Institute of Architecture of Application Systems

Application Threat Modeling and Automated VNF Selection for Mitigation using TOSCA

Karoline Saatkamp, Christoph Krieger, Frank Leymann,
Julian Sudendorf, Michael Wurster

Institute of Architecture of Application Systems,
University of Stuttgart, Germany
{saatkamp, krieger, leymann, sudendorf, wurster}@iaas.uni-stuttgart.de

BibTeX:

```
@inproceedings{Saatkamp2019_ThreatMitigationTOSCA,
  author    = {Karoline Saatkamp and Christoph Krieger and Frank Leymann and
               Julian Sudendorf and Michael Wurster},
  title     = {Application Threat Modeling and Automated VNF Selection for
               Mitigation using TOSCA},
  booktitle = {2019 International Conference on Networked Systems (NetSys)},
  year      = {2019},
  doi       = {10.1109/NetSys.2019.8854524},
  publisher = {IEEE}
}
```


Universität Stuttgart
Germany

2019 International Conference on Networked Systems (NetSys)

Application Threat Modeling and Automated VNF Selection for Mitigation using TOSCA

Karoline Saatkamp, Christoph Krieger, Frank Leymann, Julian Sudendorf, and Michael Wurster

Institute of Architecture of Application Systems, University of Stuttgart, Stuttgart, Germany
[firstname.lastname]@iaas.uni-stuttgart.de

***Abstract*—In the era of Internet of Things (IoT) the interconnectedness of devices, and thus the need to protect them against threats increased. The widely used threat modeling method STRIDE can be used to identify the system's vulnerabilities and to determine appropriate mitigation solutions. In connected environments, especially the network layer plays a critical role in achieving security. Based on the Network Functions Virtualization (NFV) concept, network functions can be virtualized and provisioned on standard IT hardware. Virtualized Network Functions (VNFs) increase the flexibility of the provisioning, and thus security network functions, such as firewalls, can be easily deployed. However, in a complex distributed system it is time-consuming, error-prone, and for application architects even not possible to identify and provision the required security functions. For the orchestration and management of applications the TOSCA modeling language can be used to describe the application's components and their relations in a deployment model. The standard was mainly developed for cloud applications but was extended to the network layer. In this paper, we present a TOSCA-based approach for threat modeling based on STRIDE that facilitates the automated VNF selection and injection into TOSCA deployment models. The feasibility of our approach is validated by an extension of the TOSCA modeling tool Winery.**

***Index Terms*—Threat Modeling, VNF, STRIDE, TOSCA**

I. Introduction

In the era of Internet of Things (IoT), the increasing number of connected devices and applications communicating over the internet raise the demand for solutions to protect these applications against threats. Especially in the industrial context, interconnected applications create critical attack surfaces. Attacks can range from data breaches of personal information, loss of data, or even attacks that put people in danger [1].

Such threats result from system vulnerabilities which must be identified and appropriate mitigation strategies must be chosen [1], [2]. Threat modeling can be used to identify these vulnerabilities and determine their risk and their impact on the system [2]. A widely used methodology to identify threats is STRIDE [2], [3], [4], [5]. It focuses on the identification of potential threats for each component of a system. It is often applied to data flow diagrams that describe systems by their processes, external entities, data stores, and data flows. However, the individual application components, the required middleware and infrastructure components, as well as underlying network components cannot be considered in detail.

978-1-7281-0568-0/19/$31.00

Especially, the network layer plays a critical role in achieving security in connected environments. The introduction of virtualization technologies on the network layer to deploy network functions on standard IT hardware increased the flexibility. The Network Functions Virtualization (NFV) concept introduced by ETSI [6] decouples network functions from proprietary hardware. Of course, this also eases the provisioning of security mechanisms, such as firewalls [7], [8]. However, in a complex distributed system it is time-consuming, error-prone, and for application architects even not possible to identify and provision the required security functions.

The modeling language TOSCA can be used for such automated deployment model adaptations [9], [10], [11], [12]. It is an OASIS standard that was initially developed to ease the orchestration and management of cloud applications in a vendor-neutral manner and has been lately extended to the network layer [13], [14]. In addition, a first draft for a NFV specific data model using the TOSCA language is published [15]. The TOSCA modeling language enables the description of the deployment of an application by its components and their relations. This includes application-specific, middleware, infrastructure, as well as network components.

