

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

Ricardo Yaben

*Dep. of Applied Mathematics and Computer Science  
Technical University of Denmark  
Kongens Lyngby, Denmark  
rmyl@dtu.dk*

Emmanouil Vasilomanolakis

*Dep. of Applied Mathematics and Computer Science  
Technical University of Denmark  
Kongens Lyngby, Denmark  
emmva@dtu.dk*

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Internet on port 4840 suffer from various vulnerabilities associated with misconfigurations and abandonment, such as outdated software, broken access control, and certificate management issues.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

**Index Terms**—OPC UA, Internet-wide scans, OT, ICS

## 1. Introduction

As one of the few technologies within the Operational Technology (OT) space following Secure by Design principles [1], OPC UA stands out as a protocol that allows interoperable integration between vendors and products, abstracting legacy protocols lacking security features under a unified standard [2]. OPC UA offers many security features required for today's communications between remote devices, such as encryption, authentication, and granulated access control [3].

However, previous studies show that there are thousands of insecure OPC UA servers exposed to the Internet [4], [5], with common issues such as supporting deprecated or anonymous authentication methods, (re)using insecure certificates, and allowing untrusted clients to browse freely through the controllers linked to the server. Other authors pointed out that many implementations, manuals, and setup guidelines omit these security features in the first place [6], [7], [3], potentially leading owners to deploy insecure servers without realizing the risks. In addition, system owners do not always follow best security practices and leave their devices unattended for long periods, slowly drifting away from a secure state.

This paper explores today's security state of OPC UA servers exposed to the Internet through an Internet-wide scan on its two default ports: 4840 and 4843 (TLS) [8]. Our study strives to identify common and recurrent security pitfalls, emphasizing issues stemming from poor maintenance and misconfigurations. To this end, we examine certificates, access control mechanisms, and product versioning. Our analysis offers guidance for operators to detect weaknesses and lays the groundwork for refining OPC UA deployment manuals and security recommendations to prevent the proliferation of vulnerable servers. Our main contributions are as follows:

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

The remainder of this paper is structured as follows. Section 2 gives an overview of OPC UA and its security properties. Section 3 includes a brief walk-through of the previous work about Internet measurements for OPC UA and other protocols used in OT. Section 4 describes our scanning methodology, and ethical considerations and limitations of this study. In Section 5 we cover the results of our analysis and principal findings. Then, Section 6 discusses the general aspects of our security concerns of OPC UA servers facing the Internet as well as potential future work. Lastly, we conclude this paper Section 7 with a brief summary of our work and the key takeaways.

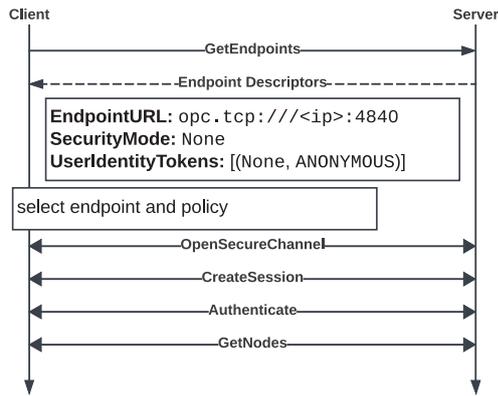


Figure 1: Communication steps with OPC UA servers to authenticate and retrieve nodes from endpoints.

## 2. Background

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

Figure 1 shows how clients establish sessions with OPC UA servers and their endpoints. To find OPC UA endpoints, clients can send discovery requests to OPC UA servers, which will respond with endpoint descriptors (this step is not required and can be skipped when the target endpoint is already known). Each endpoint descriptor includes a `SecurityMode` field indicating the level of confidentiality supported. In addition, descriptors include the field `UserIdentityTokens` with combinations of authentication methods and security policies allowed (`UserTokenPolicy`) – clients must choose one if authentication is required. In cases where the security policy is not included in this field, clients must default to the global value from the descriptor. Security modes and encryption policies are used to establish a communication channel between the client and server; once established, clients authenticate using the advertised method. However, most combinations of authentication and security policies and modes are not safe for Internet communications and should be avoided.

