A Concept of an Attack Model for a Model-Based Security Testing Framework

Introducing a holistic perspective of cyberattacks in software engineering

Tina Volkersdorfer, Hans-Joachim Hof
Security in Mobility
CARISSMA Institute of Electric, Connected, and Secure Mobility (C-ECOS)
Technische Hochschule Ingolstadt, Germany
Email: tina.volkersdorfer@carissma.eu, hans-joachim.hof@thi.de

Abstract — **In this paper, we present a framework for model-based security testing. The primary advantage of our framework will be the automation of manual security reviews as well as automation of security tests like penetration testing. The framework can be used to decide on single steps for the test procedure. This paper focuses on the concept of the framework, describing the necessary components and their use. Our framework can simulate the behaviour of an adversary that executes multiple attacks to reach his primary goal. Using our approach, it is possible to continuously and consistently address security in software development, even in the early phases of software engineering when no running code is available. Due to the consistency, some of the necessary tests can be executed with less effort. This makes security tests more efficient. Our preliminary evaluation shows that it is possible to use our attack model in a wide range of domains and that there is potential reuse of modelled elements.**

Keywords-attack model; adversary model; model-based testing; security testing; penetration test.

I. INTRODUCTION

The research project MASSiF (Modellbasierte Absicherung von Security und Safety für umfeldbasierte Fahrzeugfunktionen) addresses model-based safety and security testing in the automotive software domain. In the automotive domain, software engineers thoroughly use model-based safety engineering and model-based safety testing. However, to our best knowledge, there are currently no approaches for holistic attack-model-based security testing. This argument is also supported by [1]. Depending on what the use case requires, a suitable attack model of the existing multitude of isolated solutions is used. If the use case changes or new questions arise, the applied model may have to be updated, or further models may have to be used, e.g., the MITRE ATT&CK Framework [2] (used for details of a specific adversary profile) in contrast to attack trees [3] (focusing the system security on identifying security improvements). Using different models or the constant development of new models is time-consuming and causes security to be inconsistent and untraceable, which in turn may have a negative impact on the quality of security testing. This paper and the associated master thesis [4] introduce a holistic modelling framework for attacks that provides an adversary-based and target-based foundation for a guided model-based security testing to close this gap. As model-based security testing is likely of benefit for other domains than automotive software, our framework is domain-agnostic. For example, penetration testing is usually applied towards the end of software engineering to evaluate implemented security controls

Penetration is a common means to evaluate implemented security controls [5]. However, penetration testing usually takes place in the late phases of software development, when it is expensive to fix security problems. Also, the effectiveness and efficiency of penetration test depend on the skills of the tester[5]. Vulnerabilities could go unnoticed. In contrast, a holistic attack model that provides automated mechanisms for generating security test cases could be applied in the early design phase. Hence, it mitigates some of the shortcomings of penetration testing. Our approach is a complement for penetration tests. The automatable test execution is more cost-effective, and early weaknesses can already be detected. The primary focus of this paper is on the attack model and its use in the framework.

The rest of this paper is structured as follows: Section II discusses related work on attack modelling. Section III states the requirements for the holistic modelling framework for security testing. Section IV presents our approach to attack modelling. Section V shows the preliminary evaluation of the part of the modelling framework presented in this paper. Section VI concludes the paper.

II. RELATED WORK

Several adversary and attack models exist. Dependent on the perspective of the attack, there are various modelling concepts.

The process modelling approach focuses on representing the attack based on phases. For example, the Lockheed Martin Cyber Kill Chain [6] defines an attack with seven phases that have to be passed through by the adversary. The kill chain model intends to model Advanced Persistent Threats (APTs) and malware behaviour. Hence, an attack is seen as a linear process, and it does not represent information about the attack surface that is provided for an adversary. Testing requires exploring multiple attack techniques, so bare process modelling approaches are typically not sufficient for testing.

Another standard method is graph-based modelling that uses attack graphs to represent various attack opportunities. Kaynar [7] presents examples of this class of adversary and attack models in the domain network security.

 ISBN: 978-1-61208-821-1

A specific graph representation of attacks is the attack tree by [3]. An attack tree focuses on the primary goal of an adversary. This primary target represents the root of the attack tree, the elementary attack steps to are the leaves, and the various associated subgoals link these nodes. Existing attack trees can easily be reused or combined to form more comprehensive attack trees for threat and risk analysis. Attack trees incorporate multiple paths adversaries may take, but they do not include any characteristics of an adversary or about an adversary's decision on next steps in an attack. Efficient testing requires an approach that also takes into consideration realistic assumptions about attack paths. Our work uses tree structures in combination with adversary modelling and target modelling to overcome the shortcomings of attack trees.

