Knowledge-Augmented Security Risk Identification for OT Container Deployments

Yannick Landeck[1,*], Dian Balta[1], Tomas Bueno Momcilovic[1], Martin Wimmer[2] and Christian Knierim[2]

[1]fortiss GmbH, Munich, Germany
[2]Siemens AG, Munich, Germany

Abstract

Container deployments in operational technology (OT) environments pose unique security challenges, especially when privileged configurations are used. Traditional risk identification methods often fall short in addressing the complexity, dynamic nature, and interdisciplinary collaboration required in these settings. We propose a knowledge augmentation approach that combines semantic modelling, automated reasoning, and tool support to enhance security risk identification. Our approach is demonstrated through an industrial case study, highlighting its practical application. We also examine how large language models (LLMs) can support the instantiation and integration of the approach, improving usability and scalability.

Keywords

Security Risk Identification, Knowledge Augmentation, Container Security, Operational Technology, LLMs

1. Introduction

Industries increasingly adopt containerised applications to modernise operational technology (OT) environments [1, 2]. However, containers often require runtime privileges that introduce significant security risks [3, 4]. For instance, anomaly detection on industrial networks may require deployment with `–net=host` in Docker. It is often unclear whether such privileges are necessary or if safer alternatives exist—making risk assessment essential to understand potential impacts.

Yet, risk identification is complicated by dynamic architectures, frequent changes, and the need for collaboration across roles [5, 6]. Traditional approaches, like security consulting, are too resource-intensive and fail to address these challenges. As a resolution, we propose a knowledge augmentation approach that integrates semantic modelling, automated reasoning, and tool support to help stakeholders identify risks based on deployment configurations. We demonstrate the approach in an industrial case study and explore how large language models (LLMs) support its instantiation and integration.

2. Challenges in Security Risk Identification

Identifying security risks in OT container deployments presents both technical and organisational challenges. These arise from the decoupling of development and deployment, where containers built in controlled environments are deployed in dynamic, interconnected OT systems [5, 7]. OT systems often prioritise availability and integrity over confidentiality, shifting the threat landscape compared to traditional IT environments [8]. The convergence of IT and OT introduces hybrid architectures that blur traditional security boundaries and create new dependencies and attack surfaces [9, 10].

A key issue is the fragmentation of knowledge. Security risk identification relies on stakeholders—such as developers, operators, and security experts—interpreting shared artefacts like Dockerfiles to assess

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan
*Corresponding author.
landeck@fortiss.org (Y. Landeck); balta@fortiss.org (D. Balta); momcilovic@fortiss.org (T. B. Momcilovic); martin.r.wimmer@siemens.com (M. Wimmer); christian.knierim@siemens.com (C. Knierim)
0009-0008-0340-3602 (Y. Landeck); 0000-0001-8311-3227 (D. Balta); 0000-0003-4503-2244 (T. B. Momcilovic); 0009-0009-2716-8886 (M. Wimmer); 0000-0002-5713-4654 (C. Knierim)

risks [11, 6]. However, these artefacts are often incomplete, inconsistently documented, or interpreted differently across roles, leading to gaps in understanding and inconsistent assessments.

The dynamic nature of containerised systems further exacerbates these issues. Frequent updates to container images, deployment settings, or host configurations can invalidate previous risk assessments [12, 13]. Without mechanisms for continuous knowledge exchange and automated reasoning, stakeholders struggle to keep pace with changes, resulting in outdated or incomplete evaluations.

3. Proposed Approach: Knowledge-Augmented Risk Identification

We propose a model-based approach for identifying security risks in OT container deployments, using semantic web technologies to formalise domain knowledge. Ontologies represent deployment configurations, system assumptions, and threat scenarios in a structured, interoperable format, enabling consistent and context-aware assessments across roles.

Automated reasoning applies formal semantics to infer threats, validate assumptions, and assess the impact of changes. This integration—referred to as **knowledge augmentation** [14]—enhances expert judgement by making relevant knowledge more accessible, contextualised, and actionable. It supports scalable and collaborative security engineering in dynamic OT environments.

3.1. Industrial Case Study

We applied our approach in a case study on a large-scale industrial platform using Docker Compose for deployment. While operators can inspect Docker images (e.g., via vulnerability scans), doing so for every container is often too costly. As a result, decisions rely mainly on Docker Compose files. We formalised risks associated with these settings to support stakeholder risk identification.

Figure 1 illustrates the approach, structured into three iterative phases: *Model*, *Instantiate*, and *Integrate*. In the case study, we revisited earlier phases to refine the ontology and improve the knowledge graph based on feedback from operators and container developers.