Security modes appear in three flavors: `None`, `Sign`, and `SignAndEncrypt`. Advertising `None` as security mode carries severe security implications, allowing clients to join the endpoint anonymously with an unencrypted session; similarly, `Sign` does not offer confidentiality either and should be disabled in all Internet-facing servers. Furthermore, the OPC UA Foundation has deprecated two policies that rely on SHA1 for signing: `Basic256` and `Basic128Rsa15`. Endpoints can choose not to offer encryption, which, together with deprecated policies, compromises the session’s privacy. In addition, clients can authenticate using one or more of the following methods supported by the endpoint: with an anonymous user, a set of credentials, a token, or an v3 X.509 certificate. Anonymous login must be disabled in all cases, as it al-

TABLE 1: List of authentication methods, security policies, and security modes supported by OPC UA endpoints. Insecure options for Internet communications colored in red.

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

lows unknown clients to join endpoints without providing any proof of identity [9], [7]. Moreover, endpoints must refuse to authenticate clients with untrusted certificates. In practice, this leaves Internet-exposed OPC UA servers with a few viable options to authenticate clients, as they must discard dangerous authentication methods (1 or 2 out of 4), security modes without confidentiality (2 out of 3), and deprecated or insecure policies (3 out of 6). Table 1 summarizes the possible values to authenticate into OPC UA endpoints, with insecure options marked in red. It should be noted that other security policies exist (e.g., supporting elliptic curves), though our dataset does not contain any examples.

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

## 3. Related Work

The research community’s efforts have brought numerous studies covering security issues in OT [10], [11], [12], [9], [13]. The state of the literature is rich and diverse, with many different methods to identify vulnerabilities, propose mitigation strategies, and raise awareness. Notably, Gao et al. [14] spoke about the security issues with SCADA systems exposed to the Internet while proposing multiple mitigation strategies. Then, Mirian et al. [15] analyzed the state of vulnerable Internet-facing Industrial Control Systems (ICS) through multiple Internet-wide scans with probes for 10 protocols, in addition to deploying honeypots to identify those conducting similar scans; however, OPC UA is not covered in their work. Moreover, Nawrocki et al. [16] used passive scanning methods to identify ICS traffic through an IPX, focusing on developing new mitigation strategies against malicious activities.

Related work in OPC UA is limited and often part of larger studies. This is expected from emerging technologies with limited adoption and specific use cases. Nevertheless, OPC UA is a technology primarily used in critical systems with higher security requirements, thus

needing further attention from the community. Here we highlight four studies tackling security concerns in OPC UA implementations and Internet-exposed services.

The work of Dahlmanns et al. [17] analyzes the use of TLS in industrial IoT systems, covering OPC UA and 10 other protocols. Their results indicate that none of the 2,193 OPC UA servers they found support TLS. In addition, they found no evidence of OPC UA servers exposed on the default port for TLS communications (4843). However, these results are likely an overgeneralization and must be interpreted carefully. There are numerous reasons to refuse connections from unknown clients. Erba et al. [7] takes a different direction in assessing OPC UA security implementations. They evaluate vendor implementations of the protocol in 22 products and 16 libraries, including their manuals and example setups. Their analysis of the implemented security features reveals several issues in all libraries and 15 vendor products.

To date, Dahlmanns et al. [4] is the closest work to our study, which focuses on security issues found in Internet-facing OPC UA servers. They analyzed seven months' worth of weekly Internet-wide scans to uncover insecure OPC UA server implementations. Their analysis covers the distribution of manufacturers and products, security policies and authentication methods, and access control and certificate-related issues. Their results suggest that 92% of the 1,114 OPC UA servers facing the Internet suffered from severe security issues. In addition, the authors analyze the number of servers reappearing throughout their study and carry out a responsible notification campaign. However, they received limited feedback, aligning with concerns expressed in similar studies. Recently, we conducted a similar study covering multiple protocols used in IoT and OT devices, including OPC UA [5]. Our goal was to uncover security issues associated with neglected, obsolete, and abandoned devices (diverging from common vulnerability reports) to identify human errors, misuse, lack of maintenance, and poor security hygiene. We reported a significant increase in OPC UA servers compared to [4], identifying 1,797 exposed servers, of which 1,210 were vulnerable, and 30 showed malicious behavior. However, we used a less intrusive probe; this probe was sufficient to survey the Internet for exposed OPC UA servers, but it limited our evaluation of the issue. Overall, these studies (i.e., [4], [5]) miss valuable insights (e.g., product and version distributions, location, and usage) to determine where and how OPC UA fails.