In this paper, we present a concept based on TOSCA and the STRIDE method (i) to identify threats and thus, vulnerabilities of components in the TOSCA deployment model, (ii) to select appropriate abstract security network functions in an automated manner for mitigating the identified threats, and (iii) to inject a concrete deployable VNF by deployment model refinement for an automated provisioning. For this, we present an approach to model threats as TOSCA policies that can be attached to components in TOSCA deployment models and that can be referenced by components representing security network functions that mitigate one or multiple threats. A selected abstract security network function, such as a firewall, can then be refined and substituted by a concrete implementation depending on the deployment model and the required configuration. The selection, as well as the deployment model refinement, can be automated. For validating our approach, we extended the TOSCA modeling tool Winery [16].

The remainder is structured as follows: Section II introduces fundamentals about TOSCA and the STRIDE method. In Section III our approach is presented and in Section IV the prototypical implementation is shown. Finally, Section V discusses related work and Section VI concludes the paper.

II. Modeling Fundamentals

In the following, we cover the fundamental concepts of TOSCA [13], [14] that are relevant for our approach as well as the fundamentals of the threat modeling method STRIDE [2].

A. Cloud Modeling Language TOSCA

The modeling language TOSCA is an OASIS standard and stands for *Topology and Orchestration Specification for Cloud Applications*. TOSCA can describe the structure and behavior of cloud-based services by focusing on portability and interoperability using a vendor-neutral and technology-agnostic metamodel. The combination of structural and behavioral information is referred as *Service Template*. However, the structure of an application, its *topology*, can be represented as a directed graph with nodes describing the components and edges defining the relations among them. In TOSCA terms, such a structure is called *Topology Template*, which consists of *Node Templates* and *Relationship Templates* as its core modeling entities. TOSCA provides a type system that is intended to express common semantics of the modeling entities, defined in so-called *Node Types* and *Relationship Types*. Figure 1 shows a topology representing the structure of a *PHP* web application, which is provided by an *Apache Web Server* that is installed on an *Ubuntu* virtual machine (VM). In addition, a *MySQL* database is used running on a separate VM for storing the application's state. Each component is an instance of a certain Node Type, e.g., the *Web Server* component is of type *Apache 2.4* and inherits its semantics. This applies also to relations between components: for example, the type *connectsTo* defines that a certain component connects to another one. Furthermore, TOSCA specifies so-called *Deployment Artifacts* that represent the actual business functionality of a component, e.g., the files implementing a web application (cf. *App* component in Fig. 1). Moreover, TOSCA introduces constructs to describe non-functional requirements (NFRs) in the form of *Policy Types* and *Policy Templates*. Policies are utilized to fulfill NFRs of different aspects in application provisioning, e.g., to express aspects regarding quality of service, placement, or restrictions on properties [17], [18], [19]. For example, the database component (cf. top right in Fig. 1) has a policy attached that could enforce a certain strength for the password property *DBPassword*. The more recent versions of TOSCA [14] describe how to express and control network semantics using TOSCA. It defines several normative modeling entities in order to express physical or logical network elements. For example, in Fig. 1 these normative types are used to describe that a virtual machine is bound to a port and this port is in turn linked to a respective network. In 2017, the TOSCA NFV profile was published that specifies a NFV specific data model using TOSCA language [15]. TOSCA for NFV introduces a few new normative Node Types that are aligned with ETSI's definition of components in the NFV domain [20]. However, the presented types can be mapped one-to-one to the existing types in the original TOSCA version. To ease the understanding, these existing types instead of specialized NFV types are used in this work.

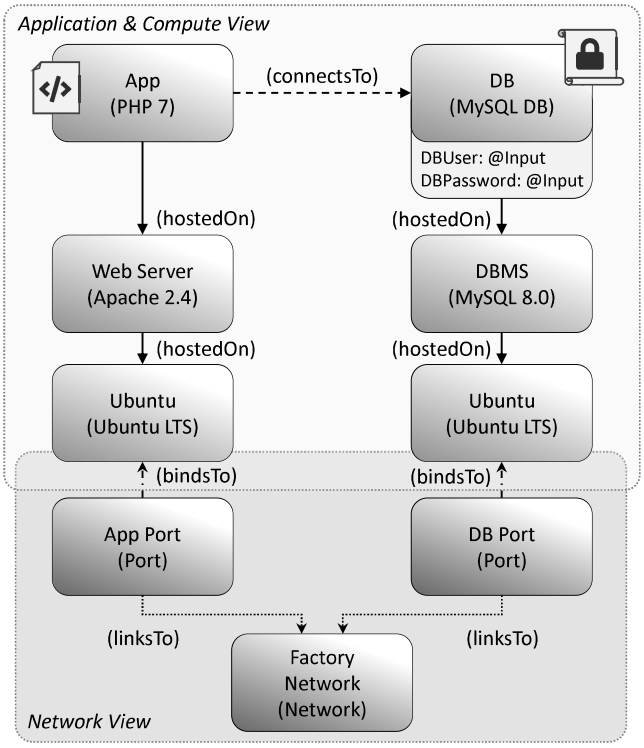

Fig. 1. Example TOSCA Application Topology using NFV elements.

