Secure and Flexible Deployment of Industrial Applications inside Cloud-Based Environments

1st Marco Ehrlich
rt-solutions.de GmbH
Cologne, Germany
ehrlich@rt-solutions.de

2nd Henning Trsek
rt-solutions.de GmbH
Cologne, Germany
trsek@rt-solutions.de

3rd Martin Gergeleit
Hochschule RheinMain
Wiesbaden, Germany
martin.gergeleit@hs-rm.de

4th Julius Paffrath
rt-solutions.de GmbH
Cologne, Germany
paffrath@rt-solutions.de

5th Kostyantyn Simkin
Hochschule RheinMain
Wiesbaden, Germany
kostyantyn.simkin@student.hs-rm.de

6th Jürgen Jasperneite
Fraunhofer IOSB-INA
Lemgo, Germany
juergen.jasperneite@iosb-ina.fraunhofer.de

Abstract—**Future industrial production systems following the paradigms of Industrie 4.0 will be reconfigured frequently and new system configurations will be deployed automatically as part of the engineering processes. In order to keep pace with this requirement of increased flexibility, it will be needed to achieve the adequate security levels in an automated way and to reduce the current static procedures and manual efforts as much as possible. Therefore, a modelling of all security-related functionalities and capabilities is mandatory. This paper further describes an approach for such a modelling based on the IEC 62443 security standard and an implementation of an automated deployment of components inside an industrial environment established with the OASIS Tosca and Ansible tools. The first results of the whole tool chain are evaluated with a real-world use case of software deployment in a suitable lab environment.**

Index Terms—**Industry, Cyber Security, Cloud, Deployment, IEC 62443, OASIS TOSCA, Ansible**

I. Introduction

The emerging Industrial Internet offers a great potential for innovations in the area of industrial automation. Future intelligent networks and innovative services are the basis to be able to link virtual and physical processes as a fundamental concept of Industry 4.0 [1]. Techniques from the Information Technology (IT) domain like (edge) cloud computing and Software-Defined Networking (SDN) are becoming increasingly important also during the digitalisation of the Industrial Automation and Control Systems (IACS).

For the definition of the involved logical components, such as computing nodes, network links, operating systems, and services, a number of concepts exist, e.g. the Topology and Orchestration Specification for Cloud Applications (TOSCA) proposed by the Organization for the Advancement of Structured Information Standards (OASIS) for cloud deployment and orchestration [2]. However, these specifications do not explicitly address the mandatory topic of security, i.e. the specified system configuration has to be checked manually by experts whether it fulfills the demanded security requirements of the given production environment. Especially the increased flexibility of systems results in high efforts to meet security requirements at every point of time. Furthermore, the requested dynamics of future industrial systems will allow various new attack vectors causing a predicted global loss of $6 trillion by 2021 [3], which makes security even more important also inside the industrial automation domain [4], [5].

To reduce the needed resources it will be required to achieve the adequate security levels in an automated way and to reduce the existing manual efforts as much as possible [6]. Therefore, the modelling of requirements and capabilities for all security related aspects is needed as well as an automated deployment of the checked systems. In [7] and [8] the approach of how to enhance a TOSCA specification in order to become security aware and the implementation of checking mechanisms have been presented. In this paper, the approach will be further extended by introducing new tool chain components for the automatic deployment of a container based solution.

The remainder of the paper is organized as follows: First the relevant background is introduced in terms of the concepts of the IEC 62443 security standard, the approach of security modelling with TOSCA, and the automated Ansible playbooks. After this, the modelling framework as well as the tool chain and its implementation is described. Finally, the automated tooling process of the approach is evaluated using a simplified edge cloud use case to obtain and present first results.

II. Essential Background

This section will introduce the fundamentals of the modelling approach and the tool chain for automated deployment.