The rapid developments in OPC UA and their proposed security measures call for new Internet measurements to identify security pitfalls in today's deployments. Previous studies have shown that even certified OPC UA products suffer from multiple security issues [7], [18], such as hardcoded certificates, deprecated authentication mechanisms, and insecure examples in their documentation. Others highlighted the challenges of deploying and maintaining secure OPC UA servers, and the limited feedback received after notifying their owners [17], [4], [5]. In this study, we offer a granulated analysis of OPC UA deployments in the wild, covering security issues across endpoints and nodes beyond what has been examined in the related work. In addition, we give further details on products, versions, and locations to connect particular issues.

## 4. Scanning Methodology and Ethical Considerations

In February 2025, we deployed a scanning campaign to identify OPC UA servers facing the Internet. The campaign targeted two ports commonly used for discovery services: 4840 and 4843 (TLS). Our scans are divided into two phases. First, we use ZMap [19] to identify hosts accepting TCP communications at either of the targeted ports. ZMap is a stateless L4 scanner capable of sweeping the entire IPv4 in a matter of hours using gigabit network interfaces. Then, we scan the responding hosts using the OPC UA probe provided by Dahlmanns et al. [4] (with minor modifications) for Zgrab2 [20], an L7 scanning tool from the ZMap family designed to capture banner information. This probe crawls OPC UA servers, mapping endpoint applications known to the discovery server and their resources. In addition, the probe attempts to access endpoints with security features disabled using an anonymous user and a self-signed certificate. However, this probe cannot identify OPC UA servers communicating over UDP.

Regarding our experimental setup, we used a single vantage point hosting a website along the scanner with information about our research, scanning methodology, and contact information to opt out of our studies [21]. In addition, we use a blocklist to remove several IP ranges, including local and private networks, reserved spaces, networks that belong to government institutions around the globe, various network telescopes, and those who previously requested to be removed from ours or similar studies, accounting for roughly 25% of the IPv4 address space. Lastly, each connection is limited to 30 seconds per host and includes identifiers to help administrators distinguish traffic from our scanner. Note that this vantage point was recently used in other Internet surveys, which may have impacted our results.

To identify security issues, we focus on three principal aspects of the communication with exposed OPC UA servers: access control, certificates, and device meta-data. We evaluate access control issues based on the depth to which our probe accesses internal resources: first, i) access to one or more endpoints, ii) authentication using anonymous credentials or self-signed certificates, and iii) browsing through the registered nodes and retrieving their values. As pointed out in previous research [4], [5] as well as by the OPC Foundation security guidelines [22], endpoints must implement access control policies and limit communications with untrusted clients. Furthermore, we analyze server certificates for weak cryptography, including short key lengths, re-usage issues, and expired or long-lasting validity periods (while the common recommendation is 1 year, OPC UA defaults to 5 years as of [23], [24], [22]). In addition, we analyze the device metadata exposed during the communication, such as manufacturers, products, and versions, usage indicators, and implementation details.

## 5. Results

Despite the numerous revisions and security improvements OPC UA has received recently, and the efforts of

TABLE 2: Summary of OPC UA servers facing the Internet, with a breakdown of security issues found on each server. Totals show servers with one or more issues within the same category.

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

the OPC Foundation to provide security guidelines [22], the landscape of insecure OPC UA servers facing the Internet has only continued to grow [4], [5]. The OPC Foundation has contributed to CISA’s Security by Demand and Secure by Design initiatives [25], which aim to mitigate many security challenges owners face while deploying OT products. However, such initiatives are not categorical requirements, but recommendations and suggestions that often fail to get through. In this section, we analyze the observed OPC UA servers and their security implementations to identify common pitfalls and compare them with the results from previous studies (see Section A for detailed explanations on the evaluation criteria).