B. Threat Modeling Approach STRIDE

The STRIDE method was initially developed by Microsoft to systematically identify the threats to a system by considering the individual components of this system [2]. For this, STRIDE defines six threat types:

- *Spoofing Identity* includes threats allowing attacker to pretend to be someone or something else.
- *Tampering* threats involve malicious modification of stored data or code as well as of data during transmission.
- *Repudiation* means that an attacker is able to deny that he performed specific actions or transactions.
- *Information Disclosure* threats occur if information are exposed to individuals who must not have access to it.
- *Denial of Service (DoS)* attacks reduce the availability and reliability of a system by flooding specific targets with requests and processing stops for everyone.
- *Elevation of Privilege* threats occur when a user gains right he must not have, e.g., a user gains admin rights.

Such threats can occur due to vulnerabilities of the system's components. Thus, at the end the vulnerabilities causing the threats must be identified for determining an effective mitigation. The analysis of possible threats is often done based on Data Flow Diagrams (DFDs) of a system [2], [5]. However, because it enables a holistic view also on the infrastructure and network components and it provides model adaptation mechanisms, we use TOSCA as a basis for threat modeling as presented in the following sections.

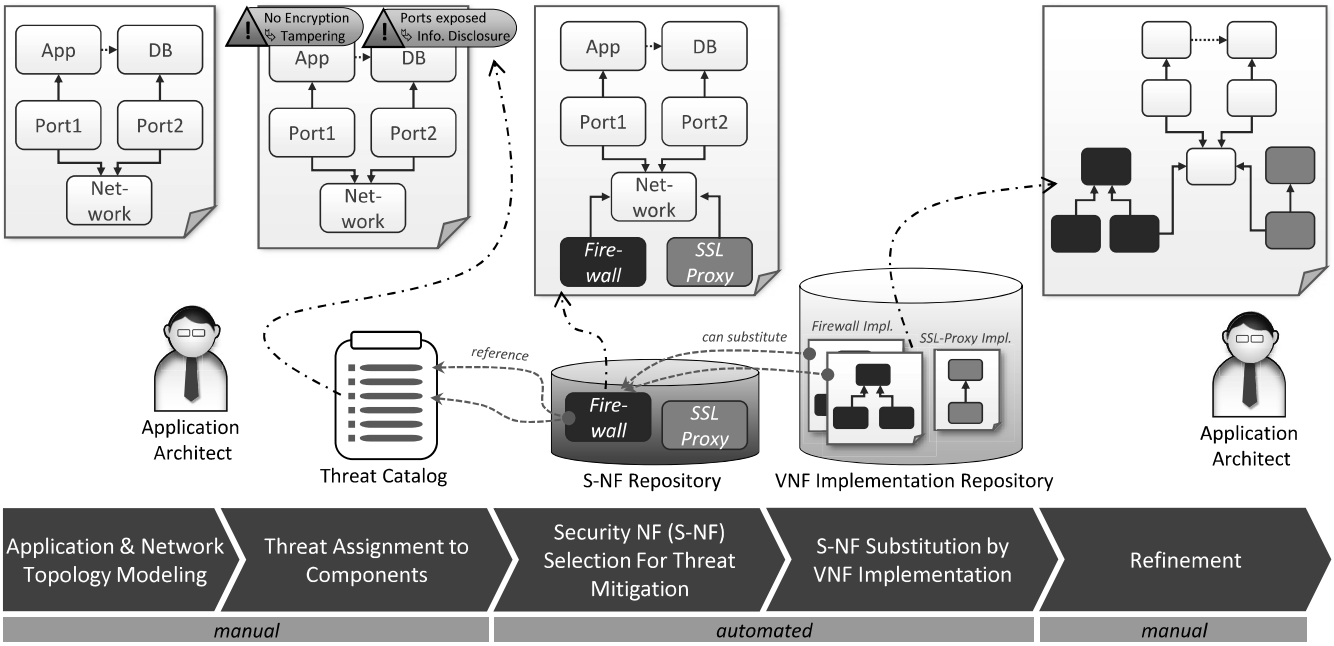

Fig. 2. Overview: Threat Modeling and Mitigation Approach using TOSCA.