A. *IEC 62443 Security Standard*

The IEC 62443 security standard was established for the usage inside Industrial Automation and Control Systems (IACSs) and was developed into the most important one covering the fundamental issues of industrial communication networks and their intrinsically linked topic of security. The standard classifies information security in a four stage scaling system in

978-1-7281-0303-7/19/$31.00 ©2019 IEEE

which each stage is stated as a **Security Level** (SL). The SLs are designed with a focus on the attacker's motivation, skills, and resources. SL 0 means no security requirements at all and SL 4 is described by protection against intentional violation using sophisticated means with extended resources. The various security goals available are expressed in seven **Foundational Requirements** (FRs) [9], covering the commonly identified dimensions in security, which are shown in Table I.

TABLE I
FOUNDATIONAL REQUIREMENTS (FRS) OF THE 62443

FR No.	Topic
FR 1	Identification and Authentication Control (IAC)
FR 2	Use Control (UC)
FR 3	System Integrity (SI)
FR 4	Data Confidentiality (DC)
FR 5	Restricted Data Flow (RDF)
FR 6	Timely Response to Events (TRE)
FR 7	Resource Availability (RA)

In addition, the IEC 62443 standard defines **System Requirements** (SRs) for each FR, which represent specific technical requirements that have to be fulfilled to achieve a certain SL. In general, there is a differentiation between three characteristics of SLs [9]. The result of the initial risk analysis is the definition of the SL-T vector for a planned system. The system integrator configures an automation solution based on available components, which inherit their own SL-Cs. It is tried to achieve the demanded SL-T. If the accumulated SL-A protection level cannot meet the requirements from the SL-T vector, the operator must accept the remaining risks or compensate through further measures within his area of responsibility [10].

- **Target Security Level (SL-T):** Desired level of security for a particular system during conception phase
- **Achieved Security Level (SL-A):** Actual level of security for a particular system after finished setup
- **Capability Security Level (SL-C):** Security level that the chosen components in a setup can provide

B. Security Modelling with TOSCA

The Topology and Orchestration Specification for Cloud Applications (TOSCA) is an OASIS standard language that has been developed to simplify the definition and deployment of services in cloud environments [2], [11]. It allows to describe the topology of cloud based services, their hardware and software components, and the processes that manage them. TOSCA uses an object-oriented approach to model "topologies" consisting of "nodes", such as computing nodes, networks, or software services, their attributes, their relationships e.g. "runs on" or "linked to", capabilities, and requirements [2]. As IACSs are also moving towards cloud infrastructures, TOSCA is a candidate for modelling these kinds of automation systems. The classic way of defining TOSCA models is to write a declaration in a YAML language dialect given by the OASIS standard. While not initially targeted towards IACS, TOSCA is generic enough to describe any kind of topologies and the language is also open for the definition of new types, either derived from the standard types or in a separate type hierarchy. Therefore, the "puccini" compiler is used to enhance the TOSCA framework with our concepts [12]. In [7] and [8] we already proposed an approach of modelling industrial applications including their security capabilities and requirements, which uses the FR and SL abstractions of the IEC 62443 standard. The TOSCA language is used to specify the supported SL-C vectors of the components and the SL-T vector of the given system.

C. Ansible Playbooks

Ansible is an open source IT configuration management, cloud provisioning, ad-hoc task execution, network automation, and multi-node orchestration tool with a focus on consistency, security, and reliability [13], [14]. The software manages components, such as computers, network nodes, or cloud storages, via OpenSSH (or WinRM on Windows) remotely in order to increase productivity by implementing automated solutions e.g. operating system configuration or in our case container deployment [15]. This is done by means of so-called playbooks defined in YAML, which describe host configurations (from a statically defined host in the 'inventory' or discovered during runtime inside the 'dynamic inventory') and an ordered list of tasks to perform on those hosts. A task is a call to an Ansible module performing a specific task remotely.

III. IMPLEMENTATION

This section will describe the further evolution of the modelling approach which is extended by an optional mask vector and the tool chain is enhanced with a backend to generate Ansible playbooks for an automated deployment of topologies.

