Downloaded from orbit.dtu.dk on: Apr 20, 2025

DTU Library

Towards identifying neglected, obsolete, and abandoned IoT and OT devices

Yaben , Ricardo; Lundsgaard, Niels; August, Jacob; Vasilomanolakis, Emmanouil

Published in:
Proceedings of the 8th Network Traffic Measurement and Analysis Conference (TMA Conference 2024)

Link to article, DOI:
10.23919/TMA62044.2024.10558996

Publication date:
2024

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Yaben , R., Lundsgaard, N., August, J., & Vasilomanolakis, E. (2024). Towards identifying neglected, obsolete, and abandoned IoT and OT devices. In *Proceedings of the 8th Network Traffic Measurement and Analysis Conference (TMA Conference 2024)* IEEE. https://doi.org/10.23919/TMA62044.2024.10558996

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Towards identifying neglected, obsolete, and abandoned IoT and OT devices

Ricardo Yaben *, Niels Lundsgaard†, Jacob August†, Emmanouil Vasilomanolakis *
Technical University of Denmark
Kongens Lyngby, Denmark
*{rmyl,emmva}@dtu.dk
†{s220474,s220473}@student.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. 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.**

***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 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 [2], [3], [4], proposing mitigation strategies to reduce the number of exposed and vulnerable devices [5], and investigating society's cyber-security posture towards their devices [6]. 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.

978-3-903176-64-5 ©2024 IFIP

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 Industrial Control Systems (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 [7] and Censys [8] 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 [9]. 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 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 neglected, obsolete and abandoned devices. 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 [10], [11], [12]. The methods for scanning the Internet are well-established [13] and most authors use off-the-shelf common tools such as those from the ZMap ecosystem [14] or Masscan [15], alongside meta-scanners (e.g., Shodan and Censys), and IP reputation services (e.g., Virustotal [16] 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 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 proceeding 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, while [2] 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.

III. 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.

A. Scanning the Internet

Drawing on the latest trends in the literature [14], [17], [25], [26], 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 [13]. 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 [27]. 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.

B. Ethical considerations and limitations

Conducting Internet-wide scans produces a substantial load of traffic on target networks [28], [13]. 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 [14], 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 (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 V-B.

C. 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 [17], [14], [29]. Adding to this, we are aware of the behavior of some controllers exposing RTPS services that will not stop transmitting data for long periods [30].

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 [6], 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 [31] 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 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.

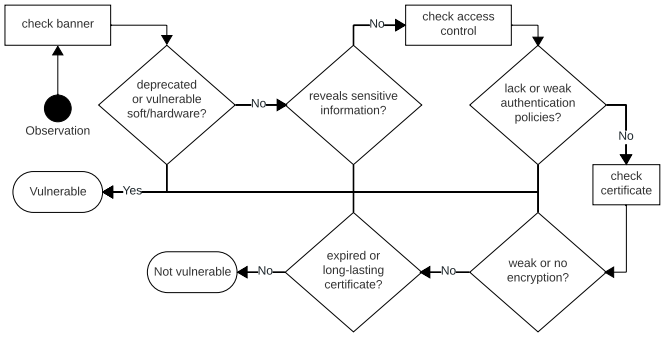

Fig. 1. Decision pipeline to identify neglected, abandoned, and obsolete devices.

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	

TABLE I
SUMMARY OF EXPOSED AND VULNERABLE SERVICES PER PROTOCOL
PROBE: ○ DEFAULT, ◐ MODIFIED, ● NEW

IV. 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 I.

A. 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.

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.

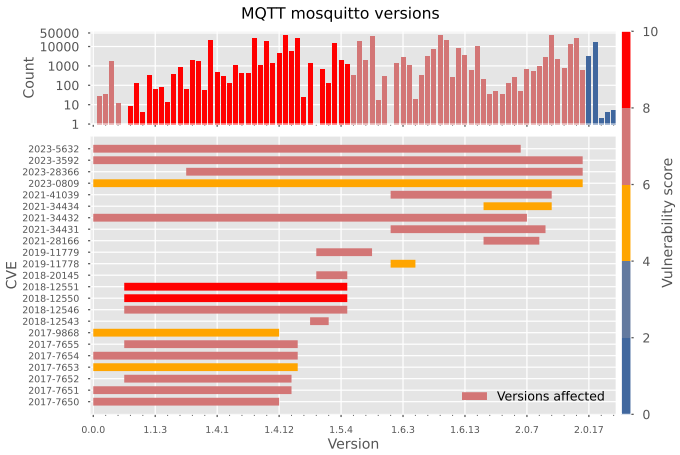

Fig. 2. Distribution of MQTT Mosquitto versions found in our dataset (top), and their vulnerabilities (bottom) colored by severity score.