III. Threat Modeling & Automated Mitigation

The objective of this paper is to develop a method to support application architects during modeling time to identify potential threats and to select and inject required security network functions into the deployment model. Thus, the resulting TOSCA model describes not only the application but also the required network functions for securing the application. This can be processed by a TOSCA runtime, such as OpenTOSCA [21], for provisioning the modeled components.

Figure 2 presents an overview of the *threat modeling and automated mitigation* approach using TOSCA. All relevant steps from modeling of an application to the injection of the required security network functions are covered. Our approach is based on five steps: First, the application and network topology is modeled using TOSCA. This includes the application-specific components, such as a *PHP* application, middleware components, such as a *Web Server*, infrastructure components, such as *virtual machines*, and network components, such as virtual *ports*. A detailed topology is shown in Fig. 1.

In the second step, the application architect analyzes the components to identify potential threats that result from vulnerabilities of the components. From a *Threat Catalog* the application architect selects the respective threats and annotates the components. Further, the architect can also specify the severity of the threat. In the catalog, for example, the *Ports exposed* threat can be found, which is part of the *Information Disclosure* threats in STRIDE. The catalog contains predefined threats created by security experts. However, new threats can easily be added and categorized according to STRIDE.

For each threat from the catalog, there is a mitigation solution in the form of a security network function (S-NF). Since the exact implementation of such a network function is irrelevant for the application architect, in the third step abstract S-NF are proposed based on the annotated threats in the topology. If multiple S-NFs are available able to mitigate a threat, the application architect can select the preferred one. The S-NFs are stored as abstract Node Types in TOSCA. The selected S-NFs are instantiated as abstract Node Templates in the Topology. In the example shown in Fig. 2 a *Firewall* for mitigating the *Ports Exposed* threat and a *SSL Proxy* for mitigating the *No Encryption* threat are selected and injected.

To obtain a deployable topology, in the fourth step all abstract S-NFs are replaced by concrete implementations. In TOSCA an abstract Node Template can be substituted by a Topology Template stored in a separated Service Template [14]. For this, a *substitution mapping* is defined for the substituting Topology Template. This substitution mechanism is used to replace the abstract S-NFs in the topology by concrete implementations, e.g., the *VM* to host the firewall and the *Ports* bound to the firewall VM. The *Threat Catalog*, the *S-NF Repository*, and the *VNF Implementation Repository* as well as the references and substitution mappings are provided and maintained by security experts. Thus, the set of available security functions and therefore mitigable threats is extensible.

In the final step, the application architect adjusts the resulting topology manually if necessary. For example, if a S-NF is used for mitigating several threats, additional relationships may be required to bind the S-NF to the network accordingly. The resulting deployment model can then be processed by a TOSCA runtime [21] and the application as well as network components are automatically deployed. The next section presents more details about S-NF selection and substitution.

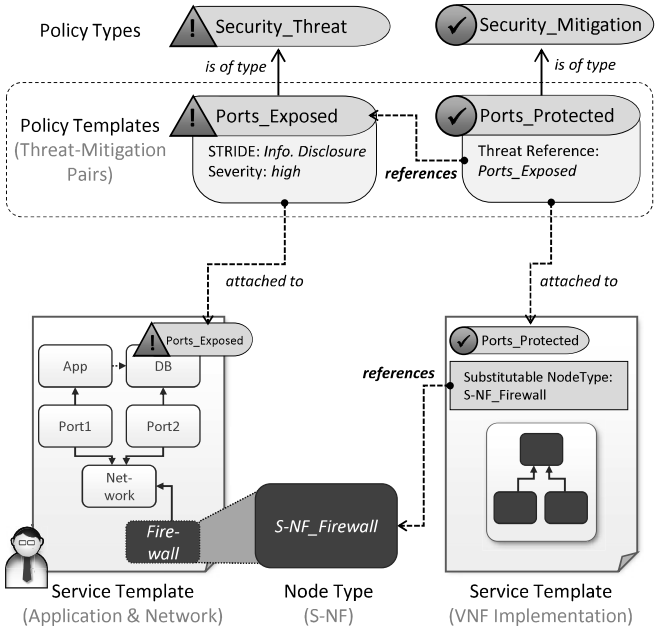

Fig. 3. Overview: TOSCA Elements for S-NF Selection and Substitution.

