Modeling Security Requirements and Controls for an Automated Deployment of Industrial IT Systems

Martin Gergeleit¹, Henning Trsek², Till Eisert², Marco Ehrlich³

¹Hochschule RheinMain
65195 Wiesbaden
martin.gergeleit@hs-rm.de

²rt-solutions.de GmbH
50968 Köln,
trsek@rt-solutions.de, eisert@rt-solutions.de

³inIT - Institute Industrial IT
Technische Hochschule Ostwestfalen-Lippe
32657 Lemgo
marco.ehrlich@th-owl.de

Abstract. Due to the dynamic nature of the Industrial Internet and Industry 4.0, future production systems will be reconfigured frequently and as a part of the engineering process, new system configurations will be deployed automatically. In order to keep pace with this development, it will be required to achieve the needed security level in an automated way and to reduce the current static procedures and manual efforts as much as possible. Therefore, the development and modeling of requirements and capability profiles for all cyber security related aspects is needed. The paper describes an approach for such a modeling based on security requirements and levels of the international standard IEC-62443-3-3 and a system description based on OASIS TOSCA. The approach is applied to a real industrial use-case scenario and an evaluation is performed to demonstrate its feasibility.

1 Introduction

The emerging Industrial Internet offers a great potential in assuring and extending Germany's position as a powerful location for production technology and innovations in the area of industrial automation. Future intelligent networks and innovative services, which are based on the Industrial Internet, are the basis to be able to link virtual and real processes as a fundamental concept of Industry 4.0. Techniques from business IT like (Edge) Cloud Computing and Software-Defined Networking (SDN) are becoming increasingly important in Industrial Automation and Control Systems (IACS). Their introduction allows a fast and automated deployment of logically defined, virtualized architectures

onto physical hardware, which is an important factor for the promised adaptable manufacturing in Industry 4.0.

For the definition of these logical structures like computing nodes, network links, operating systems, and services, as well as their configuration parameters a number of concepts exist, like e.g. OpenFlow [MAB⁺08], NETCONF [ESB11] for SDNs or OASIS TOSCA [OAS13] for cloud orchestration. However, these specifications do not explicitly address another important factor for the success of I4.0 systems: Security. Currently each specified system configuration has to be checked manually whether it fulfills the specific security requirements of the current production environment. This imposes a high effort that leads to a trade-off between the dynamics of the adaption of production processes and their security. To avoid this in the future it will be required to achieve the needed security level in an automated way and to reduce the current static procedures and manual efforts as much as possible [EWT⁺17]. Therefore, the development and modelling of requirements and capability profiles for all cyber security related aspects is needed.

The international standard series IEC 62443 defines procedures for implementing secure IACS. It specifies four security levels (SLs) that are related to the skills, the resources, and the motivation of an adversary, ranging from an accidental intrusion up to a secret service style attack. Especially IEC 62443 part 3-3 [IEC15] defines the technical requirements that have to be met by a complete system to conform to one of the security levels. It distinguishes seven foundational requirements (FRs) covering the most important security objectives, each of them having detailed technical system requirements (SRs). For any specific IACS a risk analysis has to be performed to identify the required SLs for these different aspects of security, as not all of them necessarily have to fulfill the same level. Thus, the resulting security requirements can be summarized in a vector of these seven SLs.

The methodology of the IEC 62443 suggests that a supplier of an industrial IT component specifies which FRs/SRs at which SL are covered by the security controls of the component. Often a component has the ability to fulfill requirements at different SLs, even the higher ones, but there is a trade-off in terms of performance, resource consumption, or often easy-of-use and productivity. When a system integrator builds a system architecture out of these components, the integrator has to select and parametrize the components. Furthermore, the sub-systems (zones) and connections (conduits) must be analyzed to assess whether they meet as a whole the FRs/SRs of the desired SL. However, with virtualization techniques and automated deployment of components in a highly flexible production environment this process gets very dynamic. Manual validation of conformance with security requirements will either become a time-consuming bottleneck or tends to be sloppy and introduces new security risks.