Classification modelling approaches model attacks on different abstraction levels. For example, MITRE proposes the ATT&CK framework [2] to model attacks based on the adversary's perspective. Tactics, techniques, and procedures define adversary behaviour. MITRE ATT&CK can be used both to derive behaviour-based adversary scenarios and to establish attack profiles of an implemented system. It is suitable for testing and verifying the security of a software product. However, the MITRE ATT&CK framework is not designed for use in the early design phase to support a model-based security testing based on a specific adversary strategy. Our work closes this gap.

There are also combined approaches to attack aspects shown above. Adepu and Mathur [1] present unified adversary and attack models with a focus on both security and safety aspects in the context of Cyber Physical Systems in [8]. The relevant system information is part of an attack domain model. However, Adepu and Mathur limit the proposed framework by not considering the characterization of an adversary, e.g., the adversary's current knowledge about the target. However, realistic assumptions about an adversary are necessary for comprehensible modelling the strategic and tactical attack actions of this adversary.

ADVISE [8] is the work most similar to our approach. It addresses the structured and goal-oriented procedure of an adversary. ADVISE is based on an executable security model on system-level to generate security metrics. Our work can be applied earlier in the design process of a system. The application of ADVISE is neither limited to a specific domain nor a certain level of detail. In contrast to our work, the adversary's decision function of ADVISE for the simulated attack procedure does not include the different aspects of designing and launching an attack, e.g., reconnaissance actions. ADVISE is proposed for a repeatable usage in the security engineering to support the evaluation of system security. However, this security analysis method is not designed to support security testing by providing a guideline for performing security tests based on a specific adversary.

III. REQUIREMENTS

In this paper, we propose a concept for a holistic attack modelling to support the model-based security testing by simulating the strategic actions of an adversary in terms of traceability. The general basis for the requirements engineering is [9] by interpreting security tests as business processes. Concerning the modelling of dynamic behaviour, there are analogies between model-based testing and models for business processes [9] [10]. Using models, complex scenarios can be simulated. The suggested model is intended to be used to decide on the next steps during testing activities, e.g., the structured use of existing penetration testing tools. Hence, relevant requirements for the design of our approach can be derived from [9]:

a) Model-based: The expectation is that applying a model-based perspective to an attack presents a suitable basis for formalization similar to the formalization of the software development process in IT that came with the introduction of model-based software engineering [11]. This formalization is a basis for automation of security testing of system models.

b) Expressive: The purpose is to model as many attacks as possible by the proposed general attack model. A generic attack model should express all necessary information regarding attack, adversary, and target. As already shown in Section II, most attack models only incorporate certain aspects of an adversary and the target. Area of application is a relevant factor for the choice of an attack model. Using a holistic attack model for multiple use cases can involve less effort than the application of several different attack modelling techniques, and it offers a widespread usage.

c) Reusable: The holistic attack model should consist of reusable components to reduce the time-consuming modelling of new attacks [9]. For example, already modelled elements of the attack model should be reusable for as many different use cases as possible (e.g., change of target, change of adversary, change of attack). The requirements a) "model-based" and b) "expressive" support this requirement "reusable".

d) Systematic: The proposed model should ensure a systematic and continuous (re-)use of attack information in all phases of the software engineering process. Today's software development often lacks such a systematic and continuous reuse of information about attacks.

e) Consistent: The proposed attack model should be consistent. A consistent model can be verified and validated. Formalization and automated tool support require a consistent model.

f) Visualizable: The model should use visual means to model attacks. An appropriate visual graphic representation of attacks facilitates the readability and understandability, especially of complex attacks. Visual illustrations are more intuitive for humans than prose text [9] [10]. The use of visual elements supports the formalization as it is missing the ambiguity and inaccuracy of prose. The aim is to achieve a concise expressiveness of the model. The connection of individual attack model components should be easily identifiable, such that security can be consistently verified and software quality increases.

g) Understandable: Software engineers that are no security experts should be able to use our models throughout the software development process. Hence, our models should be understandable, easy to learn and uncomplicated to use. Complex models tend to be difficult to understand [11]. This disadvantage should be avoided.