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 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.

B. 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 targets 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.

C. 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 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,

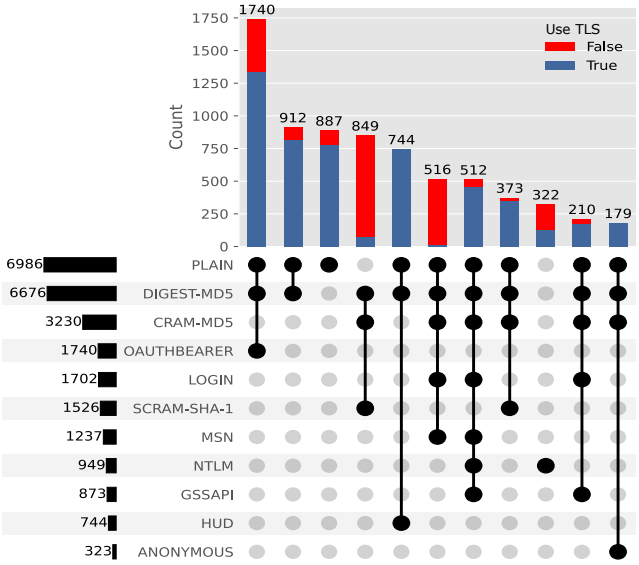

Fig. 3. XMPP top 10 most frequent combinations of authentication mechanisms and count of observations using TLS.

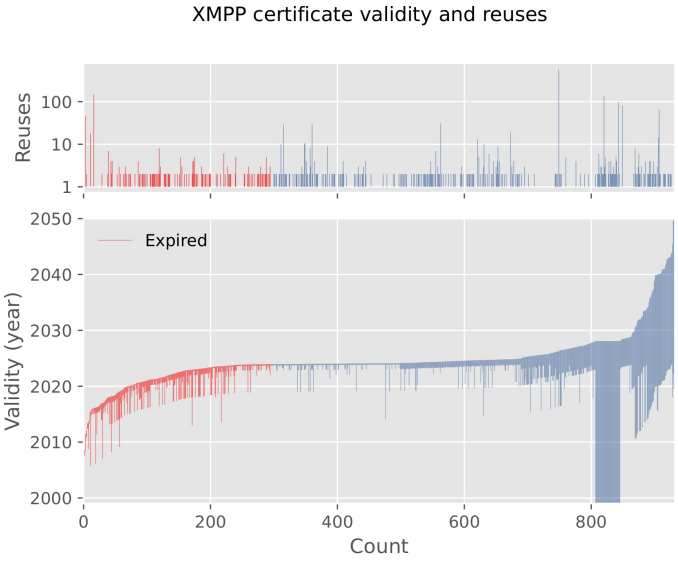

Fig. 4. Validity of XMPP certificates and reuses. On the top, is the count of reuses for each certificate. On the bottom, the validity ranges for each certificate. Expired certificates are represented in red.

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 [32]. 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 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.

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 triggers a response 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 [21]. Our dataset contains 299 different products from over 80 vendors. Figure 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

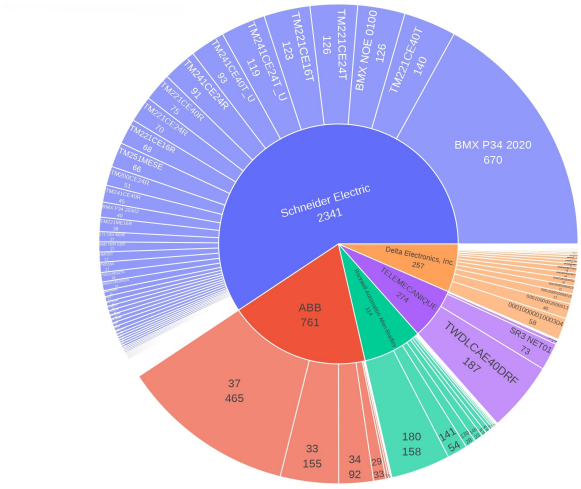

Fig. 5. Top 5 vendor distribution of products exposing Modbus services on the Internet.