IV. Prototypical Implementation & Validation

The presented approach in the previous section is realized based on TOSCA and for validating our approach we extended the existing TOSCA modeling tool Winery [16]. For the automated S-NF selection and substitution several TOSCA elements such as Policy Types and Templates are required. In the following, we present details on necessary modeling elements and the enriched system architecture of Winery.

A. TOSCA Threat and Mitigation Modeling

As already mentioned in Section II-A, TOSCA provides a type system to express common semantics for modeling entities. For threat modeling the Policy Type *Security_Threat* is defined that specifies the properties a Policy Template or threat can have. For each Policy Template representing a threat, the *STRIDE type* and the *Severity* are added. An example can be seen in Fig. 3 at the top left. To identify mitigation solutions a corresponding *Security_Mitigation* Policy Type is defined. A Policy Template of this type can use the *Threat Reference* property for referencing the respective Policy Template of type *Security_Threat* that can be mitigated. In the above example, a Policy Template *Ports_Protected* is defined that can mitigate the threat *Ports_Exposed*.

Policy Templates can be attached to Service Templates or Node Templates [13]. Thus, on the one hand, a mitigation Policy Template can be attached to a Service Template containing the VNF implementation that can be deployed to mitigate the respective threat. On the other hand, a threat Policy Template can be attached to a Node Template for which the threat is identified (cf. step two in Fig. 2). The association of a VNF implementation with a Policy Template is done by a security expert, while the identification of threats as well as the respective annotation in an application and network topology model is done by an application architect.

The Service Template in turn references the abstract Node Type, which can be replaced by the Service Template. The abstract Node Type, e.g., the S-NF_Firewall in the example, can be referenced by several Service Templates that contain different implementations of this Node Type, e.g., firewalls from different providers, with different configurations, or for different environments. Due to the references between the two Policy Templates as well as the Service Template and the abstract Node Type, the abstract Node Type that represents a network function being able to mitigate the attached threat Policy Template can be automatically identified. These references are the basis for the selection as well as substitution of S-NFs in TOSCA Topology Templates.

B. Extended Winery Architecture

Winery [16] is a web-based tool for TOSCA to manage and graphically model application topologies. To support and partially automated the described process in Section III, the TOSCA modeling tool Winery is extended[1]. Figure 4 presents the existing (light grey) and newly developed (dark grey) components. The *TOSCA Topology Modeler Editor* is used for modeling application topologies. The *Management UI* is mainly developed to manage all TOSCA elements, such as Service Templates and Node Types. Both frontend components communicate with the backend using the *HTTP REST API*. The backend is capable to import TOSCA elements packaged as self-contained Cloud Service Archives (CSARs). This package format is defined and standardized by OASIS for the use in TOSCA [13]. The backend management component offers an interface to access TOSCA elements in the data store. The *CSAR Packager* exports CSARs that can be consumed and processed by a TOSCA runtime, e.g., the OpenTOSCA Container [21].

Winery has been extended to include threat modeling and mitigation capabilities. First of all, the TOSCA Topology Modeler is extended by a *Threat Modeler*. With this module the user can add threats to Node Templates from the Threat Catalog. Based on the attached Policy Template representing a threat, the available mitigation solution can be computed. For this, the backend provides a functionality to detect matching mitigating S-NFs for attached threats in Topology Templates. The *Mitigation Recommender* browses all VNF implementations and checks their attached mitigation Policy Templates. For all matching Service Templates, the referenced abstract Node Type stored as *Security Network Function* is selected. It is possible that multiple S-NFs are capable to mitigate a threat. Thus, all available mitigation S-NFs are shown in the Threat Modeler and can be selected by the user. For an overview about all attached threats in a Topology Template and their mitigation, the *Threat Assessment* component is developed and integrated into Winery. The *Node Type Substitution* component