A. Mask Vector

When modelling various application topologies with the extended TOSCA language it turned out that often it was quite difficult to assign a sensible SL-C to some dimensions of certain components. While e.g. FR 5 (Restricted Data Flow) is an obvious and necessary capability of a network devices like a gateway, it doesn't make much sense for a pure software component that has only one service interface like a data base. How to handle this during the modelling process? Choosing SL 0 would be an obvious option as this component doesn't provide any features to control data flows in a topology. However, this means that during security evaluation the tool will throw a warning for this component for any required level greater than SL 0, as it obviously doesn't support it. This is probably not a desired behaviour, as in this case the user has to ignore this warning every time. Even worse, a user has to distinguish between 'real' security issues and those that are implied by irrelevant capabilities resulting in additional efforts. On the other hand defining this value to the highest level SL 4 will surely avoid any warning, but the modelling isn't correct and self-explanatory any more. SL 4 in such a case would be more a work-around than a valid security classification.

What is needed is an indication that this security dimension of this component isn't relevant for the overall analysis. An additional value of ‘don’t care’ (DC) can serve this purpose. It indicates the checker tool to ignore this dimension during the security evaluation process. It might be convenient during modelling to set all values simply to DC that are not immediately obvious or that cause warnings. However, the tool requires a mandatory responsible author entry and an explaining comment for all ‘Don’t care’ entries. These will be copied to the final security analysis log and it documents the decisions of the systems designer. This encourages not to abuse the DC mechanism for a too sloppy evaluation.

In our implementation we don’t use a DC value directly, but we define an optional mask vector that marks all DC dimensions of a component as ‘false’. This allows to assign correctly an SL 0 as IEC 62443 security level but it gives the directive to the checker to ignore this value during security evaluation. An example of a “SecureDatabase” type is shown in Listing 1. Here FR 5 in the security vector is set to SL 0 while it is marked as irrelevant for evaluation with *false* in the mask vector. The mandatory entries for author and comment explain who has made this decision and why.

Listing 1. Definition of a data base with a mask vector

```
ProductionDataDB:
  type: SecureDatabase
  properties:
    name: private_cloud_server_db
    sec_vector: [1,2,3,2,0,3,3]
    sec_mask:
      mask: [true, true, true, true,
             false, true, true]
      author: 'Kostyantyn Simkin'
      comment: 'No impact on data flow'
  requirements:
    - belongs:
        node: it_environment
    - host:
        node: private_cloud_server
```

B. Tool Chain Implementation

Figure 1 depicts the workflow with the current tool chain to check and deploy a modelled application. At first the model is described as a TOSCA topology using the security extensions presented in Section II-B. The “puccini-compile” tool includes all standard declarations, checks syntactical and semantic correctness, and generates an intermediate “.clout” file. This file (again in YAML format) contains the complete expanded structure of the model with all nodes and all attributes. The “security-eval” tool uses this information to check for all components (compute, network, and service nodes), whether their SL-C vectors are greater or equal than the maximum of the required SL-T values of the security domains in all dimensions (not marked as irrelevant in the component’s mask vector).

The result of this analysis, including the comments for the mask vectors are written to a “Security.log” file, that documents possible mismatches between the security requirements and the capabilities of the components. The “puccini-js” tool also uses the intermediate “.clout” file as input for generating the deployment instructions. This tool is a variable backend that can be customized to generate various output formats. We modified it to write Ansible “playbooks”, scripts that can control the complete life cycle of a distributed application, at least starting and stopping all components. In our current implementation we finally use Docker containers and SSH access to deploy software components in an edge cloud environment.

