Protocol Security in the Industrial Internet of Things

Markus Dahlmanns and Klaus Wehrle
Communication and Distributed Systems, RWTH Aachen University, Germany
{dahlmanns, wehrle}@comsys.rwth-aachen.de

***Abstract*—Advances like Industry 4.0 lead to a rising number of Internet-connected industrial deployments and thus an Industrial Internet of Things with growing attack vectors. To uphold a secure and safe operation of these deployments, industrial protocols nowadays include security features, e.g., end-to-end secure communication. However, so far, it is unclear how well these features are used in practice and which obstacles might prevent operators from securely running their deployments.**

In this research description paper, we summarize our recent research activities to close this gap. Specifically, we show that even secure-by-design protocols are by far no guarantee for secure deployments. Instead, many deployments still open the doors for eavesdropping attacks or malicious takeovers. Additionally, we give an outlook on how to overcome identified obstacles allowing operators to configure their deployments more securely.

***Index Terms*—industrial internet of things, security analysis**

I. Introduction

While industrial networks, e.g., for factory and process automation, traditionally were designed as isolated networks, advances like Industry 4.0 [1] significantly increase the network connectivity leading to a growing Industrial Internet of Things (IIoT). Thus, today's modern Internet-connected industrial networks offer a large variety of attack vectors that already have been exploited by several incidents, e.g., NotPetya or manipulation attacks on several industrial devices [2]. Hence, IIoT deployments require notable adaptations in security. Particularly, end-to-end secure communication via the Internet and access control are important to prevent attackers from (i) eavesdropping sensitive data and (ii) maliciously controlling production lines.

To account for these arising needs, industrial communication protocols increasingly incorporate security features. On the one hand, modern protocols, e.g., OPC UA [3] and MQTT [4], were designed with security in mind. Hence, they include tailored security mechanisms that are attested to be secure [5] (OPC UA) or were directly specified for usage via TLS, the most prevalent protocol for secure communication on the Internet (MQTT). On the other hand, manufacturers increasingly retrofit traditional industrial protocols, e.g., Modbus or EtherNet/IP, by specifying it for usage via TLS as well.

Before our studies, related work looked into the connectivity of industrial appliances using traditional insecure protocols [6], robots [7], and their communication via the Internet [8]. However, still open was whether the new streams of *secure* industrial communication protocols are used in practice, adequately configured, and thus lead to a more *secure* IIoT.

Research Questions: To close this gap and guide our research we derive the following research questions.

Q1: *Do today's IIoT deployments implement security features introduced in current communication protocols?*
↪ Recent advances for the IIoT aim to increase the security of deployments, particularly modern and retrofitted protocols, with security features. However, so far it is unclear whether operators of IIoT deployments implement these advances. Hence, we focus on analyzing the current security state of the IIoT and examine whether deployments implement best practices, e.g., secure ciphers.

Q2: *Which obstacles hinder operators of IIoT deployments from adopting security best practices?*
↪ For insecurely configured deployments it remains open which pitfalls might prevent their operators from configuring them securely. We thus target to examine insecure configurations in comparison to secure configurations and derive aspects that lead operators to such settings or prevent using modern security mechanisms.

Q3: *Which measures might improve the security of the IIoT?*
↪ After identifying insecure deployments and aspects that lead to insecurities, it is open how to support a secure IIoT in the future. To perspectively increase the security of the IIoT, we aim to propose novel approaches that can help operators to securely configure their deployments.

In our research, we intend to answer these research questions by (i) designing and performing Internet-wide measurements that help to assess the state of the IIoT at scale, (ii) analyzing our results beyond the border of each measurement as well as incorporating answers to our responsible disclosures, and (iii) transforming our insights into the IIoT's current state to the proposal of novel mechanisms that can help to increase the security of the IIoT in the future.

II. Accomplished and Planned Contributions

Backed by our research questions, we have already accomplished (☑) and planned (□) the following six contributions. Figure 1 shows how they cover our research questions.

