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

Ricardo Yaben *, Emmanouil Vasilomanolakis *
Technical University of Denmark
Kongens Lyngby, Denmark
*{rmyl,emmva}@dtu.dk

***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.**

***Index Terms*—vulnerability identification, Internet-wide scans, IoT, OT**

I. Introduction

THE emergence of the Internet of Things (IoT) and Operational Technology (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 [2], [3], [4], propose mitigation strategies to reduce the number of exposed and vulnerable devices [5], or investigate society's cyber-security posture towards their devices [6]. 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 [7] 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 [8] for granulated classification, Greynoise [9] to identify malicious hosts, the NIST vulnerability database for known vulnerabilities, and RIPEstat [10] 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 Digital Ghost Ships (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 II begins with an overview of the relevant literature for

identifying vulnerable IoT and OT devices over the Internet. In Section III, 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 IV we analyze our scanning results to identify DGSs. Lastly, Section V summarizes our findings, touching on the IP reputation of the potentially vulnerable devices we discovered, and the responses to our vulnerability disclosure. Section VI concludes this paper.

II. Related Work

Numerous studies conduct Internet-wide scans to investigate vulnerabilities in IoT and OT devices [11], [12], [13]. The methods for scanning the Internet are well-established [14] and most authors use off-the-shelf common tools such as those from the ZMap ecosystem [15] or Masscan [16], 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) [17].

A significant part of the literature focuses on Industrial Control Systems (ICSs) exposed to the Internet [18], [19], [20], given that many of those systems operate in critical environments and lack security features. In [21], 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, [6] 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, [22] 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 [23]. In [3], the authors scanned for specific IoT devices over the Internet to identify vulnerabilities and other issues associated with this technology. Moreover, [24] 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 [2] 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 [2] is in the aim of our work; [2] 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 Common Vulnerabilities and Exposures (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.

III. 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.

A. Scanning the Internet

We divide our scans into two phases following the learnings from the literature [25], [15], [17], [26], [27], 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 [14]. 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 [28].

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.

B. Ethical considerations and limitations

Conducting Internet-wide scans produces a substantial load of traffic on target networks [29], [14]. 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 [15], 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 [2], limiting our connections to 30 seconds and setting limits to the amount of data we gather (cf. Section IV 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 V-B.

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 [17] .

C. 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 [17], [15], [30]. Adding to this, we are aware of the behavior of some controllers exposing RTPS services that will not stop transmitting data for long periods [31].

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 I: Classification criteria for DGSs

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 [6], 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 I. 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 IV.

		$S1$		$S2$		
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	○

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

IV. 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 II 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 [7], 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 $S2$.

A. 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 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 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.