A formal description of the security requirements of the target environment in terms of the IEC 62443 abstractions and an according description of the security controls of the deployed components will be beneficial here. This allows at least for basic sanity checks and can give hints on mismatches of required

functionality and the defined system configuration. Moreover, it allows to select the appropriate security configuration parameters of components for deployment. Finally, this model can be used to define rules for the composition of components and a validation of a complete system.

Therefore, the approach of this paper is to propose a formal notation that models the security requirements of a system as well as the security controls and parameters of its components – similar to [Tun17] – in accordance to the well-accepted international standard IEC 62443. This model, based on YAML, a simple markup-language, can then be used to automatically check and/or parameterize the deployment specifications. The approach is applied to a real industrial use-case scenario and an evaluation is performed.

The reminder of the paper is organized as follows: First the specification of IEC 62443 security parameters is explained and an introduction to TOSCA is given. Then the approach for checking such models against given security requirements is explained. Finally, an example use-case scenario of a non-trivial IACS system is described including the security parameter modeling and the approach is evaluated.

2 IEC 62443 Security Standard

In general, several approaches from various domains, such as telecommunication, multimedia, home computing, industrial automation, or research, are available for security modelling nowadays [EWT⁺17,EWT⁺18]. Nevertheless, the furthest developed standard is the IEC 62443 by the International Society of Automation (ISA), which is originally suited for the Industrial Automation and Control Systems (IACSs) inside the industrial automation and manufacturing domain and is enriched by ideas and concepts from various other standards, guidelines, and numerous worldwide distributed organizations. This standard gained a lot more attention during the past years and was developed into the most important one for the industrial domain covering the topics of industrial communication networks and system cyber security [Rol13].

During various editing and improvement phases the current status offers a structure including all aspects of industrial cyber security today. The standard describes possible threats or attacks regarding cyber security in a four stage scaling system in which each stage is stated as a Security Level (SL). **SL 1** delivers protection against casual or coincidental violation, **SL 2** provides protection against intentional violation using simple means, **SL 3** gives protection against intentional violation using sophisticated means, and finally **SL 4** is described by protection against intentional violation using sophisticated means with extended resources. The SLs are designed in the way of using the attacker’s motivation and resources, which is seen as a future-proof definition. The standard further describes so-called Foundational Requirements (FRs) based on multiple System Requirements (SRs) with varying quantity on each FR, which provide an abstracted view on the overall security goals [IEC15]. For the current state of our approach we will focus on the utilization of the seven defined FRs:

- **FR 1:** Identification and Authentication Control (IAC) - Identify and authenticate all users (humans, software processes, and devices) before allowing them to access the control system.
- **FR 2:** Use Control (UC) - Enforce the assigned privileges of an authenticated user (human, software process, or device) to perform the actions on the IACSs and monitor the use of these privileges.
- **FR 3:** System Integrity (SI) - Ensure the integrity of the IACSs to prevent unauthorized data or information manipulation.
- **FR 4:** Data Confidentiality (DC) - Ensure the confidentiality of information on communication channels and in data repositories to prevent unauthorised disclosure.
- **FR 5:** Restricted Data Flow (RDF) - Segment the control system via zones and conduits to limit the unnecessary flow of data.
- **FR 6:** Timely Response to Events (TRE) - Respond to security violations by notifying the proper authority, reporting needed evidence of the violation, and taking timely corrective action when incidents are discovered.
- **FR 7:** Resource Availability (RA) - Ensure the availability of the control system against the degradation or denial of essential services.

The desired SLs regarding the different FRs can be varying and they are dependent on the specific use case, which is described using the IEC 62443 standard. Generally there is a differentiation between three characteristics of SL [IEC15]:

- **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

The basis of the given procedure is always a risk analysis. The goal is to identify risks and their impact based on a segmentation of the system into cells (zones) and communication channels (conduits). The subdivision of the network is very useful to limit possible damages to a certain cell. The protection objectives of the various cells can be quite different. The result of this exercise is an architecture divided into zones and conduits and the definition of the SL-T vector for each of these units. In response to the above, the system integrator or the plant engineer configures an automation solution based on available components or systems, which inherit their own single SL-C. It is tried to achieve the SL-T of the zones and conduits as far as possible. If this is not sufficient, it takes additional measures, so called compensating countermeasures, which increase the protection level. 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.