 ISBN: 978-1-61208-821-1

IV. Design Of An Attack Model for a Model-Based Security Testig Framework

This paper introduces a holistic modelling framework for attacks that provides an adversary-based and target-based foundation for a guided model-based security testing. We postulate the following scope for our approach. Future work will probably leverage some of these restrictions:

- The proposed attack model is limited to one or more cyber-enabled capabilities [12] as a target. The term "cyber-enabled capability" describes any software enabled technology that can be influenced by an adversary in various ways [12]. Attacks targeting on humans (e.g., Social Engineering) are out of scope.
- Our model is limited to adversaries that follow a rational goal. Random attacks are out of scope of this work. The method is limited to goal-oriented adversaries.
- The focus of this paper is to identify the necessary conceptional elements for a suitable, holistic attack modelling framework. Completeness, detailed specification and implementation of these elements are out of the scope of this paper.

Overview

In our attack model, we associate each attack with an adversary and the system under attack (target). An adversary plans, develops, and executes attacks against the target by using specific resources. The target may provide one or more access points for an attack. Both for the construction of the proposed model and its execution, it requires this basis of the content in the context of attacks.

Figure 1 shows the essential elements of our framework: the attack model, adversary model and the target model. The adversary model characterizes a specific adversary. Each adversary is defined by descriptive attributes, the goal of his attack, and his current knowledge about the target (called adversary perspective model in Figure 1). The target model represents the system under attack and all necessary associated components of the environment that can be exploited by an attack attempt. The attack model connects all components of the framework. Each attack is simulated within an iteration of defined steps. For this purpose, all necessary information from the attack base is used.

Figure 2 illustrates the pictorial representation of the context of two elementary attack iterations. In each step of the attack simulation, one elementary attack iteration is executed. The adversary's primary goal sets the direction for each attack iteration. Depending on the current adversary's knowledge, he attempts to exploit the target by an available access point. The attack base provides all actions that an adversary could possibly execute in an elementary attack iteration. For example, it stores knowledge about possible weaknesses, available attack techniques, and exploits for the vulnerabilities. The simulation of the target model provides the effect of the attack on the target. The use of a target model allows executing attacks on systems that do not yet exist. Each simulation step ends with an update of the adversary and the target model. When the adversary reached his primary goal, the simulation terminates. Otherwise, the next iteration starts.

Our executable attack model simulates the strategical approach of a specific adversary to attack a particular target. It takes into consideration the properties of this adversary as well as the knowledge the adversary has about a target system at a given time. Security testers can use each iteration to derive security tests. Thus, this holistic method for attack modelling provides a holistic basis for model-based security tests.

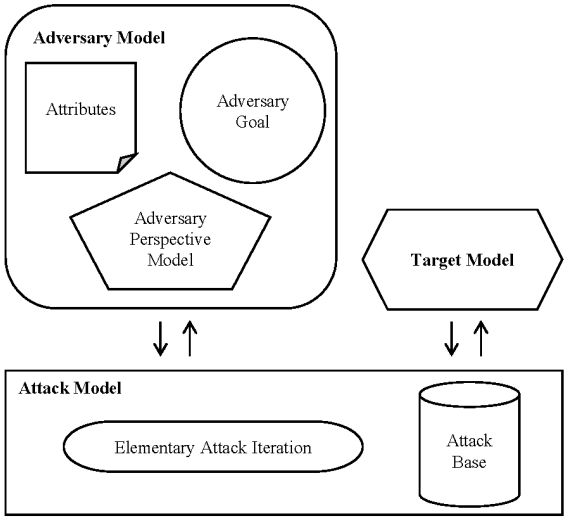

Figure 1. Components of the framework.

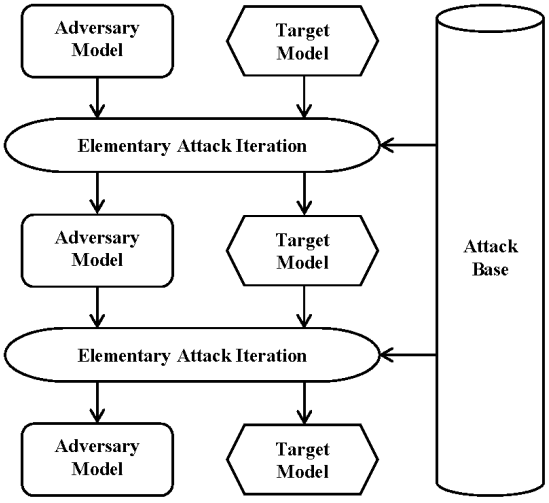

Figure 2. Interaction of the components shown for two elementary attack iterations.