B. 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 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.

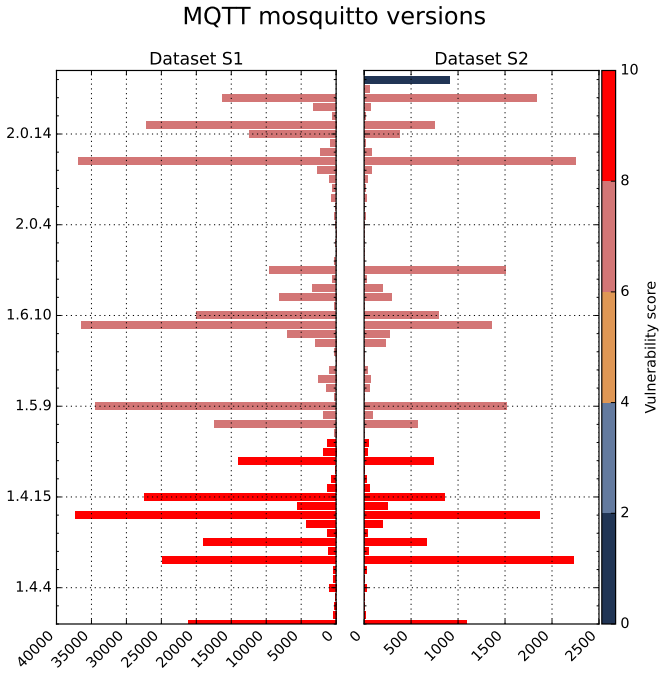


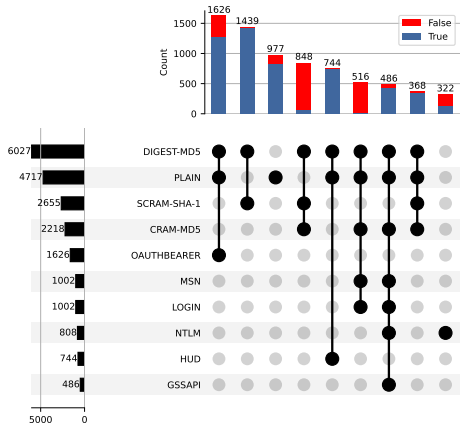
Fig. 1: Distribution of MQTT Mosquitto versions colored by the severity of their vulnerabilities, with 0-2 in dark-blue indicating no vulnerabilities found.

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.

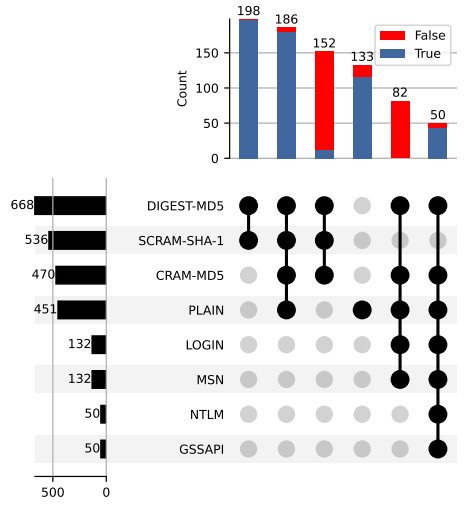
Takeaway – Allowing anonymous clients to communicate with CoAP servers has severe security and privacy implications. Clients can access sensitive information, such as implementation details and device characteristics. In addition, CoAP servers without security options enabled (NoSec mode) can be used 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.

C. XMPP

Previously known as *Jabber*, XMPP is the open Standard for messaging applications based on XML. Today, this protocol



(a) Dataset S1



(b) Dataset S2

Fig. 2: XMPP top 10 most frequent combinations of authentication mechanisms and count of observations using TLS.

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

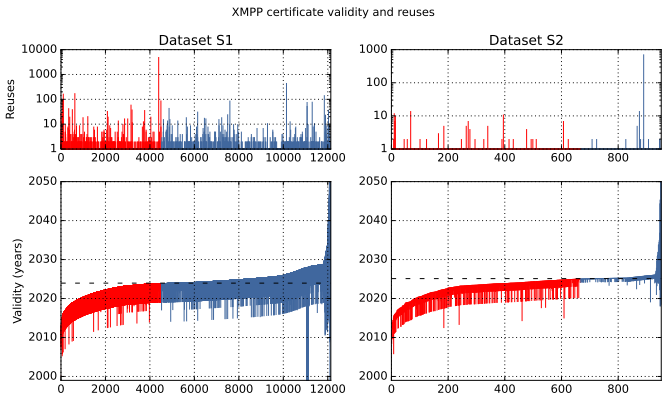


Fig. 3: Validity of XMPP certificates and reuses. On top is the count of reuses for each certificate. On the bottom, the validity ranges for each certificate. Expired certificates are represented in red.

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 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 [32].

Furthermore, we collected 14,735 unique certificates from Shodan for $S1$ and 952 for $S2$; Figure 3 shows the unique certificates and their validity ranges, with the number 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.