3 Security extension for OASIS TOSCA

TOSCA (Topology and Orchestration Specification for Cloud Applications) [BBF⁺12,OAS18] is an OASIS standard language that has been developed to simplify the definition and deployment of services in a cloud environment. It allows to describe the topology of cloud based services, their hard- and software components, and the processes that manages them. TOSCA uses an object-oriented approach to model “topologies” consisting of “nodes” (all kind of components: computing nodes, networks, and also software services), their attributes, their relationships (like “runs on” or “linked to”), their capabilities, and requirements. The TOSCA standard also contains a set of basic types that are usually used to compose cloud services [OAS18]. The classic way of defining TOSCA models is to write a declaration in a YAML language dialect given by the OASIS standard. However, in the last years there also have been developed graphical browsers, which allow for visual and interactive creation of such topologies.

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. TOSCA, in contrast to other modeling languages (like e.g. UML), has the advantage, that it has been developed with explicitly cloud computing in mind. Thus, there are also tools that can automatically deploy and maintain TOSCA models in a cloud environment. As IACS are also moving towards cloud infrastructures, even if it is “only” a local edge cloud, it becomes obvious, that TOSCA is also a candidate for modeling these kind of automation systems.

However, up to now TOSCA has no means to handle security issues explicitly. It is the approach of this paper to enhance TOSCA, its definition language as well as its compiler with its static semantic checker, to become aware of security. Thus, a system integrator can select the system components, describe the system’s topology, and declare its security requirements. The TOSCA compiler will be able to check, whether the system (as a whole or parts of it) can match these requirements. If not, it can identify those components that violate the requirements and it can give an indication about which dimensions of security these components would need an enhancement. This is the basis for later on empowering the deployment and management engine to automatically select the correct components and the proper configurations for a system.

Obviously, a good starting point for introducing security considerations into TOSCA when focusing on IACS are the abstractions of the IEC 62443 as described in the previous section. TOSCA already has the build-in type of a capability. A capability is a typed attribute that describes a feature of a node, e.g. its RAM size or its number of CPUs. Why not using these attributes also for the describing and quantifying the security features of a component? Listing 1 shows the declaration of a new property type, named IEC62443_FRs, as a vector of 7 named integer values ranging from 0 to 4 corresponding to the 7 FRs of the IEC 62443. The possible values 1 to 4 represent the four different SLs. A 0-value, as a special case, means “not applicable here”, when for some reason this FR as a dimension of security cannot be attributed to this component.

Listing 1. Declaration of FRs/SLs as capability type.

```

capability_types:
  toska.capabilities.IEC62443_FRs:
    derived_from: toska.capabilities.Root
    properties:
      FR1_IAC:
        type: integer
        default: 0
        constraints:
          - {in_range: [ 0, 4 ]}
# details of other FRs omitted for brevity
      FR2_UC:
      FR3_SI:
      FR4_DC:
      FR5_RDF:
      FR6_TRE:
      FR7_RA:

```

This capability type can now be used to extend the existing TOSCA types using object-oriented inheritance by simply deriving a “Secure” type from a basic type. As such a “Secure” type has all the properties and capabilities of the basic type, it can replace it in any topology without any harm to the standard TOSCA semantics. Additionally, it contains at least an instance of the IEC62443_FRs capability and optionally some more properties that describe the security controls provided by this component and their parameters. Listing 2 shows as an example the declaration of a “SecureNetwork” type, derived from the standard “Network”. In this case it contains additional properties describing the provided network technology (e.g. plain IP or TSL), the used authentication mechanism (e.g. certificates or username/password), and the used encryption algorithm (e.g. AES-CCMP or 128-EEA2-AES). Most importantly, it also contains a vector of the IEC62443_FRs capability to describe the SLs provided by this component.

