

Semantic integration of Security Requirements with Topologies Specification for RL-Based Orchestration

Supervisors

Dhouha Ayed (THALES)

Louis Cailliot (THALES)

Contact : Dhouha.ayed@thalesgroup.com, 0674903468

Context

Thales integrates and provides a wide range of networked services across the edge–cloud continuum. These services increasingly rely on virtualization, cloudification, and Software-Defined Networking (SDN) to deliver flexible and scalable networking and computing capabilities. While this evolution enables advanced orchestration and resource efficiency, it also raises new security challenges: dynamically orchestrated service graphs may span multiple administrative domains, execution environments, and trust levels, making it difficult to guarantee security properties consistently—especially when deployments must adapt to changing conditions or security threats.

In this context, the ability for systems to self-adapt against security threats is a major step toward more robust and scalable solutions. Rather than relying solely on manual incident response, Thales aims to develop AI-based mechanisms that can initiate security protection and mitigation automatically. Such AI capabilities are intended to reduce operational dependency on human expertise while improving response speed and consistency.

The objective of Thales is therefore to tackle this challenge by developing Reinforcement Learning (RL) solutions that support decision-making during orchestration.

Problem

On the one hand, security requirements are often expressed in a format that is not directly usable by RL: they describe security policies and constraints, but they do not naturally provide a representation in terms of numerical variables or decision signals. On the other hand, topologies are described as an application structure (nodes, relationships, dependencies) that must be translated into a placement decision space. Without semantic formalization, this translation becomes ad hoc, brittle, and poorly traceable: the RL agent then learns from arbitrary features which degrades both the stability and the efficiency of learning.

Objective

Build one (or more) ontologies acting as a semantic layer for a formal, reusable, and traceable representation of security requirements and topology to be orchestrated in order to automatically produce “RL-ready” inputs.

References

1. P. Zhang, C. Wang, C. Jiang and A. Benslimane, “Security-Aware Virtual Network Embedding Algorithm Based on Reinforcement Learning,” IEEE Transactions on Network Science and Engineering, vol. 8, pp. 1095-1105, 2021.

2. Casola, V., De Benedictis, A., Rak, M., & Villano, U. (2016). SLA-based Secure Cloud Application Development. SCPE, 17(4). <https://doi.org/10.12694/scpe.v17i4.1200>
3. Geoffrey Chollon, Dhouha Ayed, Rodrigo Asensio Garriga, Alejandro Molina Zarca, Antonio Skarmeta, Maria Christopoulou, Wissem Soussi, Gürkan Gür, and Uwe Herzog, “ETSI ZSM driven security management in future networks,” in 2022 IEEE Future Networks World Forum (FNWF), pp. 334–339, Oct. 2022.
4. Bagaa, M., Taleb, T., Bernal Bernabe, J., & Skarmeta, A. (2022). QoS and Resource-aware Security Orchestration and Life Cycle Management. IEEE Transactions on Mobile Computing, 21(8), 2978 - 2993. <https://doi.org/10.1109/TMC.2020.3046968>
5. Ortiz, J., Sanchez-Iborra, R., Bernabe, J. B., Skarmeta, ..., Ayed, D., ... Lopez, D. (2020). INSPIRE-5Gplus: Intelligent security and pervasive trust for 5G and beyond networks. In Proceedings of the 15th International Conference on Availability, Reliability and Security, ARES 2020 ACM. <https://doi.org/10.1145/3407023.3409219>
6. B. Hnatkowska and M. Kimmel, Data Generation Based on Domain Ontology, 2023
7. N. F. Noy and D. L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology”, Stanford University. [ontology101.pdf](#)
8. W3C, “RDF 1.1 Concepts and Abstract Syntax”, W3C Recommendation.
9. W3C, “RDF 1.1 Semantics”, W3C Recommendation.
10. W3C, “SPARQL 1.1 Query Language”, W3C Recommendation.
11. OASIS, “TOSCA Simple Profile in YAML Version 1.3”, OASIS Standard.
12. OASIS, “TOSCA Simple Profile in YAML Version 2.0”, OASIS Standard.
13. OASIS, “TOSCA Concepts”, OASIS Standard (modèle conceptuel).

More details about technical context

In particular, we focus on placement decisions for services described using TOSCA (Topology and Orchestration Specification for Cloud Applications). In a typical orchestration workflow, a TOSCA topology represents application components (nodes), their relationships, and the way they must be connected and deployed. However, choosing where each TOSCA component is placed (e.g., which host, cluster, zone, or edge site) strongly impacts security outcomes such as isolation, confidentiality protection, compliance, and overall risk exposure.

To capture security needs in a structured way, the orchestration process must consider security requirements specified in a SSLA (Security Service Level Agreement). These SSLA properties define constraints and expectations that the deployment must satisfy.