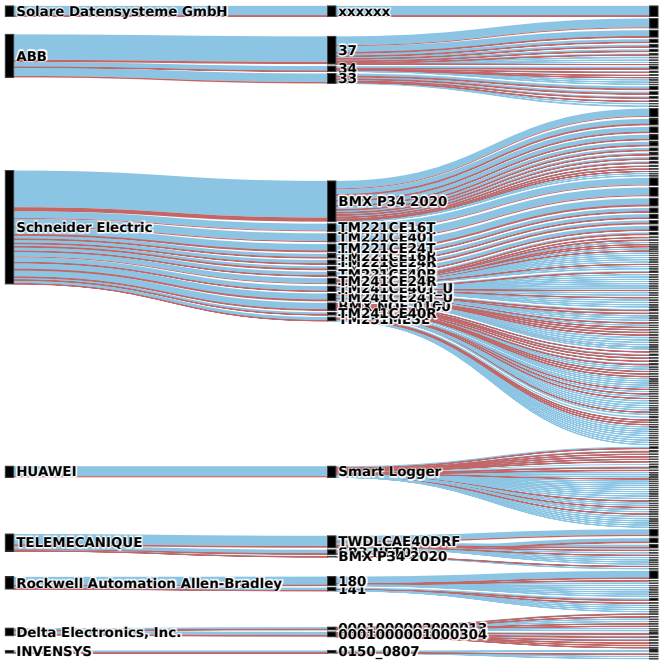


Fig. 4: Top vendor distribution of products and their firmware versions exposing Modbus services on the Internet.

D. 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 [33], [21]. Our scanner uses the default 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 [21]. This dataset contains 288 different devices from 82 vendors. Figure 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 Remote Terminal Units (RTUs) and Programable Logic Controllers (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 devices, 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.

E. 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 [18], 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 [34]. 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 [18], 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

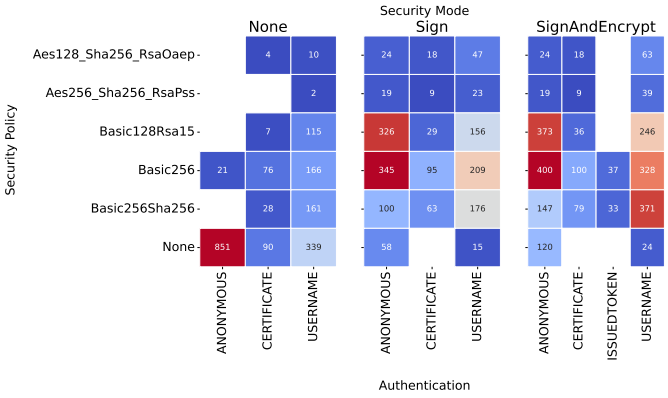


Fig. 5: $S2$ dataset OPC UA distribution of security modes, and combinations of authentication methods with encryption policies.

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 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 comparison 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 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,

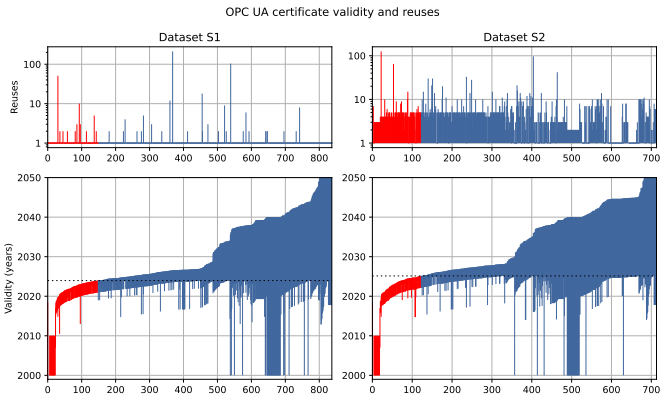


Fig. 6: Validity of OPC UA certificates and reuses. On the top, is the count of reuses for each certificate. On the bottom, the validity ranges for each certificate. Expired certificates are represented in red.

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 [18]. Furthermore, maintainers must remove support for weak authentication and encryption methods. 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. Lastly, servers with expired certificates or valid for the past 5 years are no longer considered secure or valid for cryptographic operations.