### 5.1. OPC UA servers

Our dataset contains responses from 1,812 OPC UA servers facing the Internet on port 4840, and 299 in port 4843. Based on this count, our probe retrieved endpoint information from 1,203 servers on port 4840. The rest responded with various errors, or the connection timed out before accessing any endpoint. On the one hand, these results align with the findings from [4] and [5] with marginal differences. On the other hand, none of the servers found in port 4843 allowed us to communicate past the discovery request. This contradicts the findings in [17], whose results showed no evidence of TLS use on exposed OPC UA servers. This may be due to a limitation in their probe, preventing the authors from determining when legitimate OPC UA servers refuse to communicate. We overcome such issues by capturing all traffic during the scan, instead of relying on our probes alone. Table 2 summarizes the number of OPC UA servers and their vulnerabilities (1,122 vulnerable servers in total). Furthermore, by comparing our results with the public dataset from Dahlmanns et al. [26] and our previous study’s dataset from [5], we could partially verify that nearly 25% of these servers were also observed in the past, having received none or minimal updates since.

### 5.2. Endpoint security

As previously discussed in Section 2, each endpoint includes the security mode, and combinations of secu-

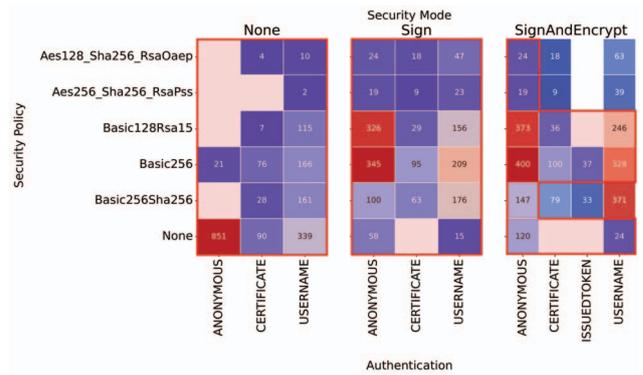


Figure 2: Combinations of security modes with authentication methods and security policies used across endpoints. Insecure combinations masked in red.

rity policies and authentication methods they support (cf. Section 2). Figure 2 shows the combinations we observed across all accessible endpoints on each server. The figure shows that most endpoints support other insecure combinations of modes and policies. Although it is recommended to use Basic256Sha256 as the minimum security policy [22], most systems were configured with the weakest option: anonymous access and no security at all. Overall, advertising multiple authentication methods allows clients to select the weaker ones to access the endpoint. When paired with the absence of TLS, methods that offer no security, or deprecated security policies, such as Basic128Rsa15 and Basic256, servers become susceptible to privacy and access control issues, from eavesdropping to an array of attacks bypassing authentication. Our dataset contains multiple examples, with servers advertising certificate-based authentication methods using deprecated policies. Allowing anonymous authentication mechanisms or deprecated policies should be a security concern for all device owners, even in the presence of other security measures. These concerns apply to 902 of the 1,203 servers exposing one or more endpoints, roughly 50% of the OPC UA servers facing the Internet. Supporting these authentication methods defeats the security measures that OPC UA implements. This suggests that their owners choose OPC UA for convenience rather than security purposes, which may be associated with the absence of TLS. It is worth mentioning that, while a significant number of servers advertise just one endpoint (223), the majority advertise between 2 and up to 30 different endpoints (see Figure 4). For reference, our dataset contains 4,569 different endpoints.

### 5.3. Manufacturers, products, and versions

As depicted in Table 2, for 534 servers, we could authenticate to one or more endpoints and crawl their nodes to identify the device manufacturer, product name, and firmware versions. It is important to remember that the level of access to most nodes is limited to reading their descriptors (e.g., node names, data types, and access), and neither read nor write their values. In fact, our dataset does not contain any nodes writable by our user, as opposed to the findings in [4]. However, we still consider it a significant risk to allow unknown clients to gain