Listing 2. Declaration of the SecureNetwork type.

```

# SecureNetwork
toska.nodes.network.SecureNetwork:
  derived_from: toska.nodes.network.Network
  properties:
    technology:
      type: string
    authentication:
      type: string
    encryption:
      type: string
  capabilities:
    IEC62443_FunctionalRequirements:
      type: toska.capabilities.IEC62443_FRs

```

As shown later in Section 5 the actual components are then classified according to the strength of their security controls. Based on a security evaluation the actual values of these additional security properties and SLs are attributed to components. These “Secure” types are then used to build topologies.

4 Model checking

Once an application is defined as a topology consisting only of “Secure” types with actual nodes including their IEC62443_FRs capabilities, the TOSCA compiler can process it. A standard TOSCA compiler would just check whether all functional requirements can be met and it would then create a database containing all the required information for deploying this topology on a cloud infrastructure. This still works, but a security-aware TOSCA environment can do more: It can check, whether the topology as a whole can meet the security requirements defined by the owner of the application. To do this, a global FRs/SLs vector has to be defined and given to the compiler. It states the security requirements for an application as a result of a prior threat and risk analysis of the application owner. Static checking is quite simple: The compiler just has to check, whether all defined components reach at least the required global value in each of their FRs/SLs vector dimensions (or have a 0-value, i.e. “not applicable”). If so, each component of a topology can meet the global requirements, if not, the compiler can identify all components that need to be revisited and their security deficits (Section 5 shows an example of such a check).

In a scenario where more than one component is available that can fulfill the same functional requirements and where the TOSCA deployment engine is free to choose an appropriate set of resources the security information given in the FRs/SLs vectors is even more valuable. If there are e.g. several compute nodes available, some in a physical secured area while others are on the shop floor, the different security classification can force the software components to be deployed automatically in the secured area. Also, if there are different network options available, some just with plain HTTP and others with HTTPS (i.e. TLS authentication and encryption), the security requirements can lead to an automatic choice of HTTPS.

5 Prototypical Evaluation in a Realistic Scenario

In order to evaluate the proposed solution approach and to demonstrate its feasibility, a realistic scenario with two different use-cases has been identified. The scenario is situated in the area of automated-guided vehicles. The first use-case (cf. App 1) is a user interface (UI) for a manufacturing execution system (MES), referred to as MES User Interface (MES UI). The MES itself is partly deployed in the edge cloud (service 1: MES local) and partly in a private cloud on premise (service 2: MES data center). The second use-case (cf. App 2) deals with the analysis of the vehicle data from various sensors collected by the vehicle data controller (VDC). The VDC provides the available sensor data to the vehicle data

analysis service (service 3: VDA) in a cloud-based system. The overall topology of both use-cases is shown in Fig. 1. The connection to the AGV can be established by using a mobile radio cell (MRC) for LTE, 5G, etc. or a Software-Defined Radio (SDR) for WLAN, ZigBee, etc. as technologies.

Fig. 1. Topology of the corresponding use-cases as used for the prototypical evaluation

The use-case scenario is first described in this section and the resulting main requirements are derived. Afterwards the requirements of one use-case are formally modeled using the proposed modeling approach based on IEC 62443 [IEC15] foundational and system requirements.

5.1 Use-case scenario: Operation and Monitoring of Automated-Guided Vehicles

The whole scenario consists of two use-cases. According to the plant life-cycle the first use-case deals with the operational phase of the automated-guided vehicle (AGV). It encompasses an MES installation using an industrial edge cloud infrastructure. An administrator deploys an existing MES into a new industrial environment with hybrid cloud components, i.e. edge cloud and enterprise cloud components. The MES component deployments are described with TOSCA and also encompass field devices, which are human-machine interfaces (HMI). The field devices are connected with a radio communication technology. The connec-

tion between the HMI and the back-end allows the operator to view existing orders or other monitoring information from the process.

According to this description, the security requirements are derived in terms of the objectives availability, integrity and confidentiality of the relevant information. They can be summarized as follows.

- The **availability** of the HMI panel is less important, because there will be always a local copy of the relevant process information. Since the process is monitored over longer periods of time, the process information does not become invalid immediately.
- **Integrity** of the data must be guaranteed, i.e. the manipulation or changes of the MES data should not be possible by unauthorized persons or devices.
- **Confidentiality** is important, because only authorized personnel should be allowed to operate the HMI panel and to get access to the displayed information.