F. 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 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 [31], 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.

G. 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 [35]. 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

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

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.

H. 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.

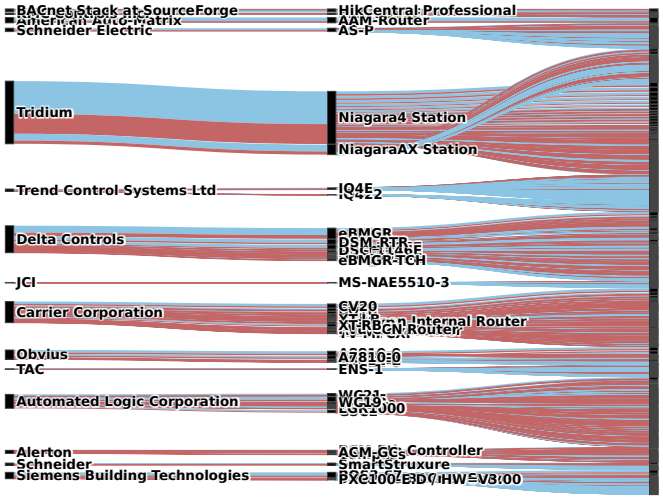


Fig. 7: Top vendor distribution of products and their firmware versions exposing BACnet services on the Internet.

Furthermore, Figure 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 [36]. Some manufacturers instruct the use of a VPN for all standard BACnet communications. The BACnet/SC addendum should be considered otherwise.

I. 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

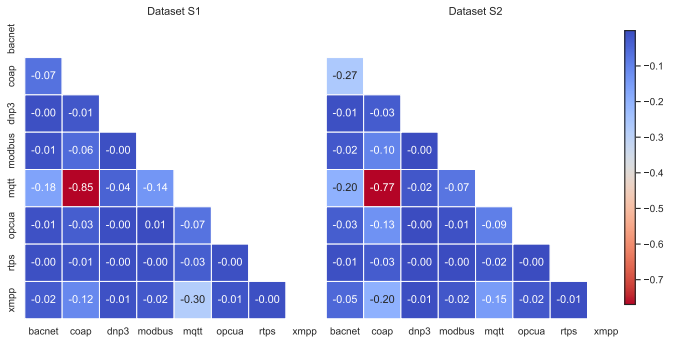


Fig. 8: Correlation of protocols exposed in hosts.

external networks perceive our probes [17], [29]. 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 $\tilde{8}0\%$, suggesting that it is highly unlikely to find hosts exposing both protocols simultaneously. Figure 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).

V. 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.

A. 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.

B. 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 [37], 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 [38], [39], [40]. Other authors raised their concerns regarding notification campaigns and the minimal impact on consumer behavior [41], [37], [42]. Generally, most notifications go unnoticed, ignored, bounce back, or receive automated responses.

C. 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 [43], [44]. However, it is their shared responsibility to maintain the security of their devices [45], [39].

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.

VI. 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.

Acknowledgment

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

References

[1] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1093–1110. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[2] 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. New York, NY, USA: Association for Computing Machinery, 2021, p. 195–215. [Online]. Available: https://doi.org/10.1145/3487552.3487833

[3] 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)*, vol. 1. IEEE, 2015, pp. 463–467.

[4] 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, p. 2702–2733, 2019.

[5] 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.

[6] 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.

[7] R. Yaben, N. Lundsgaard, J. August, and E. Vasilomanolakis, "Towards identifying neglected, obsolete, and abandoned iot and ot devices," in *2024 8th Network Traffic Measurement and Analysis Conference (TMA)*, 2024, pp. 1–10.

[8] "Shodan search engine," https://www.shodan.io/, (Accessed on 03/01/2024).

[9] "Greynoise — sensors and benign scanner activity," https://www.greynoise.io/, (Accessed on 03/01/2024).

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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. [Online]. Available: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric

[16] "robertdavidgraham/masscan: Tcp port scanner, spews syn packets asynchronously, scanning entire internet in under 5 minutes." https://github.com/robertdavidgraham/masscan, (Accessed on 03/01/2024).