[1] https://github.com/OpenTOSCA/winery/tree/thesis/security-aware-nfv

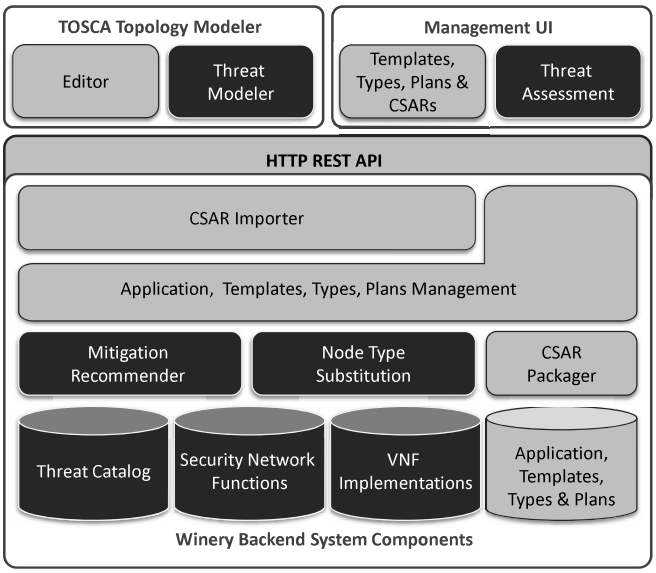

Fig. 4. Enriched Winery Architecture.

facilitates the substitution of the injected abstract S-NF Node Type by a concrete VNF Implementation stored as Service Template. The result is a deployable topology for deploying the application as well as required security network functions.

V. Related Work

To the best of our knowledge, no published work suggest combined approaches (i) to model threats based on STRIDE, (ii) to provide an automated VNF selection based on identified threat, and (iii) to inject selected network functions into deployment models using TOSCA. TOSCA is an open, provider-agnostic standard that can be used to model and deploy cloud native applications [22] as well as complex application stacks combining multiple different technologies, execution runtimes, and cloud providers [23], [11]. Interested readers find an overview of the important concepts in Binz et al. [24]. In addition, TOSCA allows to define the behavioral part of the deployment imperatively, by creating custom workflows that are executed during runtime [25], as well as declaratively, by deriving and generating respective deployment steps fully automated by a runtime [26], [27]. There are existing modeling languages and standards that are based on the notion of *deployment models* [28], such as CloudML [29], [30] or CAML [31]. An overview of different cloud modeling languages, which are similar to TOSCA, is given by Bergmayr et al. [28]. However, the resulting TOSCA deployment models are flexible, portable and can easily be interchanged. Even more, recent research have shown that deployment model adaptation, substitution, and refinement can be realized well with the modeling language TOSCA [9], [10], [11], [12].

Several works demonstrated how policies can be used to specify security requirements for applications that must be enforced during the application's deployment [18], [19], [17], [32]. Compared to the presented approach in this paper, the fulfillment of the security requirements is enforced but an adaptation of the deployment is not addressed and there is no systematic procedure to identify the security requirements. The threat modeling method STRIDE [2] is an systematically approach to identify the system's vulnerabilities using a component-based threat analysis [3], [4], [5], [33]. In this paper, the Policy-based concept to attach security requirements to Node Template and the STRIDE method to identify potential threats that must be mitigated are combined.

Based on the identified threats on the application layer, the network layer is adapted by selecting and injecting required network functions. Network Functions Virtualization (NFV) increases the flexibility of the deployment of network functions on standard IT hardware instead of proprietary hardware [6], [20]. However, existing works that use NFV to ease the integration of security functions based on policies do not provide an integrated view of the network as well as the application components [7], [8], [34], [35]. The strength of the presented approach is the holistic view on the application as well as the network layer that is enabled by TOSCA, since TOSCA is neither restricted to the application layer nor the network layer.

VI. Conclusion & Future Work

The increasing interconnectedness of devices and systems also increases the potential threats to a system. Thread modeling methods can be used to identify system vulnerabilities. However, to identify these threats and to select appropriate mitigation solutions, e.g., firewalls, is time-consuming and requires expert knowledge. In this paper, we presented a TOSCA-based approach to support application architects during modeling time to identify potential threats in application deployment models and to select and inject required security network functions in an automated manner. The approach enables security experts to provide new security functions as reusable entities to mitigate specific threat scenarios. For this, we combined different methods: We used the threat modeling method STRIDE to identify threats, expressed these threats as policies that can be attached to application components, used NFV to provide VNFs as mitigation solutions for identified threats, and selected TOSCA as modeling language.

As future work, we aim to tackle the challenge of VNF placement. The current approach only validates the presence of a required network function but does not enforce the correct placement. In addition, the reuse of existing knowledge of already deployed applications and network functions should be improved. The knowledge can be used to provide further mitigation solutions for application architects. This can be done in context of a case study in an industrial context.