C1: *OPC UA Security Assessment* [9] ☑

OPC UA is a comparatively new industrial protocol targeting to homogenize the IIoT by allowing cross-vendor communication, is secure-by-design, and attested to be secure [5]. However, to achieve this attested level of security, OPC UA deployments require the configuration of numerous security

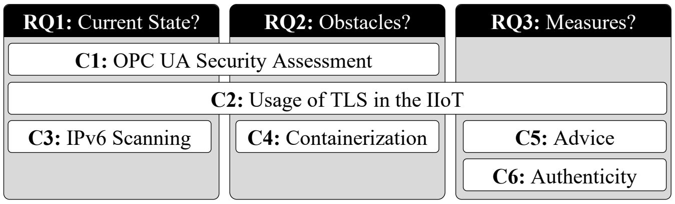

Fig. 1. Mapping of our contributions to our research questions.

settings. Hence, incautious decisions of operators lead to weak or even unsecured systems. To support operators in configuring their deployments securely, official configuration recommendations exist [10]. However, it was unclear whether operators adhere to such recommendations. Internet-wide measurements are a valuable and accepted method to assess the usage and configuration of protocols at scale as related work covering industrial devices using traditional protocols without security functionality exemplifies [6]. These advances motivated us to combine these two streams of research to analyze the security configurations of OPC UA deployments.

Using Internet-wide active measurements, in our first contribution **C1**, we unveiled that 92 % of all 1114 Internet-reachable OPC UA deployments have a deficient security configuration. We revealed that 26 % of the servers completely disabled communication security or rely on deprecated cryptographic primitives such as `SHA1`. Second, we discover the incorrect application of theoretically secure configurations on additional 35 % of systems. Partly, these systems are also affected by a systematic reuse of security-critical certificates on hundreds of systems across various ASes. Finally, we find that 44% of all servers allow unauthenticated users to read and potentially write values from devices and even execute system functionality. All in all our results underpin that *secure-by-design* protocols are no guarantee for secure deployments as operators fail to configure them securely and implementations do not support operators to make the right choices.

C2: *Usage of TLS in the IIoT* [11] ☑

Before OPC UA allowed cross-vendor communication, numerous traditional and mostly vendor-specific protocols without security features enabled communication in industrial networks. However, to address the new security requirements of the IIoT, manufacturers retrofitted their security protocols (as of 2013): They now rely on TLS to end-to-end protect the communication and allow for client authentication as well as access control, e.g., [12]. Complementing these efforts, modern protocols such as AMQP, MQTT, and CoAP, specifically targeting IIoT communication but also commonly used in the IoT, have been explicitly designed to provide security via TLS, e.g., [4]. Similar to OPC UA, these protocols need to be configured securely when deployed on industrial appliances to capitalize on their promised security benefits [13].

In our second contribution **C2**, we showed that the overall adoption of TLS for IPv4-reachable industrial deployments is comparably low and, when used, deficiently configured. Only 6.5 % of all 967 551 Internet-reachable hosts use TLS to secure their communication with a significant shift to deployments using *modern* protocols (7.2 % vs. 0.4 %), e.g., MQTT. Still, 42 % of the TLS-enabled deployments suffer from configuration issues impacting their security—even when relying on modern protocols. These issues encompass the usage of outdated protocol versions (0.6 %), ciphers (6.1 %), certificates relying on deprecated primitives (2.2 %), reuse of compromised secrets (30 %), or disabled access control (18 %). We were able to trace some of these misconfigurations back to outdated configuration templates, e.g., automated scripts, but also found that up-to-date templates can significantly help operators. Overall, we showed that the movement towards secure IIoT protocols so far has not arrived in public deployments. Aggravatingly, a large share of deployments relying on TLS are configured insecurely. Hence, operators require support in configuring security protocols.

C3: *IPv6 Scanning* [14] ☑

So far, our global assessment focused on the IPv4 address space (2^{32} addresses). While Internet-wide studies in this address space can finish in a few minutes[1] [15], they are not feasible for all 2^{128} IPv6 addresses as measurements would require 600 trillion years at the same speed. Hence, state-of-the-art Internet measurement methods resort to scanning only parts of the IPv6 address space by relying on (i) *hitlists* [16], [17] of active IPv6 addresses from different sources, e.g., DNS or email, and (ii) numerous *generators*, e.g., [18], that take seeds, such as hitlists, to produce further IPv6 addresses which might be in use and thus are valuable to scan. However, their practicability to find IoT services and thus get a global view on all Internet-reachable IIoT deployments was not well researched, i.e., related work relied on hitlists only, e.g., [19].