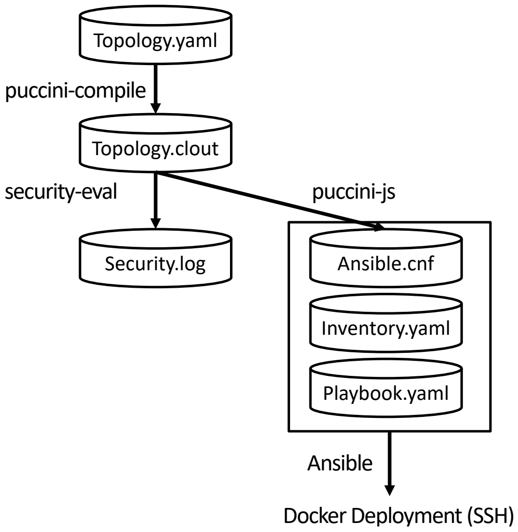

Fig. 1. Workflow for checking and deploying a topology

IV. Evaluation

In [8] the use case analysis and elaboration of common requirements regarding edge cloud computing was conducted, which is used for the evaluation of the proposed approach. For the purpose of this work in progress paper we have chosen a simplified scenario with a defined set of components, where the reader can verify the constraints with limited effort. In addition, the corresponding security modelling is still in development. The more components and domains are involved the more constraints have to be checked each time after updating any part of the system configuration. Thus, an integrated and automated checking system can exploit its strengths even more when the architecture becomes larger, especially as the inherent complexity of the checking problem itself scales linearly.

Figure 2 shows the setup of the demonstrator used for evaluation of the prototypical implementation. The sample scenario contains an industrial network with light and inductive sensors supporting a conveyor belt and a barrier actuator to sort small plastic puck-shaped objects representing produced goods. This network has increased requirements regarding real-time capabilities, timing, and determinism of events. Therefore, it is supported with a locally connected edge cloud computer for lower latency, higher performance, and availability of data. Various applications, such as a soft Programmable Logic Controller (PLC) based on OpenPLC [16] or a local database for sensor data, can be deployed via the container-based approach described in Section III.

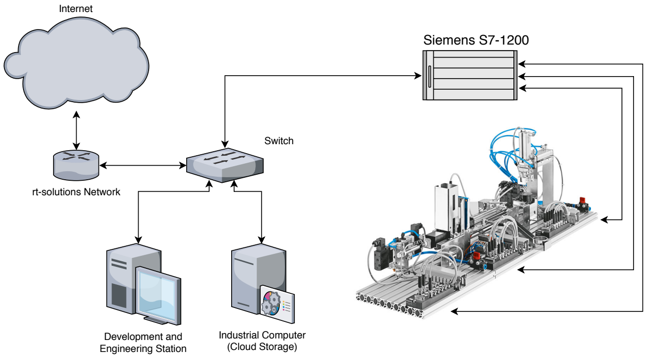
Internet
Siemens S7-1200
Switch
rt-solutions Network
Development and Engineering Station
Industrial Computer (Cloud Storage)

Fig. 2. Demonstrator setup with prototypical implementation

The current status of the implementation allows a flexible deployment of edge cloud resources via containers in a secure manner. The used IEC 62443 security standard describes the modelling of security-related information and the described tools (puccini, Ansible, and Docker) enable the operator to use the given industrial system in a dynamic and secure way. The reconfiguration and adaptation of software components proposed by the Industrie 4.0 developments can be done securely. This is achieved at present during the compilation when the check against the SL-Ts of the application's security are performed, i.e., it is checked for all components (compute, network, and service nodes), whether their SL-C vectors are greater or equal in all dimensions than the maximum of the required SL-T values. A more detailed example of the described checking procedure is available in [7]. The check reveals and the TOSCA compiler warns if there is a mismatch between the requirements and the capabilities of a component. In such a case either additional security controls have to be implemented, e.g. enabling general frame encryption on this network. Alternatively it can be stated that the potential risk is acceptable. For instance, if the physical protection of the environment is strong enough to ensure the general security objectives.

V. Conclusion

Flexible industrial production systems belong to the core concepts of Industrie 4.0 and require a fast and automated deployment of logically defined, virtualized, and networked architectures as one important factor. In this context, security aspects are of utmost importance. However, security is usually handled in a very static and manual way, which contradicts to the needed flexibility and leads to the demand for more dynamic approaches for an establishment of adequate security levels inside the industrial environments.