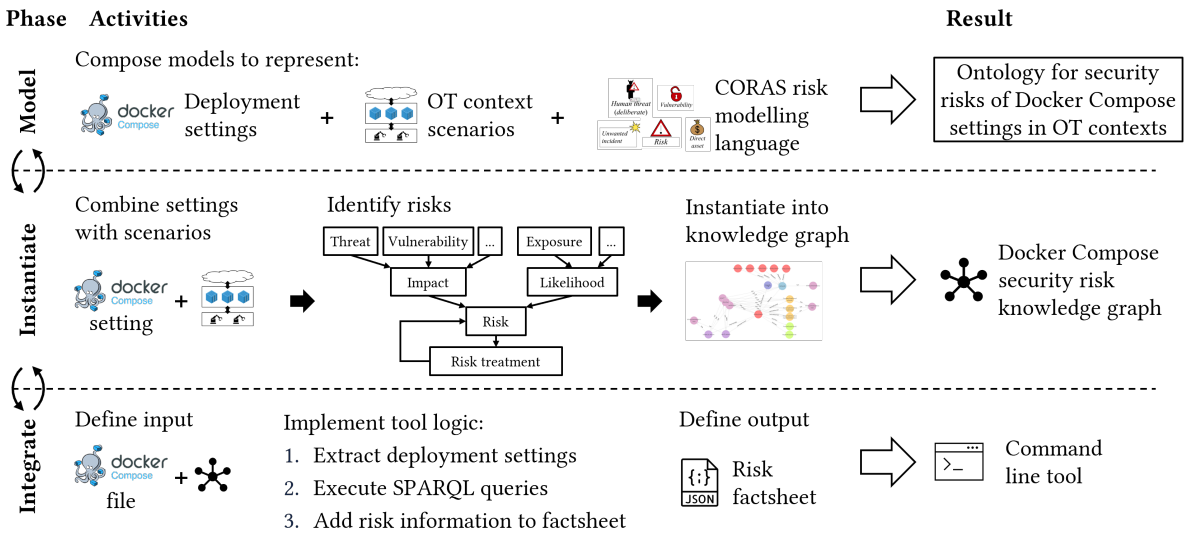

Figure 1: Overview of our approach to knowledge augmentation of security risk identification in the case study

In the **Model** phase, we developed an ontology linking Docker Compose settings, OT context scenarios, and CORAS [15]-based risk modelling elements. This captures how specific combinations—such as `-net=host` with `-cap-add=NET_ADMIN`—can lead to elevated risks. In the **Instantiate** phase, we manually populated the ontology with context scenarios and expert-curated risk elements. This included threats, vulnerabilities, exposures, impacts, likelihoods, and treatments, forming a security risk knowledge graph tailored to the platform. In the **Integrate** phase, we developed a command-line

tool that extracts deployment settings from Docker Compose files, queries the knowledge graph using SPARQL, and generates structured JSON-based risk factsheets.

3.2. Leveraging LLMs for Risk Modelling and Usability

Large language models (LLMs) supported our approach from two perspectives: assisting risk modelling and improving usability. First, LLMs were used during the Instantiate phase to automate aspects of risk modelling. For example, they helped generate threat scenario templates from high-level deployment descriptions. While this improved efficiency, further research is needed to explore how LLMs can identify novel risks, connect attack vectors to impacts, and model complex, chained threat scenarios.

Second, LLMs enhanced the readability and accessibility of the CLI tool's output by translating machine-readable JSON factsheets into human-friendly summaries. These summaries included key risks, treatment measures, and references to hardening guidelines. LLMs also show promise in adapting such resources to user-specific scenarios, such as translating general advice into context-specific recommendations. However, integrating LLMs into modelling introduces challenges: hallucinated risks may reduce trust, and overlooked threats may lead to false negatives. Thus, while LLMs offer valuable support, expert validation remains essential.

4. Discussion

Scalability and Practical Impact The semantic structure of the knowledge base enables scalability through modular updates, versioning, and integration with existing workflows. Users in our case study recognise the potential to automate risk assessment, especially as frequently updated applications from external providers are deployed. The adoption of risk factsheets reduces reliance on manual consulting and supports cross-role collaboration in dynamic OT environments. While initial effort is required to design the ontology and instantiate the knowledge base, the integration of command-line tools and LLMs improves long-term value. By making tool outputs deterministic, traceable, and actionable, the approach ensures that modelling efforts yield lasting benefits through structured knowledge reuse.