Adversary Model

The adversary model consists of three main components: Adversary primary goal, characteristic attributes, and the adversary perspective model, as shown in Figure 1.

The adversary primary goal indicates the direction of the attack. For example, the primary goal of an adversary may be the extraction of financial data from a financial data transfer system. During modelling, the adversary's goal does not change. The adversary's goal is used to derive the behaviour of an adversary during an attack simulation. In the example above, the goal derives attractive data stores in the target system. The adversary tries to navigate from any access points available to the adversary to these data stores.

Attributes characterize each adversary. Attributes include, e.g., the location of an adversary (remote adversary, local adversary) and his skills.

The adversary perspective model represents the adversary's knowledge about the target at a given time. During the attack simulation, each elementary attack iteration increases this knowledge, thus changes the adversary perspective model. For example, the adversary may get access to further access points after the first attack iteration, that he can use for an attack attempt in the next attack iteration. As long as the adversary's current knowledge is not sufficient to achieve his primary goal, the adversary tries to expand his knowledge in the appropriate direction through further attack attempts.

The difference between the target model and the adversary perspective model is that the target model holds only correct information. Still, the adversary perspective model may keep incomplete or blurry details on the target. It represents the current, preliminary view an adversary usually has on the target.

Target Model

The target model represents one or more cyber-enabled capabilities that an adversary wants to attack. For example, the target model holds information about available access points of the target. An access point, based on [13], provides adversaries unintended access or unintended information disclosure. The access point is either part of or related to a cyber-enabled capability. During an attack iteration, an adversary analyzes or uses access points to gain knowledge or to control or manipulate the target.

When an adversary chooses to execute an exploit as part of an elementary iteration, this exploit is applied to the target model. The outcome of the exploitation updates both the target model and the adversary model. Thus, the target model is a necessary element for holistic attack modelling.

Attack Characterization and Simulation

The attack model shows various perspectives of an attack. The process perspective focuses on the execution of an attack. The attack model simulates each attack within one iteration that incorporates four steps. We call such an iteration an elementary attack iteration, as shown in Figure 1, as it constitutes the smallest attack unit possible from a process perspective. Each elementary attack iteration includes the four steps: (1) Identify available access points, (2) Select one access point, (3) Probe the target, and (4) Update adversary's knowledge. To achieve the adversary's primary goal, usually, several elementary iterations are necessary.

The technical perspective focuses on the selection of available exploits in a proper order to achieve the adversary's goal. An exploit is an umbrella term for various means of actions to execute attacks [14]. It represents one specific step of an attack and is the elementary element of the technical perspective. An attack technique summarizes the necessary exploits to achieve an adversary's primary goal. Selection and execution of an exploit in our simulation can be subject to preconditions [2]. For example, the adversary first has to ensure that the provided access point is vulnerable before he can take further actions in this regard.

The strategic perspective brings together both the process perspective and the technical perspective. It simulates the strategical behaviour of the adversary in the attack simulation, as exemplarily shown in Figure 2. To do so, it selects the next iteration in the attack simulation as well as decides, when the adversary reached his primary goal.

V. PRELIMINARY EVALUATION

In this section, we evaluate if our attack model framework meets the requirements of Section III. We evaluated the attack model framework under the following restrictions (future work will leverage some of these restrictions):

- The application of the modelling is limited in each case to one attack iteration.
- The attack scenarios under consideration focus the first activities of an attack, comparable to Reconnaissance of the Lockheed Martin Cyber Kill Chain [6].
- The proposed attack model is applied to two significant attack scenarios by way of example. The first example incorporates vulnerabilities of the OWASP Top Ten 2017 [15], hence is highly relevant in the domain of web application. In contrast, the second example stems from the automotive domain. We use the idea of UML activity diagrams [16] and attack tree [3] for our examples. We choose UML as it is common in many relevant application domains.

Evaluation Criteria

The following criteria were identified for the evaluation:

a) Model-based: The criterion refers to the extent to which the attack modelling method is based on a model.
b) Relevant attacks: The criterion refers to the extent to which relevant attacks can be modeled using the proposed attack model.
c) Application domain independence: The criterion refers to the ability to model different attacks independently of the application domain.
d) Reusable elements: The criterion refers to the extent to which the modelled contents and elements of the attack model can be easily reused in conjunction with other attack scenarios.
e) Systematic structures: The criterion refers to the extent to which there is a systematic approach to the structure and procedure of the proposed attack modelling concept so that an attack can be modelled in a comprehensible and repeatable way.