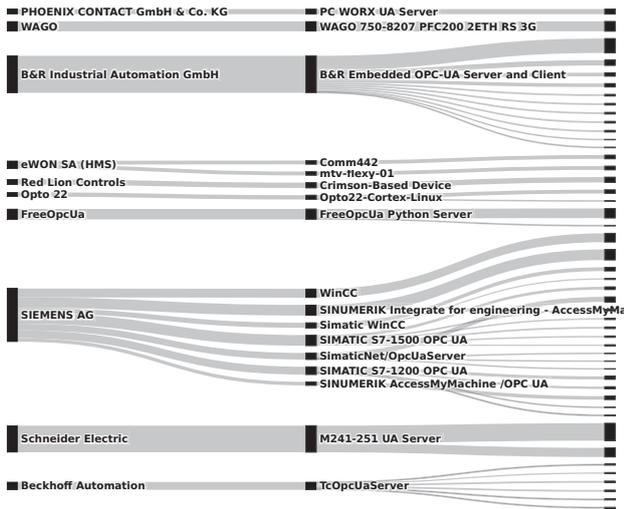


Figure 3: Distribution of the most common manufacturers and products, and their firmware versions.

knowledge of endpoints and nodes. In total, 66% (357) of the endpoints from which we could read nodes disclosed their system information. Figure 3 shows the distribution of the most common products, led by Siemens (210), B&R Industrial Automation GmbH (146), and Schneider Electric (105) devices. These devices included multiple control and alarm systems overseeing various tanks, as well as building automation systems. At the other end of the spectrum, we also found systems related to critical infrastructure, such as an oil pipeline (their owners have already been notified). To the best of our knowledge, none of the manufacturers had ceased operations, nor products we found were deprecated, which is a common issue with OT devices. With few exceptions, OPC UA servers and those with the same Fully Qualified Domain Name (FQDN) use the same type of product for all endpoints. On the other hand, more than 95% of these servers use non-certified products. Moreover, we highlight multiple old versions that these devices are running on: with build dates ranging between 2011 to 2025, with quantiles at 2016 (25%) and 2021 (75%), and median at 2019. Several products contain known vulnerabilities, enabling attackers to bypass authentication entirely, deplete resources, or cause DoS (Denial of Service). For example, some servers were running KEPServerEX in a deprecated version, allowing attackers to crash the server and remotely execute code <sup>1</sup>.

In many cases, risk factors accumulate, leading to a heightened state of vulnerability. OPC UA servers with several issues, such as broken access control and legacy builds, show signs of abandonment, a predictive aspect of devices potentially compromised [5]. To explore this further, we gather additional information on their origin and ownership using the RIPE Atlas service [27], while leveraging AbuseIPDB [28] and Greynoise [29] to identify hosts engaged in unsolicited traffic. From these records we could see that the majority of servers were located in China (318), the United States (242), Germany (182), and South Korea (93), with the major providers being Alibaba (170), Akami Cloud (169), Deutsche Telekom (99) and

1. <https://www.cve.org/CVERRecord?id=CVE-2020-27265>

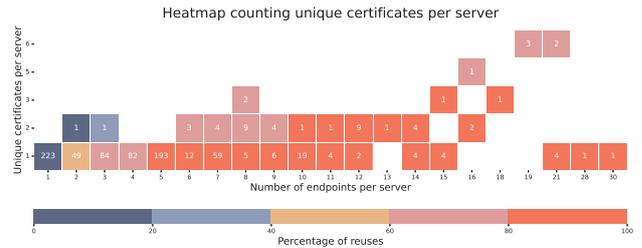


Figure 4: Certificate reuses across endpoints per server. Counts represent servers, and colors the ratio of certificate reuses within the server.

Korea Telecom (40); this suggests that most OPC UA servers exposed to the Internet are routed through cloud services, which may be an important factor to consider when suggesting implementations and best security practices while deploying OPC UA servers.

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

#### 5.4. Certificate management

Out of the 1,203 servers exposing endpoints, we could retrieve endpoint certificates from 802. As for the rest of the servers, these only allowed credential-based authentication methods (i.e., anonymous, username, or tokens) without encryption. A major recurring issue we observed was the reuse of certificates across endpoints within the same server, affecting 72% (578) of the servers. Figure 4 shows the distribution of endpoints per server with certificates, where the colors represent the ratio of reuses within the same server. This figure depicts a daring landscape, with severe cases of servers exposing more than 20 different endpoints with the same certificate. While reusing a certificate across multiple machines may be convenient, it carries significant risk—if one certificate is compromised, the entire infrastructure is at stake.