Insights and Limitations Our evaluation highlights the potential of the proposed approach, but several limitations must be considered. We employ a qualitative method for risk assessment, meaning the quality of the generated factsheets depends heavily on the accuracy and detail of expert-driven modelling. Greater diligence during modelling improves the effectiveness of the results. In the case study, we primarily analysed Docker Compose files, which provide only partial insight into the applications. As such, the resulting factsheets reflect only the static deployment configuration and should be interpreted with caution. Additional artefacts—such as Dockerfiles, image vulnerability scans, or runtime behaviour—are currently not included but represent promising directions for future work.

5. Conclusion and Future Work

We proposed a knowledge augmentation approach for identifying security risks in OT container deployments, combining semantic modelling, reasoning, and tool support. The method addresses key challenges and improves usability through LLM integration. Future work will focus on evaluating the completeness of identified risks, refining treatment measures, and exploring reliable LLM integration. Including additional artefacts like Dockerfiles and runtime data will further enhance applicability.

Declaration on Generative AI

During the preparation of this work, the authors used Microsoft Copilot in order to: Grammar and spelling check, paraphrase and reword. After using these tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] T. Goldschmidt, S. Hauck-Stattelmann, S. Malakuti, S. Grüner, Container-based architecture for flexible industrial control applications, Journal of Systems Architecture 84 (2018) 28–36. doi:10.1016/j.sysarc.2018.03.002.

[2] L. Arnold, J. Jöhnk, F. Vogt, N. Urbach, IIoT platforms' architectural features – a taxonomy and five prevalent archetypes, Electronic Markets 32 (2022) 927–944. doi:10.1007/s12525-021-00520-0.

[3] M. Souppaya, J. Morello, K. Scarfone, Application Container Security Guide, Technical Report NIST SP 800-190, National Institute of Standards and Technology, Gaithersburg, MD, 2017. doi:10.6028/NIST.SP.800-190.

[4] A. Martin, S. Raponi, T. Combe, R. Di Pietro, Docker ecosystem – Vulnerability Analysis, Computer Communications 122 (2018) 30–43. doi:10.1016/j.comcom.2018.03.011.

[5] Y. Landeck, D. Balta, M. Wimmer, C. Knierim, Software in the Manufacturing Industry: Emerging Security Challenge Areas for IIoT Platforms, in: 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '24), 2024. doi:10.1145/3639477.3639724.

[6] Y. Landeck, D. Balta, M. Wimmer, C. Knierim, Assurance of Application Security on IIoT Platforms with Knowledge Augmentation, in: 2024 Annual Computer Security Applications Conference Workshops (ACSAC Workshops), IEEE, Honolulu, HI, USA, 2024, pp. 108–119. doi:10.1109/ACSACW65225.2024.00019.

[7] K. Tange, M. De Donno, X. Fafoutis, N. Dragoni, A systematic survey of industrial Internet of Things security: Requirements and fog computing opportunities, IEEE Communications Surveys & Tutorials 22 (2020) 2489–2520.

[8] J. Prinsloo, S. Sinha, B. Von Solms, A Review of Industry 4.0 Manufacturing Process Security Risks, Applied Sciences 9 (2019) 5105. doi:10.3390/app9235105.

[9] Y. Landeck, D. Balta, M. Wimmer, C. Knierim, Software in the Manufacturing Industry: A Review of Security Challenges and Implications, in: 18th International Conference on Wirtschaftsinformatik, 2023.

[10] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, L. Safina, Microservices: Yesterday, today, and tomorrow, Present and ulterior software engineering (2017) 195–216.

[11] F. Bolici, J. Howison, K. Crowston, Stigmergic coordination in FLOSS development teams: Integrating explicit and implicit mechanisms, Cognitive Systems Research 38 (2016) 14–22. doi:10.1016/j.cogsys.2015.12.003.

[12] A. Y. Wong, E. G. Chekole, M. Ochoa, J. Zhou, On the Security of Containers: Threat Modeling, Attack Analysis, and Mitigation Strategies, Computers & Security 128 (2023) 103140. doi:10.1016/j.cose.2023.103140.

[13] A. Mills, J. White, P. Legg, Longitudinal risk-based security assessment of docker software container images, Computers & Security 135 (2023) 103478. doi:10.1016/j.cose.2023.103478.

[14] J. P. Delgrande, B. Glimm, T. Meyer, M. Truszczynski, F. Wolter, Current and Future Challenges in Knowledge Representation and Reasoning, 2023. arXiv:2308.04161.

[15] M. S. Lund, B. Solhaug, K. Stølen, Model-Driven Risk Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-12323-8.