 ISBN: 978-1-61208-821-1

f) Visual elements: The criterion refers to the extent to which the proposed attack model has graphic elements or can be illustrated visually at a glance.

A consistent model is a requirement for the use of automatism [9] and thus, a suitable basis for supporting security tests. Therefore, proper syntax and semantic for the necessary elements have to be defined. The specification of the individual elements of the proposed concept is out of the scope of this paper. Therefore, we omit the evaluation of the model consistency. The understandability of a model helps to evaluate its usefulness. Also, we omit the evaluation of the requirement understandability. We will survey relevant stakeholders to assess the understandability of the model at the end of the still running research project MASSiF.

Findings

We iterated through the proposed attack model based on two exemplary attack scenarios. Due to lack of space, we only present an extract of exciting findings in Figure 3 and Figure 4 concerning the elementary attack step (3).

In the first scenario, we model an identity theft attack on a social media platform. This scenario incorporates attacks from the OWASP Top Ten 2017 [15]. Figure 3 shows a technical perspective of an attack on the user input field of a web application. Visualized using tree structures, the adversary selected Credential Stuffing as an attack technique and utilizes an associated exploit.

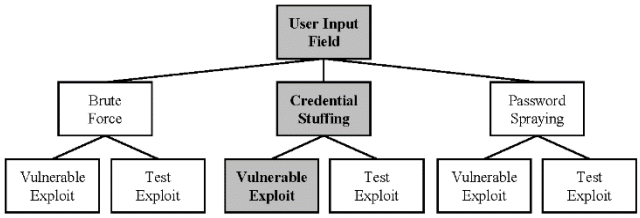

Figure 3. Selected exploit based on the access point "User Input Field".

In the second scenario, based on the research project MASSiF, an adversary tries to manipulate data on an Electronic Control Unit (ECU) in a vehicle. Figure 4 shows a technical perspective of an attack on the standardized interface OBD-2 (On-board-diagnose) Connector in a vehicle. The adversary selected Extraction Technique to extract data from the vehicle.

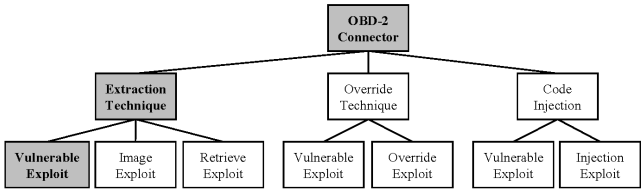

Figure 4. Selected exploit based on the access point "OBD-2 Connector".

The respective application of the exploits on the corresponding target model for the scenario leads to new information for the adversary, e.g., the specific access point is vulnerable. During the next iteration, the adversary can select the next exploit based on the new information the adversary gained from the previous attack iteration.

Interpretation

Using the example of tree structures and UML activity diagrams, model-based elements could be used systematically for attack modelling. The criterion model-based can be confirmed insofar as the developed attack model provides a suitable foundation for different modelling approaches.

The criterion relevant attacks can be confirmed to the extent that we were able to model two representative examples from very different application domains. In our opinion, the proposed concept provides a suitable basis for modelling attacks, independent of the domain. Consequently, the proposed concept accomplishes the criterion application domain independence. However, it is still an open question to what extent the specific characteristics of individual domains must or can be captured.

The content of the attack basis, e.g., defined exploits, attack techniques, or access points, as well as the elementary attack iteration process itself are exemplary representatives of reusable elements. Likewise, and regarding the criterion reusable elements, the adversary model can be applied repeatedly, e.g., with different starting positions of the adversary's knowledge for the attack modelling.

The elementary attack iteration represents the basic, systematic guideline for the execution of the attack model. Likewise, the adversary model, target model and the attack basis represent a suitable foundation for a systematic deployment, representation and reuse of attack information. In this respect, the criterion systematic structures is accomplished.

The exemplary use of tree structures and UML activity diagrams shows that the attack modelling concept provides a suitable basis for the integration of graphical model elements. In this respect, the criterion of visual elements is accomplished.

We evaluated five out of seven requirements. In the context of the criteria, our attack model meets the requirements model-based, expressive, reusable, systematic, and visualizable. Requirements e) “consistent” and g) “understandable” can only be meaningfully evaluated in a later stage of the research project MASSiF. Hence, we did not evaluate these criteria.