Therefore, our approach is to extend the existing TOSCA simple profile in YAML by a modelling of security capabilities and requirements. They are mapped to the well-established FRs and SLs of IEC 62443. After our modelling approach is briefly introduced, the implemented tool chain is described. It consists of a puccini compiler to check correctness and to generate the expanded model, a security evaluation tool, and finally the generation of Ansible playbooks for an automated deployment of the whole system. The first evaluation of the complete tool chain is performed based on a simplified scenario. It evidently shows the suitability of our approach for an automated deployment of secure industrial architectures.

In our future work, the prototype will be further enhanced by optimizing the checking algorithm, which is responsible for the decisions regarding deployment. Additional evaluation rules, such as relations between components, topology-related decisions, or different approaches e.g. fuzzy logic, will be assessed to improve the credibility of the approach. In [7] e.g. we have already started modelling a more complex use case scenario from the IC4F project [17] in order to check the feasibility also with bigger architectures.

Acknowledgments

This work was funded by the German Federal Ministry for Economic Affairs and Energy with the research project "Industrial Communication for Factories" (IC4F) and the German Federal Ministry of Education and Research with the research project "Future Industrial Network Architecture" (FIND).

References

[1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, *The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0*, IEEE Industrial Electronics Magazine, 2017.

[2] Organization for the Advancement of Structured Information Standards, "OASIS TOSCA Simple Profile in YAML Version 1.1," 2018, Online: http://docs.oasis-open.org/tosca/TOSCA/v1.0/TOSCA-v1.0.pdf.

[3] D. R. Coats, *Worldwide Threat Assessment of the US Intelligence Community*, Statement for the Record, 2018.

[4] A. Pattanayak and M. Kirkland, *Current Cyber Security Challenges in ICS*, IEEE International Conference on Industrial Internet (ICII), Seattle, USA, 2018.

[5] S. Obermeier, *Cyber Security Research Challenges - An Industry Perspective*, 23rd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Torino, Italy, 2018.

[6] M. Ehrlich, L. Wisniewski, H. Trsek, D. Mahrenholz, and J. Jasperneite, "Automatic mapping of cyber security requirements to support network slicing in software-defined networks," 2017, 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus.

[7] M. Gergeleit, H. Trsek, T. Eisert, and M. Ehrlich, *Modeling Security Requirements and Controls for an Automated Deployment of Industrial IT Systems*, 9. Jahreskolloquium Kommunikation in der Automation (KommA), Lemgo, Germany, 2018.

[8] M. Ehrlich, M. Gergeleit, K. Simkin, and H. Trsek, *Automated Processing of Security Requirements and Controls for a common Industrie 4.0 Use Case*, International Conference on Networked Systems Workshop - Advanced Communication Networks for Industrial Applications, Garching, Germany, 2019.

[9] International Electrotechnical Comission, "Automatic mapping of cyber security requirements to support network slicing in software-defined networks," 2015, IEC 62443-3-3: System security requirements and security levels.

[10] P. Kobes, *Guideline Industrial Security - IEC 62443 is easy*, VDE Verlag, Offenbach, Berlin, 2017.

[11] T. Binz, G. Breiter, F. Leyman, and T. Spatzier, "Portable Cloud Services Using TOSCA," 2012, IEEE Internet Computing, Volume: 16, Issue:3.

[12] GitHub. (2019). Puccini Compiler, [Online]. Available: https://github.com/tliron/puccini (visited on 07/08/2019).

[13] RedHat. (2019). Ansible, [Online]. Available: https://www.ansible.com/ (visited on 07/08/2019).

[14] J. Keating and J. Feeman, *Mastering Ansible*, Third Edition, 2019.

[15] D. Hall, *Ansible Configuration Management*. Packt Publishing, 2013.

[16] T. Alves. (2019). OpenPLC, [Online]. Available: https://www.openplcproject.com/ (visited on 07/08/2019).

[17] IC4F Project. (2019). Industrial Communication for Factories, [Online]. Available: https://www.ic4f.de/ (visited on 07/08/2019).