The second use-case is related to the monitoring of the AGV. The remote monitoring is implemented by sending vehicle data to a cloud infrastructure, where it is used to maintain a digital twin of the AGV. This comprises data such as geographic location, status of the AGV, etc. Usually a public cloud infrastructure is used for such monitoring applications. Therefore, the data communication must be done via untrusted networks.

The following requirements can be derived based on the previous description:

- **Availability:** The connection between vehicle and cloud (digital twin) must always be available to keep the digital twin up to date. Furthermore, it must be possible to send information in a timely manner, which requires sufficient bandwidth and a low latency.
- **Integrity:** Manipulation of the vehicle data results in a wrong virtual representation of the AGV and misleading monitoring decisions.
- **Confidentiality:** The AGV must be able to identify and authenticate itself to associate the provided data with the correct digital twin. Since the process data is probably sent on a byte level, the confidentiality is less critical.

5.2 Security Requirements Modeling based on IEC 62443

After introducing all the needed fundamentals and ideas in the previous sections, we are now able to design the security functionalities of the given topology in order to show and evaluate the proposed approach based on the IEC 62443 standard already described in Section 2. The overall goal is to enable an automatic comparison and evaluation of security requirements from applications (top-down) and security capabilities of e.g. protocols or devices (bottom-up) for the central network management. This creates a future-proof approach to support the industrial networks with additional security functionalities, which can be adapted during runtime in a dynamic manner. To show the usability and applicability of the proposed approach the focus is set on the MES UI use-case from the given scenario.

Based on the overall security goals described in the previous section it is possible to derive certain SL-T values for the seven FRs specified in the IEC 62443 standard. For now, this description depends on the subjective evaluation of the security expert. Therefore, Table 1 represents an exemplary modeling of the security requirements of the MES UI application from the given topology. To understand the values of the given SL-T vector, the evaluation and specification of the first FR, which covers the issues around the identification and authentication control, is further described and presented in the following.

In general, the first FR ensures that every action inside the industrial automation system is supervised and checked before allowing the requesting instance access to the communication network. This FR is further specified by thirteen underlying SRs ranging from the identification and authentication of human users, software applications, regulations for passwords, and the management of wireless access mechanisms. The use-case is assessed with a SL-T value of 3, which can be seen as a summary of all underlying SRs, given by the described requirements of confidentiality and integrity of the transmitted process data. This includes the identification and authentication of all users (humans, applications, and devices) with security mechanisms, which secure the underlying industrial automation system against an intentional non-authenticated access by using technical means, medium resources, automation-related skills, and an overall moderate motivation of the attacker [IEC15].

Table 1. Security Vector Example: MES UI

FR Name SL MES UI Security Mechanisms			
1	IAC	3	User identification and authentication
2	UC	2	User rights and roles management
3	SI	2	Integrity of process data with adequate resources
4	DC	2	Confidentiality of process data with adequate resources
5	RDF	3	Correct handling of wireless connections
6	TRE	1	Only non real-time requirements
7	RA	1	Availability of process data

5.3 Security Capabilities Modeling based on IEC 62443

This section contains the modeling of the security capabilities from the communication protocol functionalities (bottom-up) based on the IEC 62443 standard from Section 2 in order to make them comparable to the specified security requirements from the MES UI application from the previous section. This procedure allows a further distribution of the network traffic onto the present devices and protocols in order to enable a future-proof and resource-adequate network management for the industrial automation system. In order to further understand the modeling approach two examples from the given topology out of the

described scenario are given: ZigBee as a wireless technology from the industrial domain [Zig12] and LTE as a communication protocol from the telecommunication domain. The description offers a simplified view on the protocols and their security functionalities, which can be seen in the described vectors inside [Figure 2](#). Additional examples can be found in the previous publications [EWT⁺17] and [EWT⁺18].