Besides reuses, 156 servers had endpoints with expired certificates, and 94 with long-lasting certificates that may never expire (> 50 years, the previous default value of auto-generated certificates). Expired and long-lasting certificates are not categorically vulnerable, but a hint at the level of maintenance these servers receive. In the device’s security lifecycle, renewing certificates helps owners verify the device’s health and maintain its reliability for the duration of the certificate or until it is revoked. As we observed these issues in tandem with broken access control and reuses across servers within the same FQDN, these organizations likely struggle to configure OPC UA servers and implement security hygiene strategies. Another possible explanation is that these devices were configured by default with hardcoded certificates. However, our dataset contains multiple instances of other servers using the same devices but different configurations (e.g., versions and certificates). Figure 5 summarizes the certificates we found across endpoints and servers, showing the validity

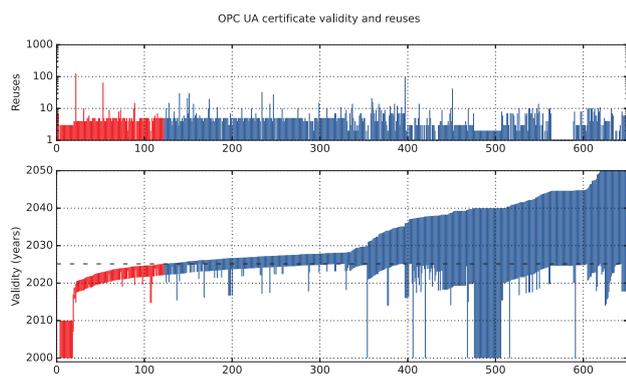


Figure 5: Certificate reuses (top) and validity periods (bottom). Expired certificates are colored in red.

period for each unique certificate we found (bottom), and the number of reuses (top), with expired certificates colored in red.

Moreover, 182 servers were found with endpoints using certificates with deprecated signing algorithms relying on SHA1. In all cases, encryption is handled by RSA, while signing is done by either SHA1 (897), SHA256 (2,542), or SHA512 (5). This is a reduction of almost 50% over the results from [4], suggesting there is some level of improvement. Additionally, 739 certificates with SHA1 use 1024-bit keys, and the same problem appears in 69 certificates with SHA256. The standard recommendation suggests 2048-bit keys since 2015, when 1024-bit keys were officially phased out. Despite other issues, 2048-bit was the most common key length among certificates, with some instances reaching over 3072 and 4096-bit keys.

## 6. Discussion

The majority of security concerns we identified throughout this paper were related to access control issues, security maintenance, and certificate management, aligning with the findings of related work [4], [5]. Our findings show that OPC UA servers communicating over TLS appear secure for the most part (i.e., refusing to communicate), but those lacking it face a multitude of vulnerabilities.

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

fix, long-term risk mitigation requires additional measures. These vulnerabilities often stem from poor security practices, highlighting the need for better training and awareness.

Regardless of these issues, we remain optimistic about the OPC UA landscape. The OPC UA Foundation already provides clear security recommendations, such as regular updates, certificate management, and secure deployment practices. In addition, the research community continues to point out the remaining challenges observed in Internet-facing servers, manuals, guidelines, and products. To expand on these efforts, we recommend further guidelines for secure deployments in cloud services and provisioning strategies for remote controllers.

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

## 7. Conclusion

As one of the few protocols within OT offering security and privacy out of the box, OPC UA soars among the competition to create safe environments accessible from remote locations. Moreover, OPC UA stands out for its approach to abstract legacy protocols instead of replacing them, allowing system owners to transition to this new technology as it becomes available in more products. However, deploying OPC UA servers is not trivial, as system owners frequently struggle to configure their servers and maintain them secure. Failing to follow the security guidelines offered by the OPC UA Foundation may lead to severe security risks [30], [22], an issue that is only worsened without proper security maintenance.