Acknowledgment

This work is partially funded by the BMWi project *Industrial Communication for Factories* – IC4F (01MA17008G) and by the German Research Foundation (DFG) project ADDCompliance (636503).

References

[1] "Cybersecurity for Industry 4.0," Report, Ernst & Young, 2018.

[2] M. Howard and S. Lipner, *The Security Development Lifecycle*. Microsoft Press, 2006.

[3] C. Möckel and A. E. Abdallah, "Threat modeling approaches and tools for securing architectural designs of an e-banking application," in *Proceedings of the 6th International Conference on Information Assurance and Security*, Aug. 2010, pp. 149–154.

[4] G. Macher, H. Sporer, R. Berlach, E. Armengaud, and C. Kreiner, "SAHARA: A Security-aware Hazard and Risk Analysis Method," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, 2015, pp. 621–624.

[5] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "STRIDE-based threat modeling for cyber-physical systems," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Sep. 2017, pp. 169–174.

[6] M. Chiosi, D. Clarke, P. Willis, A. Reid, J. Feger, M. Bugenhagen, W. Khan, M. Fargano, C. Cui, H. Deng *et al.*, "Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action," White Paper, ETSI, 2012. [Online]. Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf

[7] I. Farris, J. B. Bernabe, N. Toumi, D. Garcia-Carrillo, T. Taleb, A. Skarmeta, and B. Sahlin, "Towards provisioning of SDN/NFV-based security enablers for integrated protection of IoT systems," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*, Sep. 2017, pp. 169–174.

[8] I. Farris, T. Taleb, Y. Khettab, and J. S. Song, "A survey on emerging SDN and NFV security mechanisms for IoT systems," *IEEE Communications Surveys Tutorials*, Aug. 2018.

[9] P. Hirmer, U. Breitenbücher, T. Binz, and F. Leymann, "Automatic Topology Completion of TOSCA-based Cloud Applications," in *GI-Jahrestagung*, ser. GI. GI, Sep. 2014, vol. P-251, pp. 247–258.

[10] K. Saatkamp, U. Breitenbücher, O. Kopp, and F. Leymann, "Topology Splitting and Matching for Multi-Cloud Deployments," in *Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER 2017)*. SciTePress, Apr. 2017, pp. 247–258.

[11] K. Saatkamp, U. Breitenbücher, F. Leymann, and M. Wurster, "Generic Driver Injection for Automated IoT Application Deployments," in *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services; Salzburg, Austria, December 4-6, 2017*. ACM, Dec. 2017, pp. 320–329.

[12] L. Harzenetter, U. Breitenbücher, M. Falkenthal, J. Guth, C. Krieger, and F. Leymann, "Pattern-based Deployment Models and Their Automatic Execution," To appear in 11th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2018). IEEE, 2018.

[13] OASIS, *Topology and Orchestration Specification for Cloud Applications (TOSCA) Version 1.0*, Organization for the Advancement of Structured Information Standards (OASIS), 2013.

[14] ——, *TOSCA Simple Profile in YAML Version 1.2*, Organization for the Advancement of Structured Information Standards (OASIS), 2018.

[15] ——, *TOSCA Simple Profile for Network Functions Virtualization (NFV) Version 1.0*, Organization for the Advancement of Structured Information Standards (OASIS), 2017.

[16] O. Kopp, T. Binz, U. Breitenbücher, and F. Leymann, "Winery – A Modeling Tool for TOSCA-based Cloud Applications," in *Proceedings of the 11th International Conference on Service-Oriented Computing (ICSOC 2013)*. Springer, Dec. 2013, pp. 700–704.

[17] K. Képes, U. Breitenbücher, M. P. Fischer, F. Leymann, and M. Zimmermann, "Policy-Aware Provisioning Plan Generation for TOSCA-based Applications," in *Proceedings of The Eleventh International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2017)*. Xpert Publishing Services, Sep. 2017, pp. 142–149.

[18] T. Waizenegger, M. Wieland, T. Binz, U. Breitenbücher, F. Haupt, O. Kopp, F. Leymann, B. Mitschang, A. Nowak, and S. Wagner, "Policy4TOSCA: A Policy-Aware Cloud Service Provisioning Approach to Enable Secure Cloud Computing," in *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*. Springer, Sep. 2013, pp. 360–376.