ZigBee is one of the mostly used wireless communication standards for the (Industrial) Internet of Things, which is already widely used in private applications, but has also found its way inside the industrial automation domain. In the specified scenario from Section 5.1 ZigBee is used as a lightweight wireless communication protocol to transmit small amounts of process data in a secure way to the movable AGV on the factory shop floor. The ZigBee Alliance specified ZigBee as an open standard for wireless low-power, low-cost communication for short ranges e.g. 10-100 meters [Zig12]. Although ZigBee was developed with the importance of security in mind, the specification only contains a limited amount of security functionalities and mechanisms in order to keep up with the low resource requirements. [Table 2](#) shows the assessment of the ZigBee communication protocol for our given MES UI use-case from the overall topology.

Fig. 2. Focus on the described industrial communication protocols ZigBee and LTE

In general, the ZigBee standard defines different principles for the inherited security policies e.g. the Coordinator, which configures the overall security level of the communication network, or the Trust Center, which is run on a trusted device inside the ZigBee network and distributes the keys for the end-to-end application configuration management based on either the Standard Security Mode or the High Security Mode [Zig12]. These two exemplary security mechanisms inside the ZigBee standard affect the evaluation of the FRs 1 and 2 for the modeled security vector.

LTE, often called 4G, is the currently widely deployed standard for high-speed wireless communication for mobile devices and data terminals. Nowadays, it is typically used for wide area public telecommunication infrastructures, but it also allows for small picocells or “enterprise femtocells”. These cells just cover

Table 2. Security Vector Example: ZigBee [Zig12]

FR Name SL ZigBee Security Mechanisms		
1	IAC	2 Trust Center for device authentication
2	UC	2 Different security policies available
3	SI	2 Symmetric cryptography (128bit AES)
4	DC	3 Symmetric cryptography (128bit AES)
5	RDF	2 Separation of device types
6	TRE	2 Frame counter against replay attacks
7	RA	1 Frequency agility against simple jamming

100-200m, e.g. a private premise, and provide IP-connectivity similar to WiFi Access Points and to the envisaged future 5G infrastructure for M2M factory communication. LTE security [CFB17] is based on the USIM cryptography hardware that contains a unique key per module. It is used for mutual authentication between the module and the network. Based on a session key derived from the authentication, the communication is encrypted and integrity protected. While there are some known attacks especially by downgrading backward compatible devices to unsafe 2G communication the air-interface of pure LTE can be considered as well protected.

Table 3 shows the assessment of the LTE wireless link in our example given the criteria as defined by IEC 62443. Considering e.g. FR2–Use Control LTE can be rated as security level SL2: while the LTE standard implies authorization checks for all participants, it doesn’t provide measures against hacked devices (security status checks as required in system requirement SR 2.3 for SL3). However, e.g. the authentication (FR1) can be considered as stronger than that in ZigBee. LTE requires a hardware cryptography module – the USIM and, if authentication of an individual user is required, the USIM’s functions can be protected by a PIN. Thus, two-factor authentication can be provided and masquerading, e.g. by cloning of identities is virtually impossible.

Table 3. Security Vector Example: LTE communication

FR Name SL LTE Security Mechanisms		
1	IAC	3 USIM HW authenticates individual devices
2	UC	2 Authorization, but no status check of devices
3	SI	2 Communication integrity, but not device integrity
4	DC	3 Privacy on air link preserved when encryption enabled
5	RDF	3 Cell can be isolated from other networks
6	TRE	3 Trace logs of Node B available
7	RA	1 Can be physically jammed

5.4 Model Checking Evaluation

The complete specification of the sample topology given in in Fig. 1 with all security vectors attributed results in a structured TOSCA YAML file of about 500 line of code. It can be translated using e.g. the Apache ARIA TOSCA compiler into a relational database that contains all the relevant information for further processing the model. On this database the simple static check against the security requirements of the application can be performed as described above: i.e. for all nodes (compute, network, services) is checked, whether their security capabilities vectors are greater or equal in all dimensions than the globally required values.