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

## Acknowledgments

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] “Ua part 2: Security - 4 opc ua security architecture,” [Accessed 2025-02-25]. [Online]. Available: <https://reference.opcfoundation.org/Core/Part2/v105/docs/4>
- [2] O. U. Foundation, “Unified architecture - landingpage - opc foundation,” [Accessed 2025-02-25]. [Online]. Available: <https://opcfoundation.org/about/opc-technologies/opc-ua/>
- [3] F. O. for Information Security, “Opc ua security analysis,” 06 2022, [Accessed 2025-02-25]. [Online]. Available: <https://opcfoundation.org/wp-content/uploads/2023/11/BSI-OPCUA-2022-EN.pdf#page=10.07>
- [4] 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>
- [5] 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.
- [6] N. Mühlbauer, E. Kirdan, M.-O. Pahl, and G. Carle, “Open-source opc ua security and scalability,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 262–269.
- [7] A. Erba, A. Müller, and N. O. Tippenhauer, “Security analysis of vendor implementations of the opc ua protocol for industrial control systems,” in *Proceedings of the 4th Workshop on CPS & IoT Security and Privacy*, ser. CPSIoTSec '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1–13. [Online]. Available: <https://doi.org/10.1145/3560826.3563380>
- [8] “Service name and transport protocol port number registry,” [Accessed 2025-04-14]. [Online]. Available: <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?&page=86>
- [9] V. M. Ijure, S. A. Laughter, and R. D. Williams, “Security issues in scada networks,” *Computers & Security*, vol. 25, no. 7, pp. 498–506, 2006. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167404806000514>
- [10] M. Henze, J. Hiller, R. Hummen, R. Matzutt, K. Wehrle, and J. H. Ziegeldorf, “Network security and privacy for cyber-physical systems,” *Security and Privacy in Cyber-Physical Systems*, pp. 25–56, 11 2017. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/9781119226079.ch2>
- [11] A. R. Sadeghi, C. Wachsmann, and M. Waidner, “Security and privacy challenges in industrial internet of things,” *Proceedings - Design Automation Conference*, vol. 2015-July, 7 2015. [Online]. Available: <https://dl.acm.org/doi/10.1145/2744769.2747942>
- [12] M. Krotofil and D. Gollmann, “Industrial control systems security: What is happening?” in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*. IEEE, 2013, pp. 670–675.
- [13] A. Nicholson, S. Webber, S. Dyer, T. Patel, and H. Janicke, “Scada security in the light of cyber-warfare,” *Computers & Security*, vol. 31, no. 4, pp. 418–436, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167404812000429>
- [14] J. Gao, J. Liu, B. Rajan, R. Nori, B. Fu, Y. Xiao, W. Liang, and C. L. P. Chen, “Scada communication and security issues,” *Security and Communication Networks*, vol. 7, pp. 175–194, 1 2014. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/sec.698>
- [15] 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)*, Dec. 2016, pp. 96–103.
- [16] M. Nawrocki, T. C. Schmidt, and M. Wahlisch, “Uncovering vulnerable industrial control systems from the internet core,” *Proceedings of IEEE/IFIP Network Operations and Management Symposium 2020: Management in the Age of Softwarization and Artificial Intelligence, NOMS 2020*, 4 2020.
- [17] 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,” *ASIA CCS 2022 - Proceedings of the 2022 ACM Asia Conference on Computer and Communications Security*, vol. 22, pp. 252–266, 5 2022. [Online]. Available: <https://dl.acm.org/doi/10.1145/3488932.3497762>
- [18] P. Cheremushkin and S. Temnikov, “Opc ua security analysis,” *Kaspersky Lab ICS CERT*, 2018.
- [19] 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>
- [20] “zmap/zgrab2: Fast application layer scanner,” [Accessed 2025-02-25]. [Online]. Available: <https://github.com/zmap/zgrab2>
- [21] D. C. S. for cybersecurity engineering, “Technical university of denmark (dtu) - internet scanner,” [Accessed 2025-04-12]. [Online]. Available: <http://130.226.254.28/>
- [22] O. Foundation, “Practical security recommendations for building opc ua applications,” [Online; accessed 2025-02-18]. [Online]. Available: <https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Security-Advise-EN.pdf>
- [23] “Providing opc ua client instance certificate,” [Accessed 2025-04-13]. [Online]. Available: <https://opclabs.doc-that.com/files/onlinedocs/QuickOpc/Latest/User's%20Guide%20and%20Reference-QuickOPC/Providing%20Client%20Instance%20Certificate.html>
- [24] “Certificate generator,” [Accessed 2025-04-13]. [Online]. Available: [https://opcfoundation.github.io/UA-.NETStandard/help/certificate\\_generator.htm](https://opcfoundation.github.io/UA-.NETStandard/help/certificate_generator.htm)
- [25] I. S. Agency, N. S. Agency, F. B. of Investigation, E. P. Agency, T. S. Administration, and I. Partners, “Tlp:clear secure by demand: Priority considerations for operational technology owners and operators when selecting digital products,” 2025.
- [26] M. Dahlmanns, “Dataset to ”easing the conscience with opc ua: An internet-wide study on insecure deployments” - rwth publications,” [Online; accessed 2025-02-19]. [Online]. Available: <https://publications.rwth-aachen.de/record/802060>
- [27] “Ripe atlas - dashboard,” [Accessed 2025-02-25]. [Online]. Available: <https://atlas.ripe.net/>
- [28] “Abuseipdb - ip address abuse reports - making the internet safer, one ip at a time,” [Accessed 2025-02-25]. [Online]. Available: <https://www.abuseipdb.com/>
- [29] “Greynoise intelligence — real-time intelligence for modern threats,” [Accessed 2025-02-25]. [Online]. Available: <https://www.greynoise.io/>
- [30] F. Kohnhauser, D. Meier, F. Patzer, and S. Finster, “On the security of iiot deployments: An investigation of secure provisioning solutions for opc ua,” *IEEE Access*, vol. 9, pp. 99 299–99 311, 2021.