[19] U. Breitenbücher, T. Binz, O. Kopp, F. Leymann, and M. Wieland, "Policy-Aware Provisioning of Cloud Applications," in *SECURWARE 2013, The Seventh International Conference on Emerging Security Information, Systems and Technologies*. Xpert Publishing Services, August 2013, pp. 86–95.

[20] ETSI, *Network Functions Virtualisation (NFV); Architectural Framework*, European Telecommunications Standards Institute (ETSI), 2013. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf

[21] T. Binz, U. Breitenbücher, F. Haupt, O. Kopp, F. Leymann, A. Nowak, and S. Wagner, "OpenTOSCA – A Runtime for TOSCA-based Cloud Applications," in *Proceedings of the 11th International Conference on Service-Oriented Computing (ICSOC 2013)*. Springer, Dec. 2013, pp. 692–695.

[22] M. Wurster, U. Breitenbücher, M. Falkenthal, and F. Leymann, "Developing, Deploying, and Operating Twelve-Factor Applications with TOSCA," in *In Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services, Salzburg, Austria, December 4-6, 2017*. ACM, Dec. 2017, pp. 519–525.

[23] T. Binz, G. Breiter, F. Leymann, and T. Spatzier, "Portable Cloud Services Using TOSCA," *IEEE Internet Computing*, vol. 16, no. 03, pp. 80–85, May 2012.

[24] T. Binz, U. Breitenbücher, O. Kopp, and F. Leymann, *TOSCA: Portable Automated Deployment and Management of Cloud Applications*, ser. Advanced Web Services. Springer, Jan. 2014, pp. 527–549.

[25] F. Leymann and D. Roller, *Production Workflow: Concepts and Techniques*. Prentice Hall PTR, 2000.

[26] U. Breitenbücher, T. Binz, K. Képes, O. Kopp, F. Leymann, and J. Wettinger, "Combining Declarative and Imperative Cloud Application Provisioning based on TOSCA," in *International Conference on Cloud Engineering (IC2E 2014)*. IEEE, Mar. 2014, pp. 87–96.

[27] C. Endres, U. Breitenbücher, M. Falkenthal, O. Kopp, F. Leymann, and J. Wettinger, "Declarative vs. Imperative: Two Modeling Patterns for the Automated Deployment of Applications," in *Proceedings of the 9th International Conference on Pervasive Patterns and Applications (PATTERNS)*. Xpert Publishing Services, Feb. 2017, pp. 22–27.

[28] A. Bergmayr, U. Breitenbücher, N. Ferry, A. Rossini, A. Solberg, M. Wimmer, and G. Kappel, "A Systematic Review of Cloud Modeling Languages," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–38, feb 2018.

[29] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg, "Towards Model-Driven Provisioning, Deployment, Monitoring, and Adaptation of Multi-cloud Systems," in *Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing (CLOUD 2013)*. IEEE, Jul. 2013, pp. 887–894.

[30] G. Gonçalves, P. Endo, M. Santos, D. Sadok, J. Kelner, B. Melander, and J.-E. Mangs, "CloudML: An Integrated Language for Resource, Service and Request Description for D-Clouds," in *Proceedings of the Third International Conference on Cloud Computing Technology and Science (CloudCom 2011)*. IEEE, Nov. 2011, pp. 399–406.

[31] A. Bergmayr, J. Troya, P. Neubauer, M. Wimmer, and G. Kappel, "UML-based Cloud Application Modeling with Libraries, Profiles, and Templates," in *Proceedings of the 2nd International Workshop on Model-Driven Engineering on and for the Cloud (CloudMDE 2014)*. CEUR-WS.org, Sep. 2014, pp. 56–65.

[32] U. Breitenbücher, T. Binz, C. Fehling, O. Kopp, F. Leymann, and M. Wieland, "Policy-Aware Provisioning and Management of Cloud Applications," *International Journal On Advances in Security*, vol. 7, no. 1&2, 2014.

[33] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, vol. 16, no. 1, pp. 3–32, mar 2011.

[34] M. Pattaranantakul, R. He, A. Meddahi, and Z. Zhang, "Secmano: Towards network functions virtualization (nfv) based security management and orchestration," in *2016 IEEE Trustcom/BigDataSE/ISPA*, aug 2016, pp. 598–605.

[35] C. Basile, A. Lioy, C. Pitscheider, F. Valenza, and M. Vallini, "A novel approach for integrating security policy enforcement with dynamic network virtualization," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, apr 2015, pp. 1–5.

All links were followed on November 29, 2018.