In our third contribution **C3**, we found 6658 IPv6-reachable IoT installations by combining eleven open-source address generators and three seedlists. We showed that not all address generators are beneficial and the seed selection significantly influences their results. Notably, we demonstrated that using two generators and address lists suffice to detect 95 % of found deployments, i.e., future studies must not necessarily employ all open-sourced generators when searching for IoT deployments helping to reduce time and computation cost. Still, all these efforts including the measurement of billions of IPv6 addresses only lead to a few thousand found deployments. Hence, other measures, e.g., gathering IPv6 addresses in use from NTP-pool servers [20] might lead to a better scan-success ratio. Security-wise, we surprisingly found similar issues in the IPv4 and IPv6 address space despite their potentially more recent deployment: Only 6.2 % of IPv6-reachable deployments implemented TLS for communication security of which 7.8 % configured it insecurely, e.g., by using deprecated cryptographic primitives such as `SHA1`. Additionally, 39 % failed to implement access control, enabling attackers to easily access potentially sensitive information.

[1] We rate-limited our scans for ethical and technical reasons [9], [11], [14].

C4: *Containerization* [21] ☑

Our contributions **C1**-**C3** showed that IIoT deployments usually reuse confidential security material across several operators. We discovered some of these secrets in publicly available container images. However, including confidential secrets such as cryptographic keys or API secrets in container images, by mistake or out of negligence, can introduce two security issues: (i) attackers can misuse compromised secrets leading to potential loss of data, privacy, or control, and (ii) administrators instantiating images can rely on broken security, e.g., paving the way for Man-in-the-Middle attacks. While blog entries also produced anecdotal evidence that Docker images include further confidential security material, e.g., in [22], comprehensive analyses on revealed security secrets at scale did not exist.

By analyzing 337 171 images from the most prominent public image registry, i.e., Docker Hub, and 8076 other private registries we unveiled that 8.5 % of images indeed include secrets. Specifically, we found 52 107 private keys and 3158 leaked API secrets, both opening a large attack surface, i.e., putting authentication and confidentiality of privacy-sensitive data at stake and even allow active attacks. We further saw that operators used these leaked keys in the wild even beyond our results from **C1**-**C3**: We discovered 1060 certificates relying on compromised keys being issued by public certificate authorities and found 275 269 TLS and SSH hosts using leaked private keys for authentication. Hence, secret leakage in container images is not limited to the IIoT.

C5: *Advice* [23] ☑

In our contributions **C1**-**C4**, we showed that the security of IIoT deployments heavily depends on their configuration and many deployments are insecurely configured. While the necessary security configurations during the setup of IT services frequently overwhelm even trained system administrators [24], e.g., when creating certificate signing requests or having to select appropriate cipher suites, in the IIoT, the situation gets even worse as the intended users are not assumed to have any knowledge about IT security. Additionally, IIoT deployments are growing increasingly complex as they often contain numerous and heterogeneous components, i.e., devices and services [25]. More aggravatingly, IIoT devices oftentimes strip away security features of the used protocols [26] to reduce production costs and energy consumption. Finally, users deploy IIoT components with various and highly individual use cases in mind, e.g., requiring or denying remote access. Overall, best practices for secure configurations cannot be transferred easily from one IIoT deployment to another.

To close this critical gap between the knowledge and effort required to securely configure IIoT deployments and the end-users' capabilities to realize such a secure configuration based on their individual needs, we enabled end-users to exchange knowledge about realizable security configurations in an automated fashion. Our approach, ColPSA, first crowd-sources real-world configurations for the whole variety of IIoT protocols and devices in a privacy-preserving manner. Then, it selects the most secure configuration ever seen for each device in each scenario and notifies operators not implementing it in their deployment. Hence, ColPSA eases protocol security assessments for users and does not recommend inapplicable or inadequate configurations. Still, ColPSA does not require extensive input on all possible device and scenario configurations but learns the best possible configurations from the crowd.