## Appendix A. Evaluation criteria

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

TABLE 3: Evaluation criteria to identify vulnerable OPC UA servers exposed to the Internet

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

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

Previous measurements covering OPC UA highlighted the issue of thousands of servers exposed to the Internet without support for TLS [5], [17], [4]. However, they do not offer further insights regarding the devices themselves, such as their type, model, location, or firmware version. These details can help us in better understanding their risks and form an impression on whether these issues are localized (e.g., to one manufacturer, geographical location, community, etc.) or common to all. Part of this information is often available within many endpoints to help operators manage their assets. However, unauthorized access to this information may also lead to further attacks. Therefore, our probe must test for access control issues and attempt to access unprotected internal information. Our previous probe used in [5] severed connections immediately after testing for authentication issues, without browsing nodes or testing access control levels. On the other hand, the probe from Dahlmanns et al. [4] required minimal modifications to handle this corner-case. With our changes, the probe handles servers requiring TLS, captures certificates, iterates endpoints advertised by discovery servers, attempts to authenticate using an anonymous guest user and a self-signed certificate, and browses nodes at particular indexes (instead of exhausting all nodes, which may be in the thousands of requests). This probe allows us to expand on the literature and continue the work we presented in [5], covering misconfigurations and issues associated with human behavior.

The security vulnerabilities we discuss in this paper are tightly linked to (not following) the security rec-

ommendations from the OPC UA foundation on secure server deployments and maintenance. In summary, these recommendations guide operators to meet their security goals, covering certificate management, authentication, and access control – the same we cover here. Table 3 includes a summary of the criteria we use to classify OPC UA servers as vulnerable or not. The remainder of this section covers each criterion individually.

**Certificate management.** To detect certificate management issues, we first collect all certificates from all endpoints and servers, then cross-reference them to test for reuse. This process reveals reuse within and across servers globally, potentially uncovering unintended connections and severe issues like hardcoded certificates. However, we do not identify already compromised certificates. We also evaluate certificate validity periods against the default five years and match them to the scan date, helping identify security negligence such as expired or overly long certificates. Finally, we assess cryptographic properties to detect weak algorithms, short key lengths, and reuse.

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

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