VI. Conclusion and Future work

In this paper, we present the concept of a framework for model-based security testing. The framework addresses security throughout the software engineering process. The main goal of the framework is the automation of security tests, especially of security tests in early phases of software engineering (e.g., manual security review).

The central part of our framework is the attack model. The attack model offers several perspectives on security. It can simulate an attack against a target system model. Using a target system model allows simulating attacks on software systems that are not yet implemented. The primary goal that the adversary wants to achieve drives the simulation and offers multiple paths of attacks. Our approach can be used to drive security testing to increase its quality. The preliminary

 ISBN: 978-1-61208-821-1

evaluation of the attack model shows that the model is expressive, reusable, systematic, and visualizable.

Future work will focus on detailed specification, implementation of the proposed elements, particularly the attack base and the testing part of the proposed framework.

ACKNOWLEDGMENT

This work is part of the project MASSiF (Modellbasierte Absicherung von Security und Safety für umfeldbasierte Fahrzeugfunktionen). It is supported by the German Federal Ministry of Education and Research (BMBF) under the KMU-innovative program.

REFERENCES

[1] S. Adepu and A. Mathur, "Generalized Attacker and Attack Models for Cyber Physical Systems," in *2016 IEEE 40th Annual Computer Software and Applications Conference*, Piscataway, NJ, IEEE, 2016, pp. 283-292.

[2] B. E. Strom *et al.*, "MITRE ATT&CK: Design and Philosophy," July 2018. [Online]. Available: https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf. [retrieved: 2020.09.25].

[3] B. Schneier, "Attack Trees," *Dr. Dobb's Journal,* vol. 24, no. 12, pp. 21-29, 1999.

[4] T. Volkersdorfer, *Methodik zur Angriffsmodellierung für Security-Tests [Attack Modelling Methodology for Security Tests],* Technische Hochschule Ingolstadt, Germany: Master thesis at Department for Computer Science, 2020.

[5] K. Scarfone, M. Souppaya, A. Cody, and A. Orebaugh, *Technical Guide to Information Security Testing and Assessment,* 800-115 ed., Gaithersburg, MD 20899-8930: National Institue of Standards and Technology, 2008.

[6] E. Hutchins, M. Cloppert, and R. Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis," *Leading Issues in Information Warfare & Security Research,* vol. 1, pp. 80-106, January 2011.

[7] K. Kaynar, "A taxonomy for attack graph generation and usage in network security," *Journal of Information Security and Applications,* vol. 29, pp. 27-56, August 2016.

[8] E. LeMay *et al.*, "Adversary-Driven State-Based System Security Evaluation," in *Proceedings of the 6th International Workshop on Security Measurements and Metrics*, New York, NY, USA, Association for Computing Machinery, 2010, pp. 1-9.

[9] A. Drescher, A. Koschmider, and A. Oberweis, Modellierung und Analyse von Geschäftsprozessen [Modelling and Analysis of Business Processes], Berlin, Boston: De Gruyter Oldenbourg, 2017.

[10] M. Winter, T. Roßner, C. Brandes, and H. Götz, Basiswissen modellbasierter Test [Basic Knowledge Model-Based Test], Heidelberg: dpunkt.verlag, 2016.

[11] J. M. Borky and T. H. Bradley, Effective Model-Based Systems Engineering, Cham, Switzerland: Springer, 2019.

[12] The MITRE Corporation, "CAPEC Glossary," 4 April 2019. [Online]. Available: https://capec.mitre.org/about/glossary.html. [retrieved: 2020.08.07].

[13] J. Bryans *et al.*, "A Template-Based Method for the Generation of Attack Trees," in *Information Security Theory and Practice*, Cham, Springer International Publishing, 2020, pp. 155-165.

[14] H. Siller, "Exploit," Springer Gabler, 19 February 2018. [Online]. Available: https://wirtschaftslexikon.gabler.de/definition/exploit-53419/version-276511. [retrieved: 2020.09.15].

[15] The OWASP Foundation, "OWASP Top 10 - 2017: The ten most critical web application security risks," 2017. [Online]. Available: https://owasp.org/www-project-top-ten/. [retrieved: 2020.08.13].

[16] Object Management Group, "Unified Modeling Language," 5 December 2017. [Online]. Available: https://www.omg.org/spec/UML/2.5.1/PDF. [retrieved: 2020.09.10].

 ISBN: 978-1-61208-821-1