Fig. 3 depicts this comparison for our sample application and here especially for the wireless link components. The check reveals, that e.g. for the wireless link between the AGV and the fixed infrastructure the ZigBee link cannot fulfill the requirements completely: The application owner requested SL3 for FR1–Identification and Authentication Control. ZigBee’s capability vector only states an SL2 here, e.g. because there is no hardware protection for device authentication. In addition, the demanded SL3 for FR5-Restricted Data Flow cannot be met by ZigBee as well due to e.g. missing functionalities for network segmentation and control of data flow. Without further security measures this would rule out ZigBee for this application and the checker will issue a warning here. In contrast, LTE meets or exceeds with its capabilities the requirements vector in all security dimensions and the model checker won’t throw a warning here. If there is a choice between the two technologies, a future automated deployment tool would dynamically select LTE in order meet the overall required level of security.

	1 - IAC	2 UC	3 SI	4 - DC	5 - RDF	6 - TRE	7 - RA
MES- HMI	3	2	2	2	3	1	1
LTE	3	2	2	3	3	3	1
ZigBee	2	2	2	3	2	2	1

Fig. 3. Checks performed on the capability vectors per security dimension

6 Conclusion

Adaptable manufacturing systems belong to the core concepts of Industrie 4.0. In order to achieve the required level of flexibility for them, a fast and automated deployment of logically defined, virtualized and networked architectures will be an important factor. In this context, security aspects are of utmost importance. However, security is usually handled in a very static way, which contradicts to the needed flexibility and leads to the demand for more flexible approaches for security.

Hence, the paper introduces a simple modeling language that is based on the TOSCA Simple Profile in YAML and extends it towards security controls. The extension consists of security capabilities as additional attributes of components. The idea of this paper is the approach, that these security capabilities not only describe security configuration options and additional security-related functionalities, but also their mapping onto IEC 62443 FRs/SRs and the provided SLs. After the modeling approach is introduced, the paper describes a system model of a real world use-case scenario and the corresponding vectors of security requirements are derived. An example topology is defined and the capabilities of the topology are modeled. The performed evaluation demonstrates that the presented approach is suitable for an automated deployment of innovative industrial IT architectures.

Future work will extend the introduced modeling approach to more complex scenarios. Furthermore, it will be investigated if the current level of granularity of the modelled requirements and capabilities is sufficient in all cases. In addition to this, it has been identified that the automatic mapping of requirements and capabilities is very challenging, especially when the mapping result does not match the expectation. Therefore, this area also requires more research.

Acknowledgments

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

References

- [BBF⁺12] T. Binz, G. Breiter, F. Leyman and T. Spatzier. Portable Cloud Services Using TOSCA. *IEEE Internet Computing* 16(3): 80–85, May–June 2012.
- [CFB17] Jeffrey A. Cichonski, Joshua M. Franklin, Michael J. Bartock. Guide to LTE Security. *Special Publication (NIST SP) - 800-187*. Dec 2017.
- [EWT⁺17] 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. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sept 2017.

- [EWT⁺18] M. Ehrlich, L. Wisniewski, H. Trsek, and J. Jasperneite. Modelling and automatic mapping of cyber security requirements for industrial applications: survey, problem exposition, and research focus. In *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, June 2018.
- [ESB11] M. Enns, R. and; Bjorklund, J. Schoenwaelder and A. Bierman. Network Configuration Protocol (NETCONF) - RFC6241. Internet Engineering Task Force (IETF), Juni 2011.
- [IEC15] IEC. IEC62443-3-3, System security requirements and security levels, 2015.
- [MAB⁺08] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker and Jonathan Turner. OpenFlow: Enabling Innovation in Campus Networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008.
- [OAS13] OASIS. OASIS Topology and Orchestration Specification for Cloud Applications - ver.1.0, November 2013.
- [OAS18] OASIS. OASIS TOSCA Simple Profile in YAML Version 1.1., <http://docs.oasis-open.org/tosca/TOSCA/v1.0/TOSCA-v1.0.pdf>, Januar 2018.
- [Rol13] I. Rolle and P. Kobes. Functional Safety and Industrial Security - State of Standardisation and Risk reduction. In *Elektronik Magazin 8/2013*, August 2013.
- [Tun17] Tunc, Cihan et al. Cloud Security Automation Framework. *IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W)*. 2017, S. 307-312.
- [Zig12] ZigBee Alliance. ZigBee Specification. In *Document 053474r20*, September 2012.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.