C6: *Authenticity* □

In addition to increasing the awareness of operators for insecure security configurations, we also focus on extending the security landscape with a practicable authentication scheme for the IIoT as current authenticity mechanisms are not sufficient for the IIoT. For the Web, Let's Encrypt achieved a significant increase in end-to-end secure communication [27]. However, it requires a domain and external reachability of all deployments to allow the issuance of a certificate, is thus not directly applicable and does not allow easy-to-use authentication for all IIoT deployments. Still, authentication is one of the major requirements for end-to-end secure communication. Hence, we inherit Let's Encrypt's usability for our approach to nearly allow *secure-by-default* IIoT deployments. We think that this contribution will lead to a significantly larger share of end-to-end secure communication in this realm.

III. Answers To Our Research Questions

We extract the relevant results from our presented contributions to address our three research questions. Hence, this section describes how we advance research in IIoT security.

Q1: *Do today's IIoT deployments implement security features introduced in current communication protocols?*

In our contributions **C1**-**C3**, we leveraged Internet-wide measurements to get a global view on the *realized* security in today's IIoT. Although current IIoT protocols include security features and guidelines exist to support operators in configuring their deployments securely, we showed that a majority of Internet-reachable deployments are still insecure.

Hence, today's IIoT deployments fail to live up to expectations seeded by recent protocol developments and do not widely follow security best practices. While many appliances do not even adapt secure protocol versions, others are deployed *seemingly secure*, i.e., use a secure protocol variant, but their configuration leads to massive insecurities, e.g., no authenticity or access control. Thus, a majority of Internet-reachable IIoT deployments allow attackers to eavesdrop on sensitive data or take over control.

Q2: *Which obstacles hinder operators of IIoT deployments from adopting security best practices?*

Our contribution **C2** shows that retrofitted and thus secure versions of traditional protocols do not necessarily find their way into real deployments. Only 0.4 % of found appliances that use traditional protocols adapt the retrofitted variant. Thus, getting to a more secure IIoT does not only require secure protocol versions but also *supporting devices* and operators who *enable* security features. However, as shown by our contribution **C1**, even secure-by-design protocols, e.g., OPC UA, are no guarantee that deployments indeed are secure as numerous deployments without any enabled security features exist. Additionally, many deployments rely on deprecated protocol versions or security features, e.g., `MD5`. Hence, it is hard for operators to keep up with the development in the security landscape, e.g., when security primitives lose their promises. Finally, new technologies like containerization emerge that ease the deployment of services, but, as shown in our contribution **C4**, also make it susceptible to bypass fundamental security concepts, i.e., keeping secrets secret.

Q3: *Which measures might improve the security of the IIoT?*

To enable operators to identify insecurely configured components in their deployments and receive applicable and adequate security advice, our contribution **C5** presents ColPSA, our approach to crowd-source device- and scenario-agnostic configuration possibilities. With ColPSA deployed, operators share their security configuration with a central service and get feedback for improvements. Still, due to the crowdsourcing approach, ColPSA does not require a massive amount of security expertise and also is able to identify distributed issues, e.g., certificate reuse and secret leakage.

Furthermore, to support operators in not even deploying insecurely configured components, in our contribution **C6** we focus on proposing a nearly *secure-by-default* authentication approach, inspired by Let's Encrypt and its significant improvements to secure communication on the Web. As very simple to set up, operators must not hesitate to use our approach and, as each component independently requests its certificate from a CA, operators cannot leak any secret.

IV. Conclusion

For this research description paper, we summarized our findings from nearly five years of research (2019-2024) to answer our three research questions. We focused on evaluating whether today's IIoT deployments use protocol security features, understanding which obstacles hinder operators from adopting security best practices, and which measures might improve the security of the IIoT in this regard. In turn, we showed that operators of public deployments rarely utilize retrofitted protocol variants and that even modern, secure-by-design protocols suffer from insecure configurations. As the IIoT continuously evolves our results will help to increase the security of the IIoT in the future.

We are confident that, on the one hand, our analysis results will trigger new stages in protocol development, e.g., converting from secure-by-design to secure-by-default protocols. On the other hand, we believe that our contributions already today can positively influence network operators during the security configuration and secure operation of their state-of-the-art IIoT deployments and protocols. Overall, we underpinned that, until no secure-by-default protocol exists, following security best practices requires initial and continuous effort. However, still open to discuss is, (i) how our contributions interplay with each other, i.e., how to find a convincing overall story for the dissertation, and (ii) whether there are further aspects to address to answer **RQ3**.