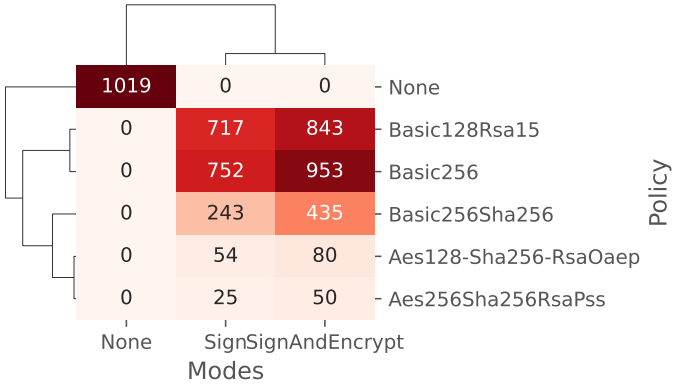

Fig. 6. OPC UA combinations of security modes and encryption policies.

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.

E. 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 [18]. 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 [18]. 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 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 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 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.

F. RTPS

RTPS is a publish-subscribe protocol used in real-time communications between distributed systems. RTPS is the

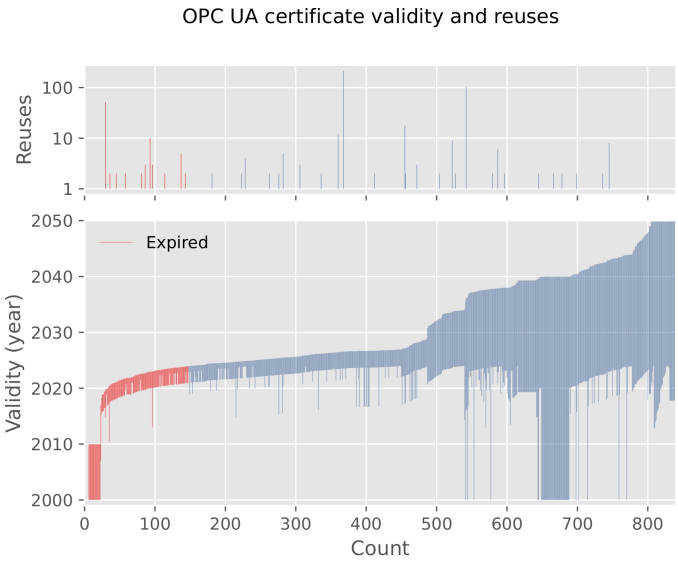

Fig. 7. Validity of OPC UA certificates and reuses. On the top, is the count of reuses for each certificate. On the bottom, the validity ranges for each certificate. Expired certificates are represented in red.

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 [30], 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.

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

G. 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 [34]. 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, 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.

H. 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 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.

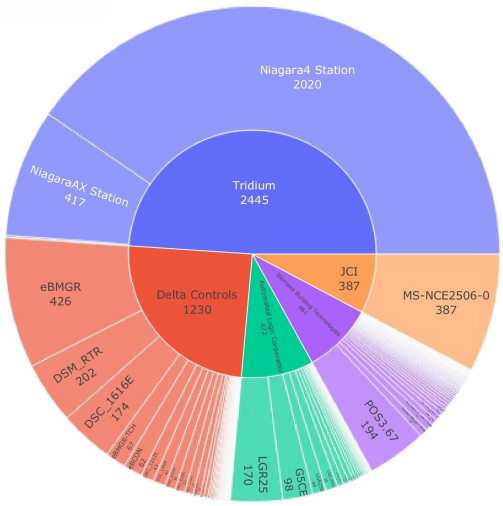

Fig. 8. Top 5 vendor distribution of products exposing BACnet services on the Internet.

Takeaway **-** Without built-in security measures to protect BACnet communications, controllers and monitoring systems depend on their infrastructure to prevent exposure to the Internet [35]. 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.

V. Discussion

A. 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 I 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 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.

B. 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 [36]. 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 [36], 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 [37], [38], [39]. Other authors raised their concerns regarding notification campaigns and the minimal impact on consumer behavior [40], [36], [41]. In general, the majority of notifications go unnoticed, are ignored, bounce back, or receive automated responses.

C. 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 [42], [43]. However, it is a shared responsibility between them to maintain device security [44], [38].

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.

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 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.

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

[8] "Censys — attack surface management," https://go.censys.com/, (Accessed on 03/01/2024).

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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

[15] "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).

[16] "Virustotal," https://www.virustotal.com/, (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] 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.

[26] 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

[27] "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).

[28] 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

[29] 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

[30] 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.

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

[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] "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).

[35] 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.

[36] 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.

[37] 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

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

[39] 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.

[40] 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

[41] 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

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

[43] 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.

[44] 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.