[17] G. Wan, L. Izhikevich, D. Adrian, K. Yoshioka, R. Holz, C. Rossow, and Z. Durumeric, "On the origin of scanning: The impact of location on internet-wide scans," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 662–679.

[18] 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. New York, NY, USA: Association for Computing Machinery, 2020, p. 101–110. [Online]. Available: https://doi.org/10.1145/3419394.3423666

[19] 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.

[20] Y. Wu, S. Song, J. Zhuge, T. Yin, T. Li, J. Zhu, G. Guo, Y. Liu, and J. Hu, "Icscope: Detecting and measuring vulnerable ics devices exposed on the internet," *Communications in Computer and Information Science*, vol. 1851 CCIS, p. 1 – 24, 2023. [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

[21] A. Mirian, Z. Ma, D. Adrian, M. Tischer, T. Chuenchujit, T. Yardley, R. Berthier, J. Mason, Z. Durumeric, J. A. Halderman, and M. Bailey, "An internet-wide view of ics devices," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, 2016, pp. 96–103.

[22] 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. New York, NY, USA: Association for Computing Machinery, 2022, p. 252–266. [Online]. Available: https://doi.org/10.1145/3488932.3497762

[23] S. J. Saidi, A. M. Mandalari, R. Kolcun, H. Haddadi, D. J. Dubois, D. Choffnes, G. Smaragdakis, and A. Feldmann, "A haystack full of needles: Scalable detection of iot devices in the wild," in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 87–100. [Online]. Available: https://doi.org/10.1145/3419394.3423650

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

[25] 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. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/hiesgen

[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.

[27] 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. New York, NY, USA: Association for Computing Machinery, 2015, p. 542–553. [Online]. Available: https://doi.org/10.1145/2810103.2813703

[28] "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).

[29] 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. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/durumeric

[30] 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

[31] F. Maggi, R. Vosseler, M. Cheng, P. Kuo, C. Toyama, T. Yen, and E. B. V. Vilches, "A security analysis of the data distribution service (dds) protocol," *Trend Micro Research, Inc., Japan*, pp. 15–20, 2022.

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

[33] 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.

[34] R. Yaben and E. Vasilomanolakis, "Drifting away: a cyber-security study of internet-exposed opc ua servers," in *Proceedings at the 10th International Workshop on Traffic Measurements for Cybersecurity (WTMC 2025)*. United States: IEEE, 2025, 10th International Workshop on Traffic Measurements for Cybersecurity, WTMC ; Conference date: 30-06-2025 Through 30-06-2025.

[35] "Sav6 and amp flyer - 2022 - final.pdf," https://www.dnp.org/Portals/0/Public%20Documents/SAv6%20and%20AMP%20flyer%20-%202022%20-%20Final.pdf?ver=z_i7KIkCzZDyYSWJPhU3KA%3d%3d, (Accessed on 02/22/2024).

[36] 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.

[37] 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.

[38] 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. [Online]. Available: https://www.usenix.org/conference/soups2016/technical-sessions/presentation/fagan

[39] C. Herley, "More is not the answer," *IEEE Security & Privacy*, vol. 12, no. 1, pp. 14–19, 2014.

[40] 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. New York, NY, USA: Association for Computing Machinery, 2009, p. 133–144.

[41] O. Gasser, Q. Scheitle, B. Rudolph, C. Denis, N. Schricker, and G. Carle, "The amplification threat posed by publicly reachable bacnet devices," *Journal of Cyber Security and Mobility*, jan 2017. [Online]. Available: http://www.net.in.tum.de/fileadmin/bibtex/publications/papers/bacnet-jcsm.pdf

[42] F. Li, Z. Durumeric, J. Czyz, M. Karami, M. Bailey, D. McCoy, S. Savage, and V. Paxson, "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. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/li

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

[44] 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, p. 495–495, Dec 2022.

[45] 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.