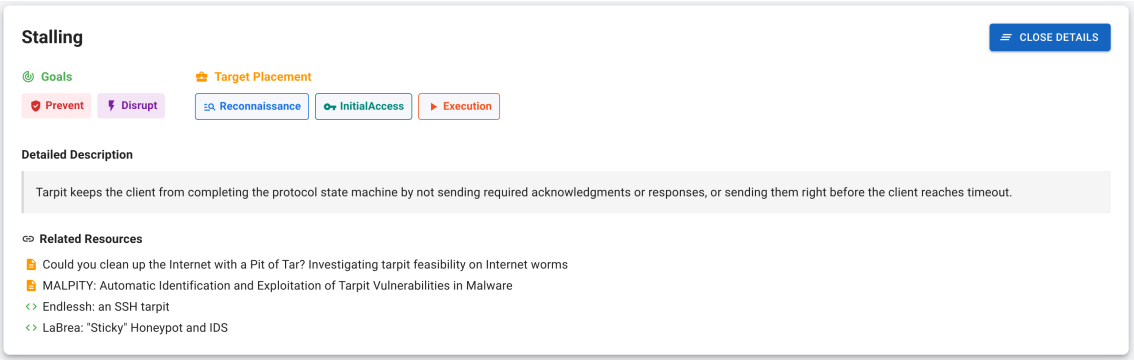


Figure A.4: Cyber deception taxonomy matrix: detailed view of tarpits and stalling techniques.

Cyber-attacks are becoming more frequent and more damaging, yet many Internet-connected devices still lack even basic security protections. From smart home systems and industrial controllers to healthcare and energy infrastructure, millions of devices are directly exposed to the Internet with weak access control, outdated software, or no maintenance at all. This PhD thesis investigates why such cyber-security weaknesses persist, how they can be identified at scale, and how their risks can be mitigated.

The thesis focuses on Internet of Things (IoT) and Operational Technology (OT) devices (i.e, systems that control physical processes such as manufacturing equipment and critical infrastructure). Using Internet-wide measurement techniques (i.e., scanning the whole Internet), the research develops and applies new methods to systematically identify exposed devices and detect signs of misconfiguration, abandonment, and obsolescence. More than ten protocol-specific probes are proposed, enabling the detection of security problems that go beyond well-known vulnerabilities and instead reflect long-term neglect and poor security management.

The thesis also presents a modular framework that supports large-scale Internet measurements and continuous monitoring: DICE, a Device Identification and Classification Engine. This framework enables reproducible research, supports responsible vulnerability disclosure, and helps translate measurement results into actionable insights for maintainers and operators.

Technical
University of
Denmark

Richard Petersens Plads, Building 322
2800 Kgs. Lyngby
Tlf. 4525 1700

https://www.compute.dtu.dk/