Acknowledgement

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy — EXC-2023 Internet of Production — 390621612.

References

[1] H. Lasi, P. Fettke *et al.*, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, 2014.

[2] K. E. Hemsley and R. E. Fisher, "History of Industrial Control System Cyber Incidents," Idaho National Laboratory, Tech. Rep., 2018.

[3] OPC Foundation, "OPC Unified Architecture — Part 2: Security Model," OPC 10000-2: OPC Unified Architecture, 2018.

[4] A. Banks and R. Gupta, "MQTT Version 3.1.1," OASIS Standard, 2014.

[5] Federal Office for Information Security, "OPC UA Security Analysis," 2017.

[6] A. Mirian, Z. Ma *et al.*, "An Internet-wide view of ICS devices," in *PST*, 2016.

[7] N. DeMarinis, S. Tellex *et al.*, "Scanning the Internet for ROS: A View of Security in Robotics Research," in *ICRA*, 2019.

[8] M. Nawrocki, T. C. Schmidt *et al.*, "Uncovering Vulnerable Industrial Control Systems from the Internet Core," in *NOMS*, 2020.

[9] M. Dahlmanns, J. Lohmöller *et al.*, "Easing the Conscience with OPC UA: An Internet-Wide Study on Insecure Deployments," in *IMC*, 2020.

[10] U. Pohlmann and A. Sikora, "Practical Security Recommendations for building OPC UA Applications," *Industrial Ethernet Book*, vol. 106, 2018.

[11] M. Dahlmanns, J. Lohmöller *et al.*, "Missed Opportunities: Measuring the Untapped TLS Support in the Industrial Internet of Things," in *ASIACCS*, 2022.

[12] Modbus Organization, "Modbus/TCP Security Protocol Specification," MB-TCP-Security-v21_2018-07-24, 2018.

[13] Y. Sheffer, R. Holz *et al.*, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)," IETF RFC 7525, 2015.

[14] M. Dahlmanns, F. Heidenreich *et al.*, "Unconsidered Installations: Discovering IoT Deployments in the IPv6 Internet," in *NOMS*, 2024.

[15] D. Adrian, Z. Durumeric *et al.*, "Zippier ZMap: Internet-Wide Scanning at 10 Gbps," in *WOOT*, 2014.

[16] O. Gasser, Q. Scheitle *et al.*, "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," in *IMC*, 2018.

[17] J. Zirngibl, L. Steger *et al.*, "Rusty Clusters? Dusting an IPv6 Research Foundation," in *IMC*, 2022.

[18] T. Yang, B. Hou *et al.*, "6Graph: A graph-theoretic approach to address pattern mining for Internet-wide IPv6 scanning," *Comput. Netw.*, vol. 203, 2022.

[19] S. J. Saidi, S. Matic *et al.*, "Deep Dive into the IoT Backend Ecosystem," in *IMC*, 2022.

[20] E. Rye and D. Levin, "IPv6 Hitlists at Scale: Be Careful What You Wish For," in *SIGCOMM*, 2023.

[21] M. Dahlmanns, C. Sander *et al.*, "Secrets Revealed in Container Images: An Internet-wide Study on Occurrence and Impact," in *ASIACCS*, 2023.

[22] M. Sequeira, "Low-hanging Secrets in Docker Hub and a Tool to Catch Them All," https://ioactive.com/guest-blog-docker-hub-scanner-matias-sequeira/, 11 2020, (Accessed on 06/13/2022).

[23] M. Dahlmanns, R. Matzutt *et al.*, "Collectively Enhancing IoT Security: A Privacy-Aware Crowd-Sourcing Approach," in *LNCS*, 2024, to be published.

[24] K. Krombholz, W. Mayer *et al.*, ""I Have No Idea What i'm Doing": On the Usability of Deploying HTTPS," in *SEC*, 2017.

[25] S. Madakam, R. Ramaswamy *et al.*, "Internet of Things (IoT): A Literature Review," *JCC*, vol. Vol.03No.05, 2015.

[26] A. Erba, A. Müller *et al.*, "Security Analysis of Vendor Implementations of the OPC UA Protocol for Industrial Control Systems," in *CPSIoTSec*, 2022.

[27] J. Aas, R. Barnes *et al.*, "Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web," in *CCS*, 2019.